SONY

ASK 2012

Nagoya, Japan

Recent Meet-in-the-Middle Attacks on Block Ciphers

Takanori Isobe

Sony Corporation

(Joint work with Kyoji Shibutani)

Outline

1. Meet-in-the-Middle (MitM) attacks on Block ciphers

2. MitM on Block cipher having *simple* KSF
   - XTEA, LED, Piccolo (@ ACISP 2012 w/ K. Shibutani)
   - GOST (@ FSE 2011 and JoC)

3. MitM on Block cipher having *complex* KSF
   - All subkeys recovery attack (@ SAC 2012 w/ K. Shibutani)
     - KATAN-32/48/64, SHACAL-2, CAST-128

4. Conclusion

1.Meet-in-the-Middle Attack on Block Cipher

Background

- Meet-in-the-Middle（MitM）attack was proposed by Diffie and Hellman (1977)
  - Applied to Block Cipher such as Triple DES.

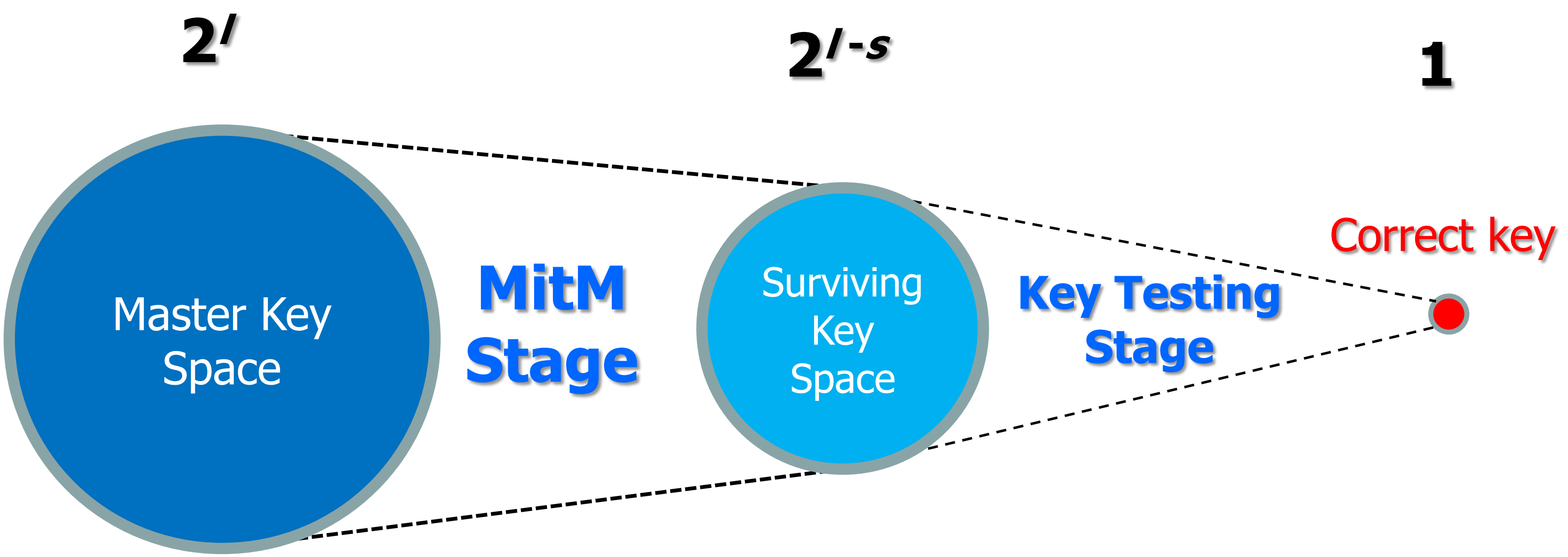
- It was extended to Preimage Attack on hash function by Aoki and Sasaki (2008) [AS08]
  - Develop several novel techniques: Splice and Cut, Initial structure, partial matching [AS08, SA09, KRS12]
  - Full Preimage Attacks on MD5 and Tiger [SA09, GLRW10]
  - Best Preimage Attacks on SHA-1 and SHA-2 [KK12, KRS12]
  - Convert it into Collision attack : Pseudo collision on SHA-2 [LIS12]
Collision Attack on Skein [K'12]

- Recently, MitM is applied to Block cipher with several techniques.
  - Single Key Attacks on full KTANTAN and GOST [BR09, I'11]
  - Best Attacks on AES, IDEA, XTEA, LED and Piccolo [BKR11, KLR11, IS12]

Meet-in-the-Middle Attack on Block Cipher [BR09]

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

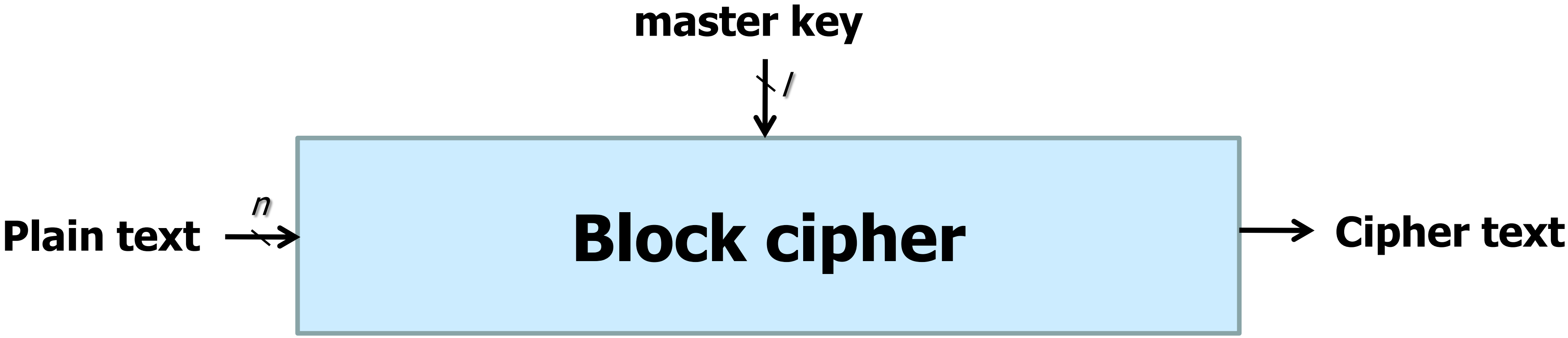
@ MITM stage : Filter out a part of wrong keys by using MitM techniques
@ Key testing stage : Find the correct key in the brute force manner.



l : key size in bit
s : matching state in bit

MitM Stage

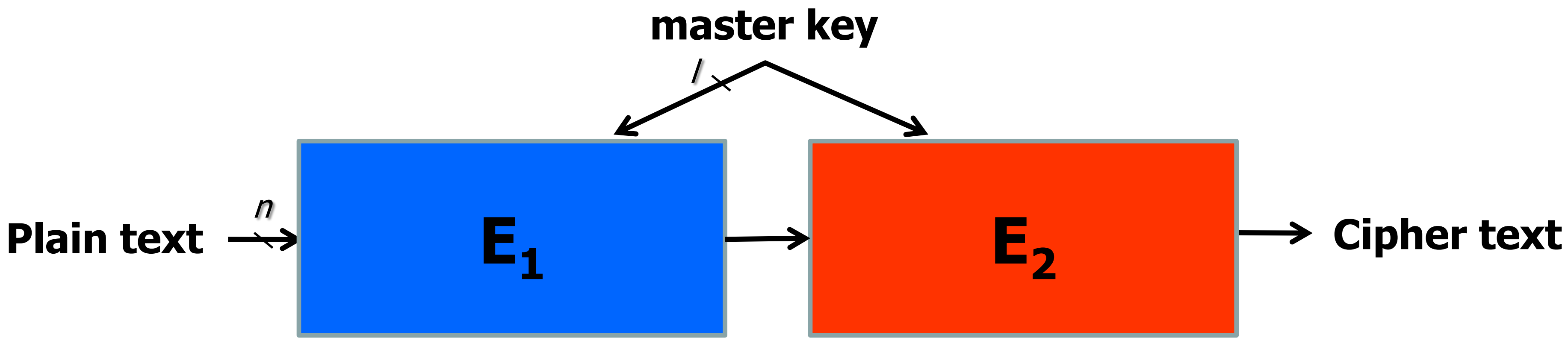
- Preparation
  - Divide Block cipher into two sub function E_1 and E_2



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

MitM Stage

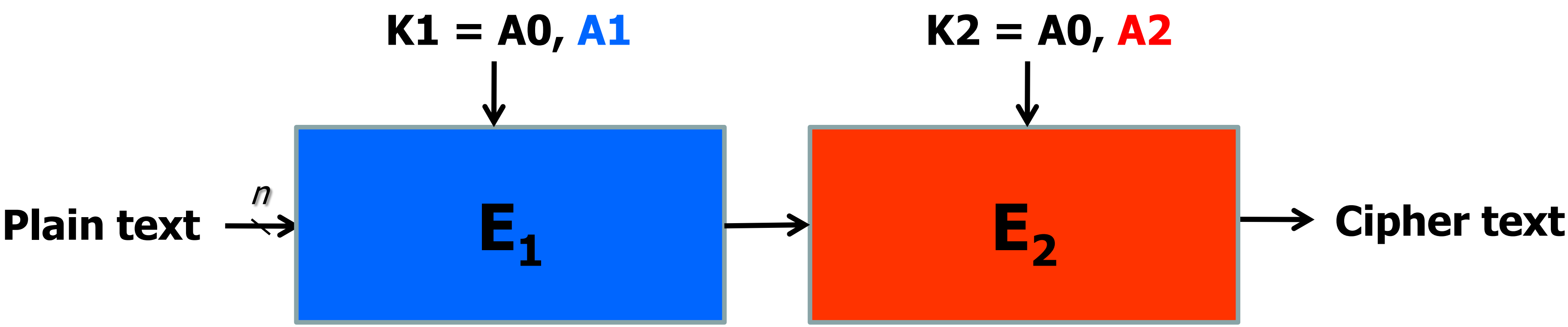
- Preparation
  - Divide Block cipher into two sub function E_1 and E_2



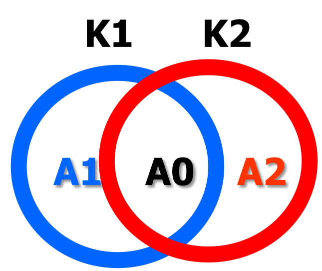
Block cipher : l bit master key and n bit block size

MitM Stage

- Preparation
  - Divide Block cipher into two sub function E_1 and E_2
  - Construct 3-subset of master key **A0, A1,** and **A2**



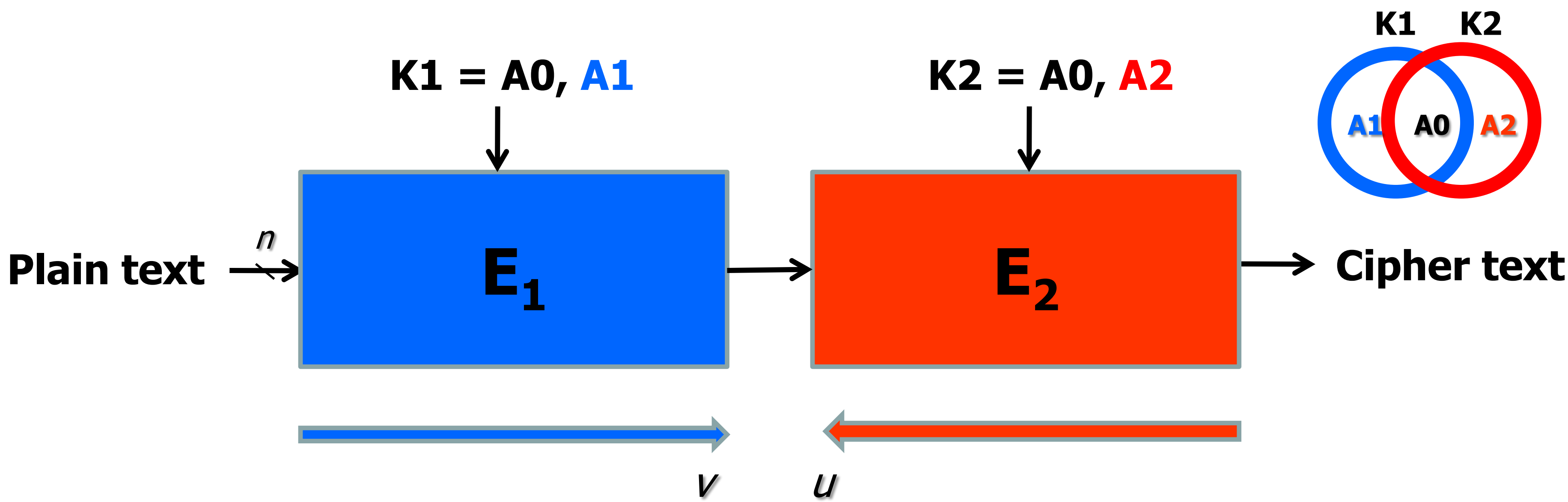
Block cipher : *l* bit master key and *n* bit block size



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

K1: sub set of key bits used in E_1.
K2: sub set of key bits used in E_2.

MitM Stage



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)

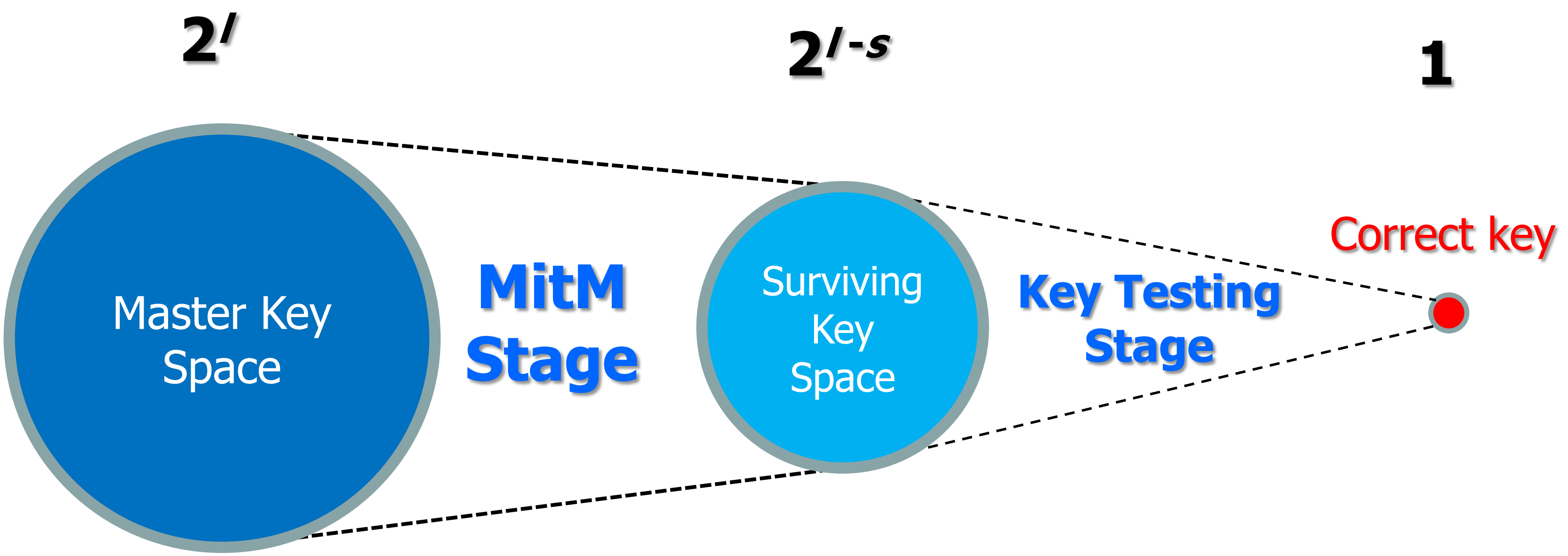
# of surviving key candidates :

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

l : key size in bit
s : matching state size

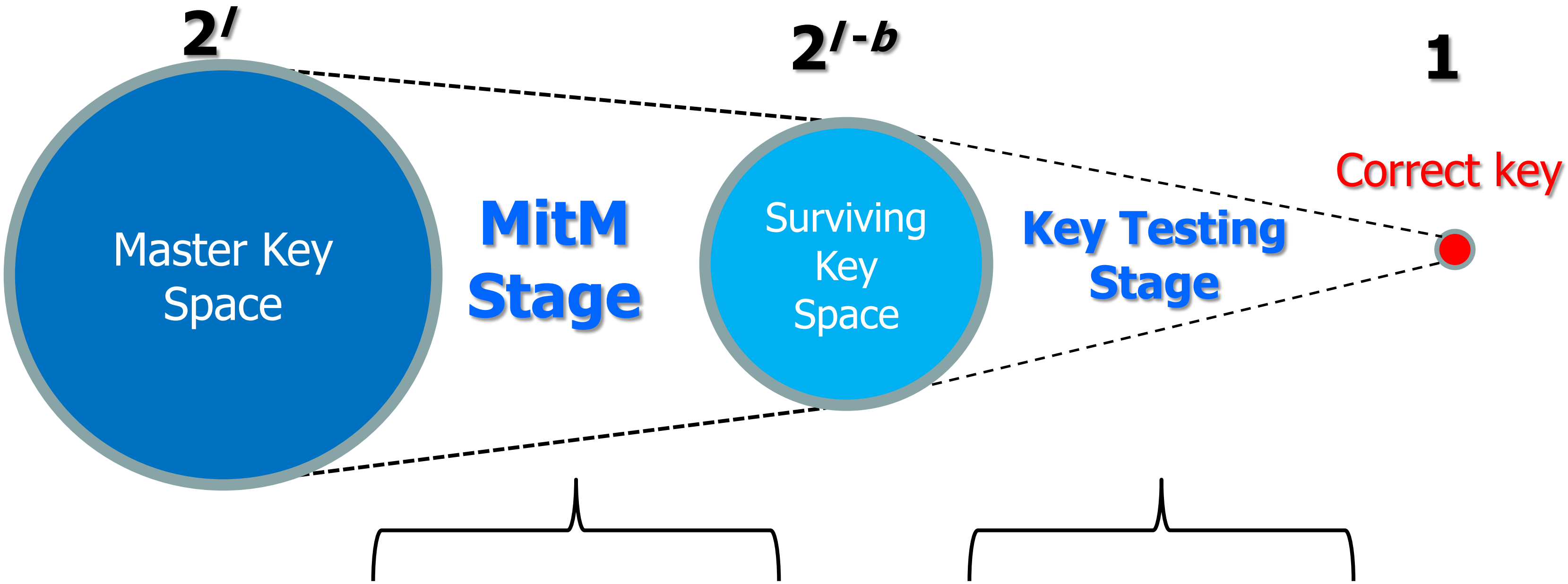
Key Testing Stage

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



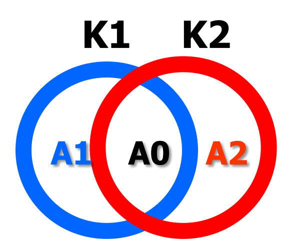
l : key size in bit
s : matching state size

Evaluation



$$\text{Complexity} = 2^{|A0|}(2^{|A1|}+2^{|A2|}) + (2^{l-s}+2^{l-2s}+\dots)$$

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



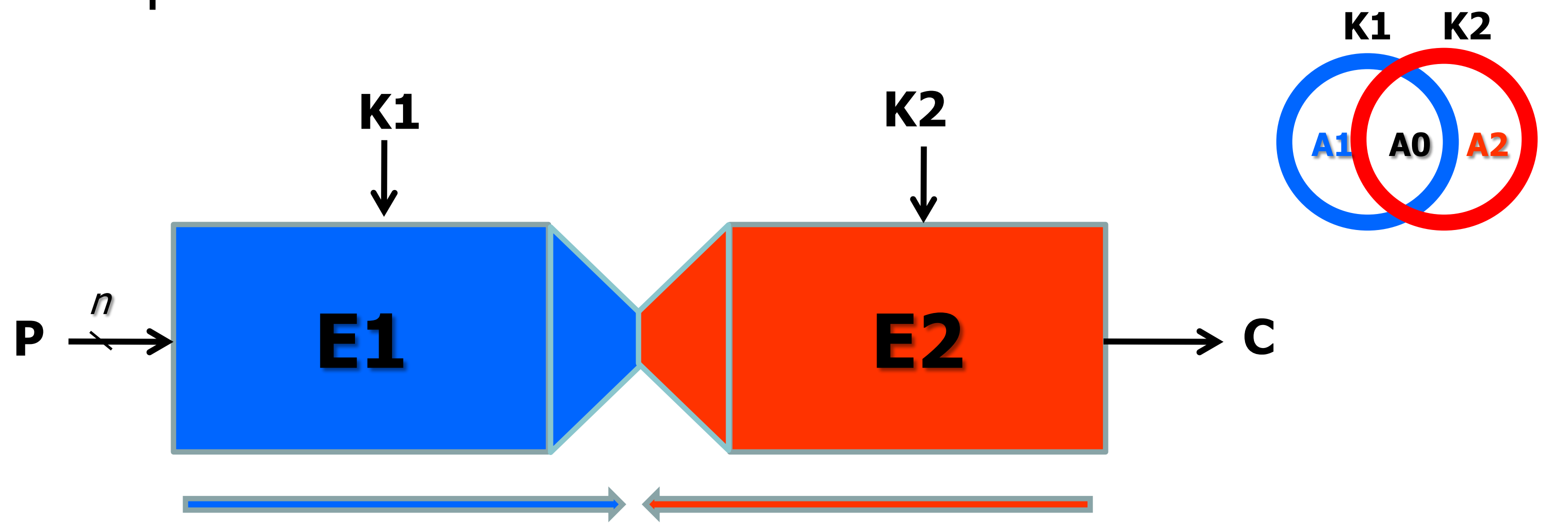
The Point of the attack :

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

Advanced techniques

Partial Matching [AS08]

- Match a part of state instead of full state



Advantage:

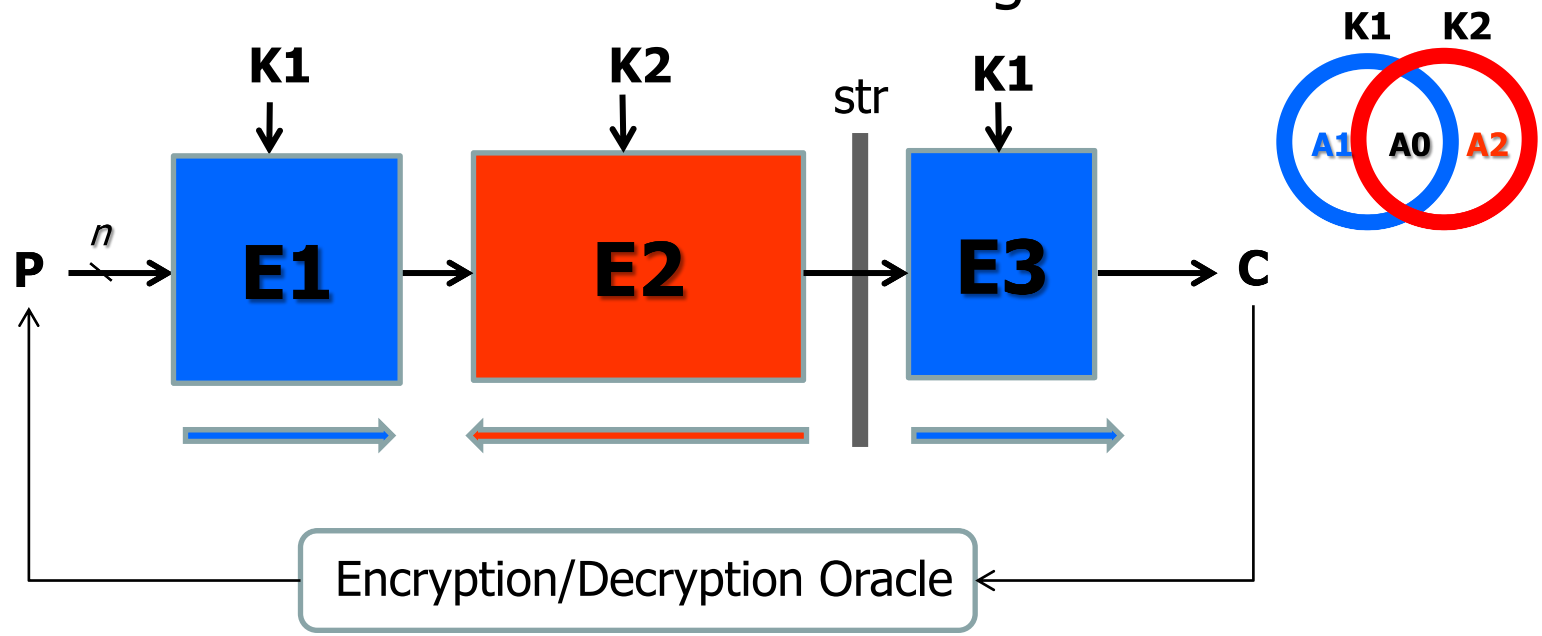
=> allow to omit key bits around matching state

Disadvantage:

=> decrease rate of rejected keys @matching state

Splice and Cut [AS08]

- regard the first and last round as contiguous rounds



Advantage:

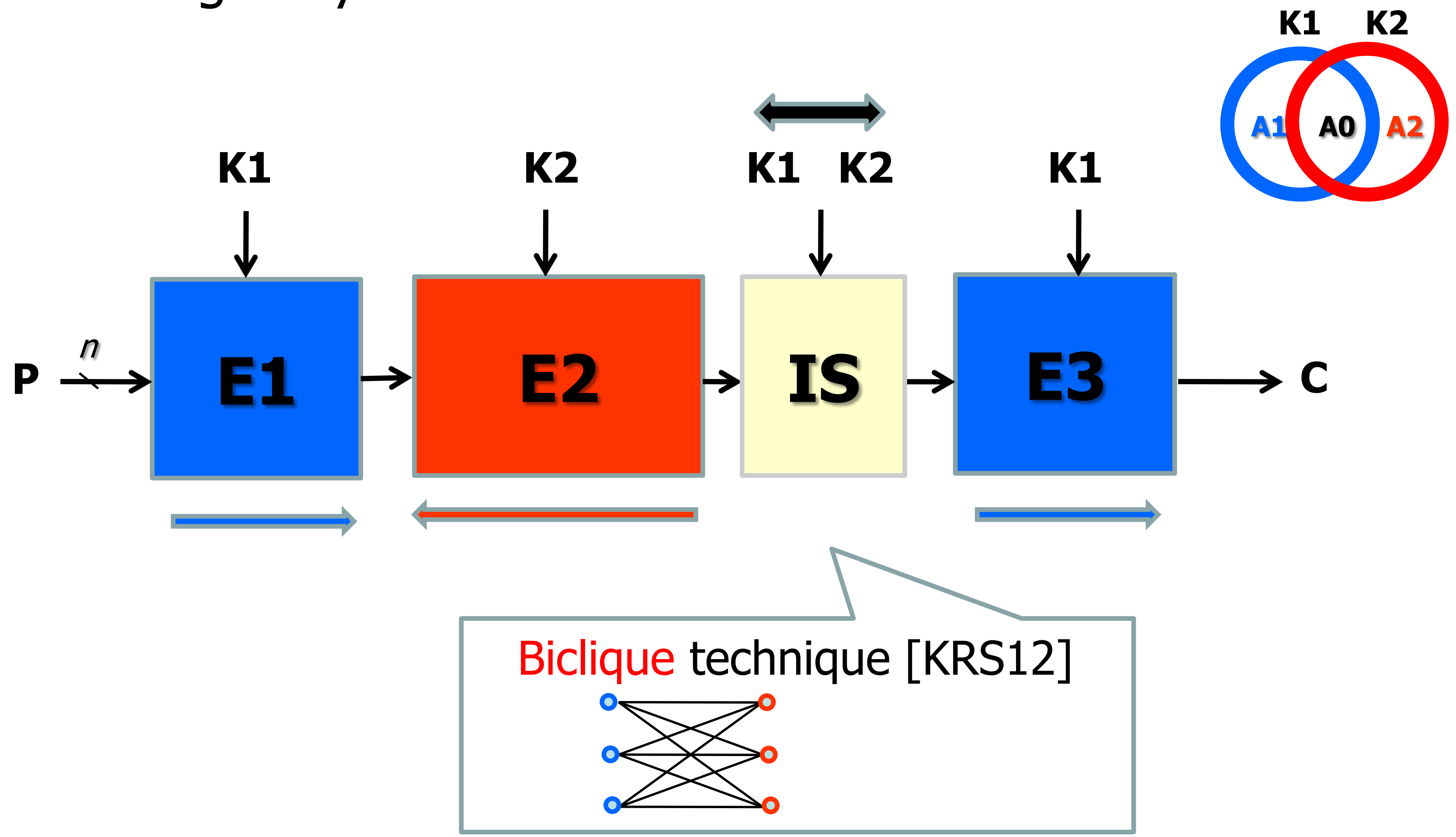
=> choose chucks freely similar to hash functions

Disadvantage:

=> increase required data complexity

Initial Structure [SA09]

- Exchange key bits around start state



2.MitM attack on Block Cipher having Simple KSF

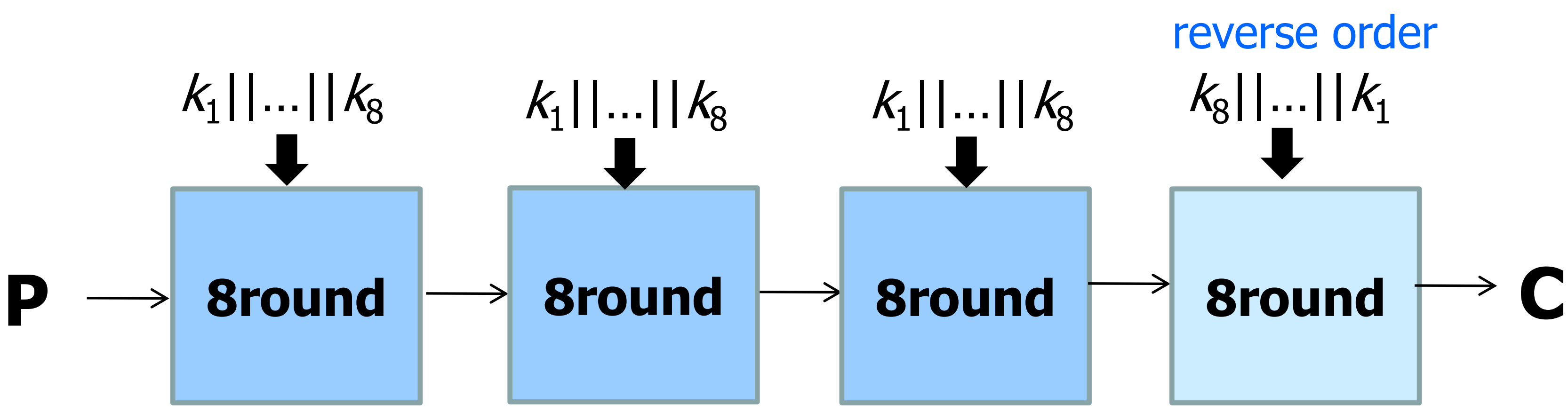
- XTEA, LED, Piccolo (ACISP 2012, w/ K.Shibutani)
- GOST (FSE 2011, JoC)

Simple Key Scheduling

- Simple Key Scheduling = Bit (word) Permutation based
  - Used in many lightweight Block ciphers
    - GOST, XTEA, HIGHT, LED, Piccolo

Ex : GOST block cipher

=>256 bit key is divided into eight 32 bit words s.t. $k_1||\ldots||k_8$



It is relatively easy to evaluate of security against MitM, because we can focus on only data processing part

Target Ciphers

- **XTEA (64-bit block, 128 bit key) [NW97]**
  - developed in 1997
  - Data processing part : Feistel

- **LED (64-bit block, 64-128 bit key) [GPPR11]**
  - Proposed @ CHES2011
  - Data processing part : SPN (AES base)

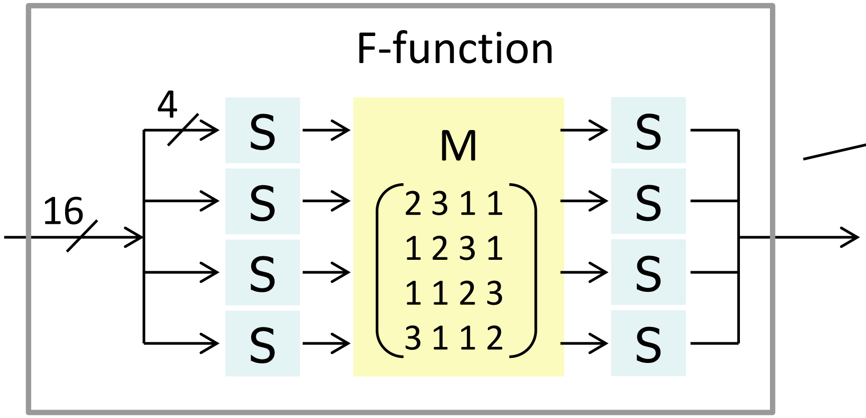
- **Piccolo (64-bit block, 80/128 bit key) [SIHMAS11]**
  - Proposed @ CHES2011
  - Data processing part : A variant of Generalized Feistel

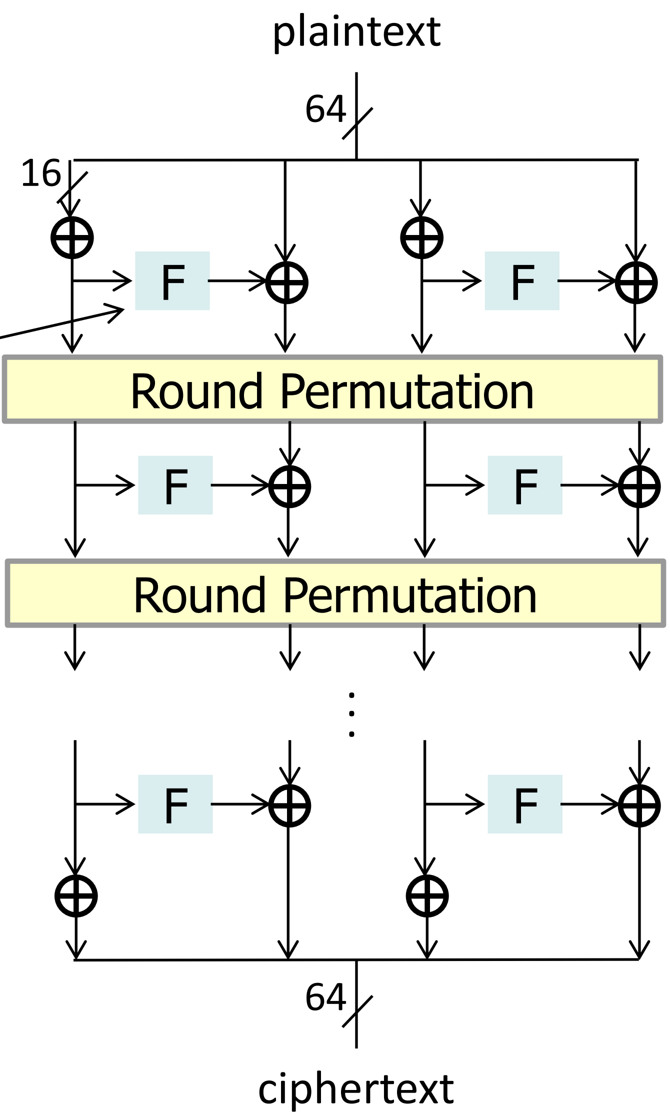
All of them employ simple key scheduling Function

Piccolo: Overall Structure

Data processing part:

a variant of generalized Feistel network





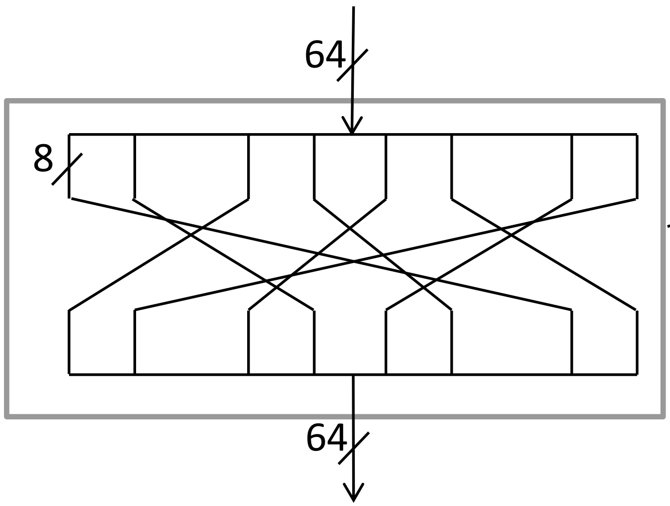
Keyless SPS-type F-function

#round : 25 (80-bit key), 31 (128-bit key)

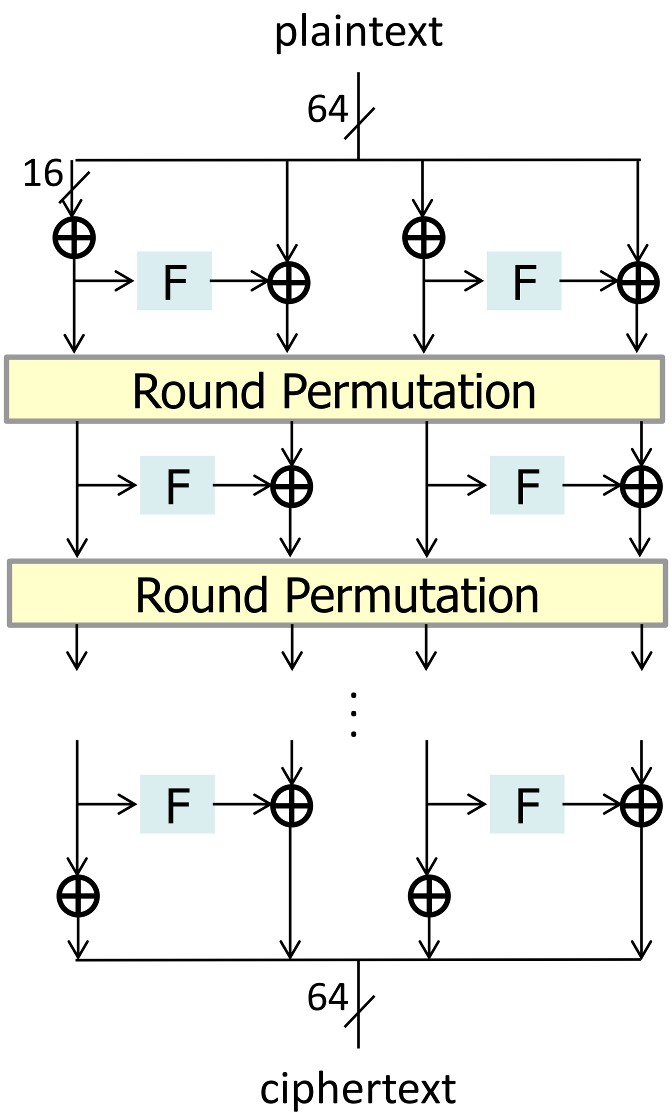
Piccolo: Overall Structure

Data processing part:

a variant of generalized Feistel network



Half-word based round permutation

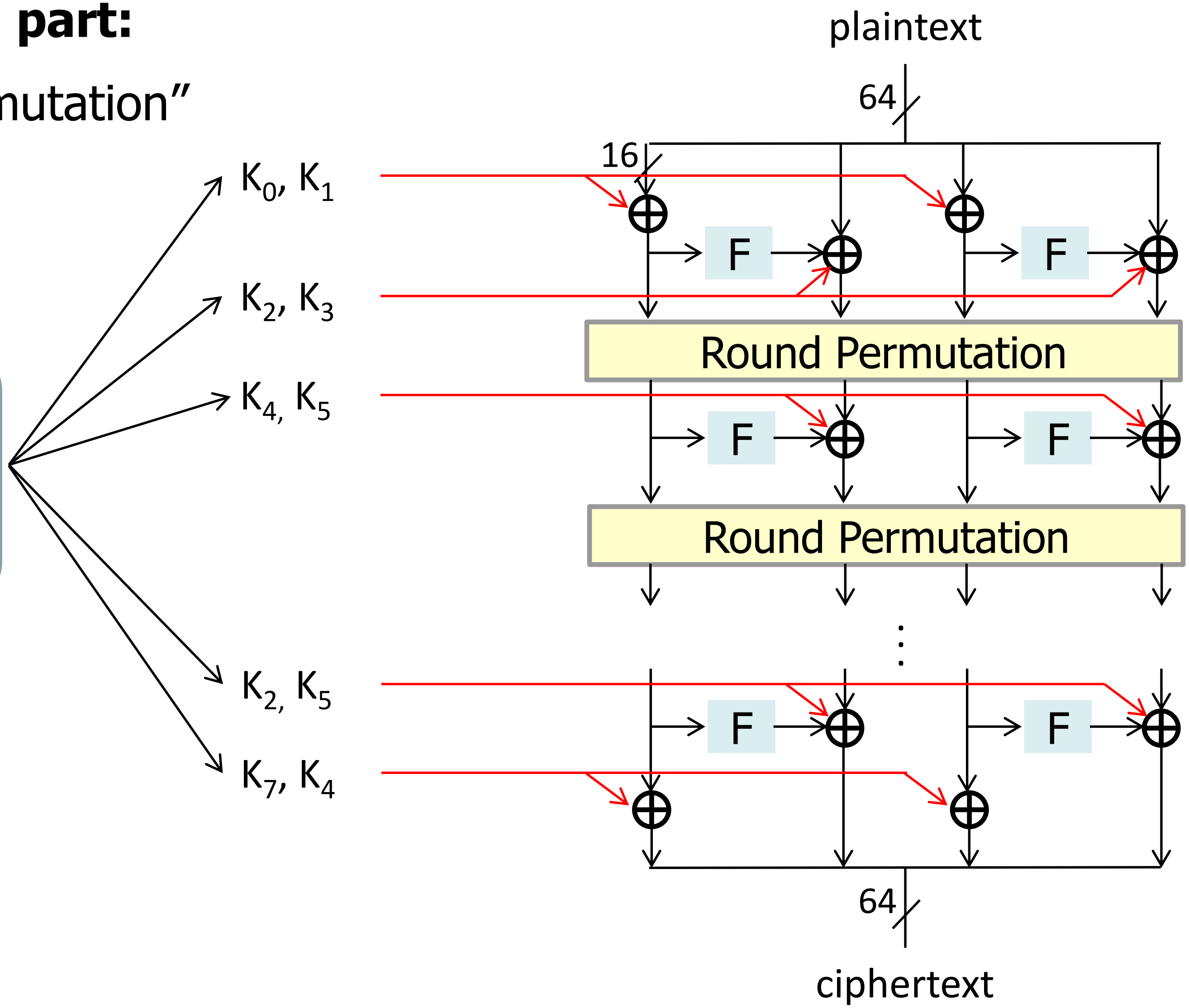


Piccolo: Overall Structure

Key scheduling part:

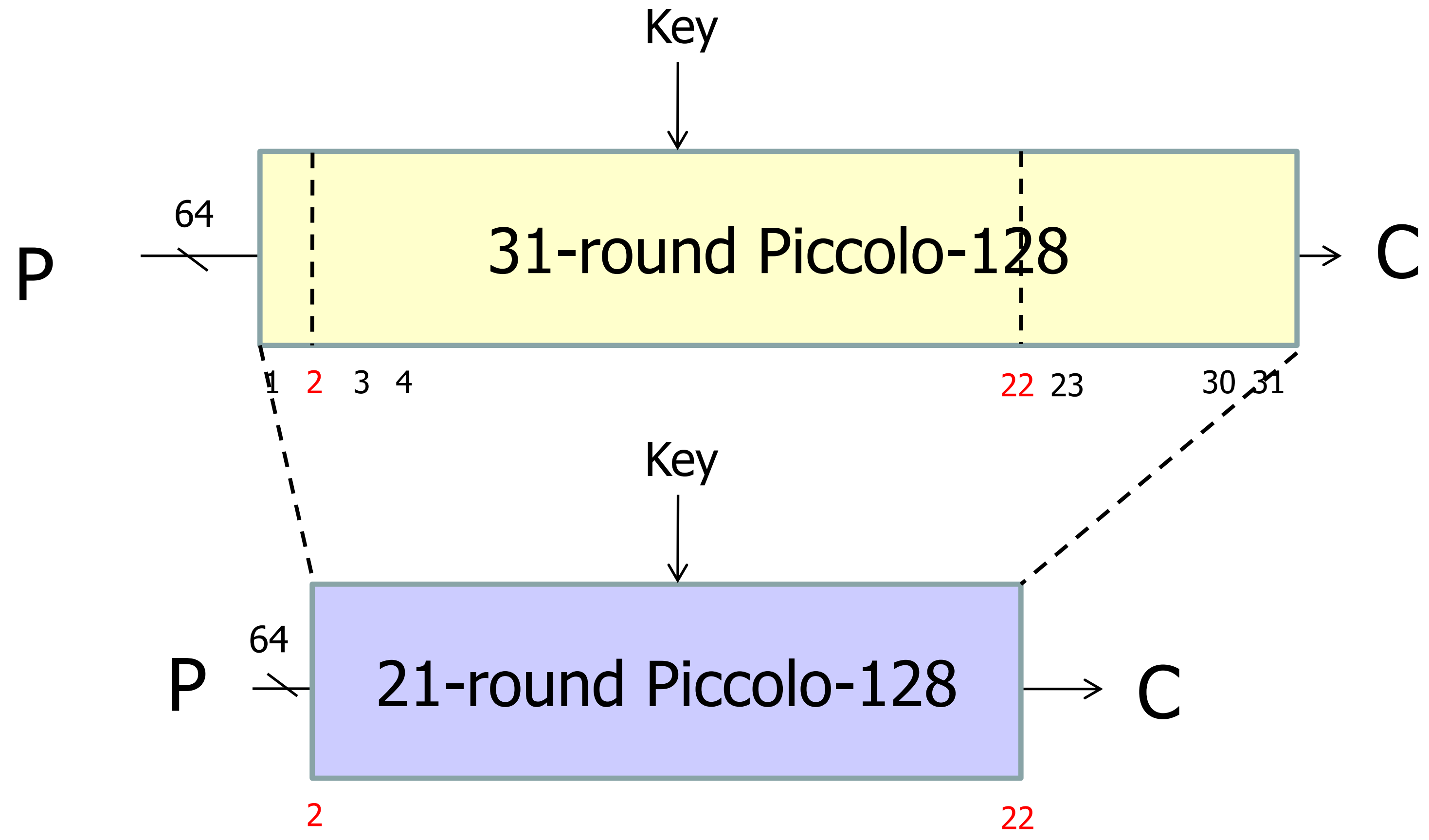
"16-bit word Permutation"

(128-bit key)
key =
$\{K_0 \mid K_1 \mid \ldots \mid K_6 \mid K_7\}$
$(|K_i| = 16\text{-bit})$



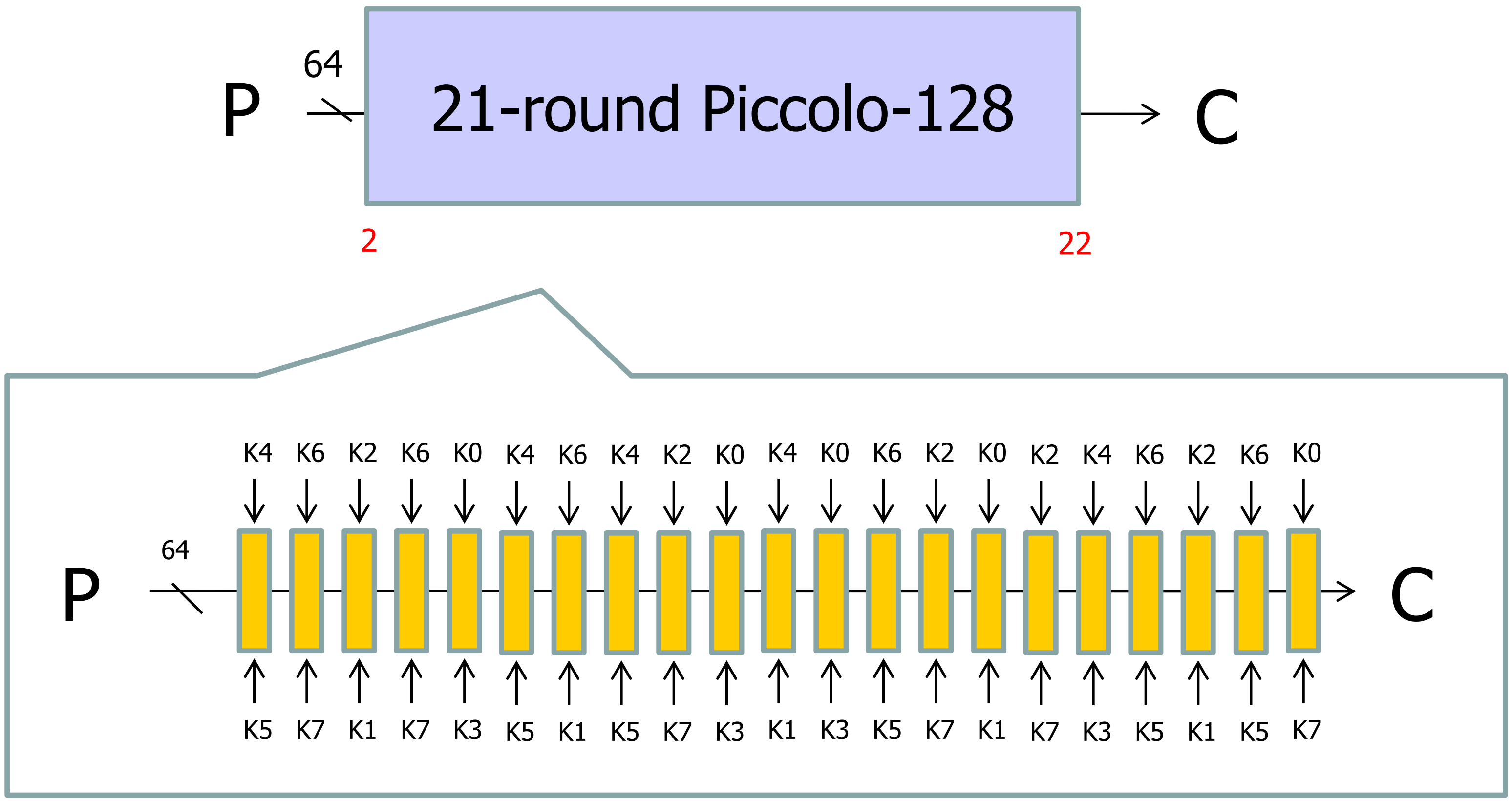
Target variant of Piccolo-128

- 21 round reduced Piccolo-128 (round 2-22)



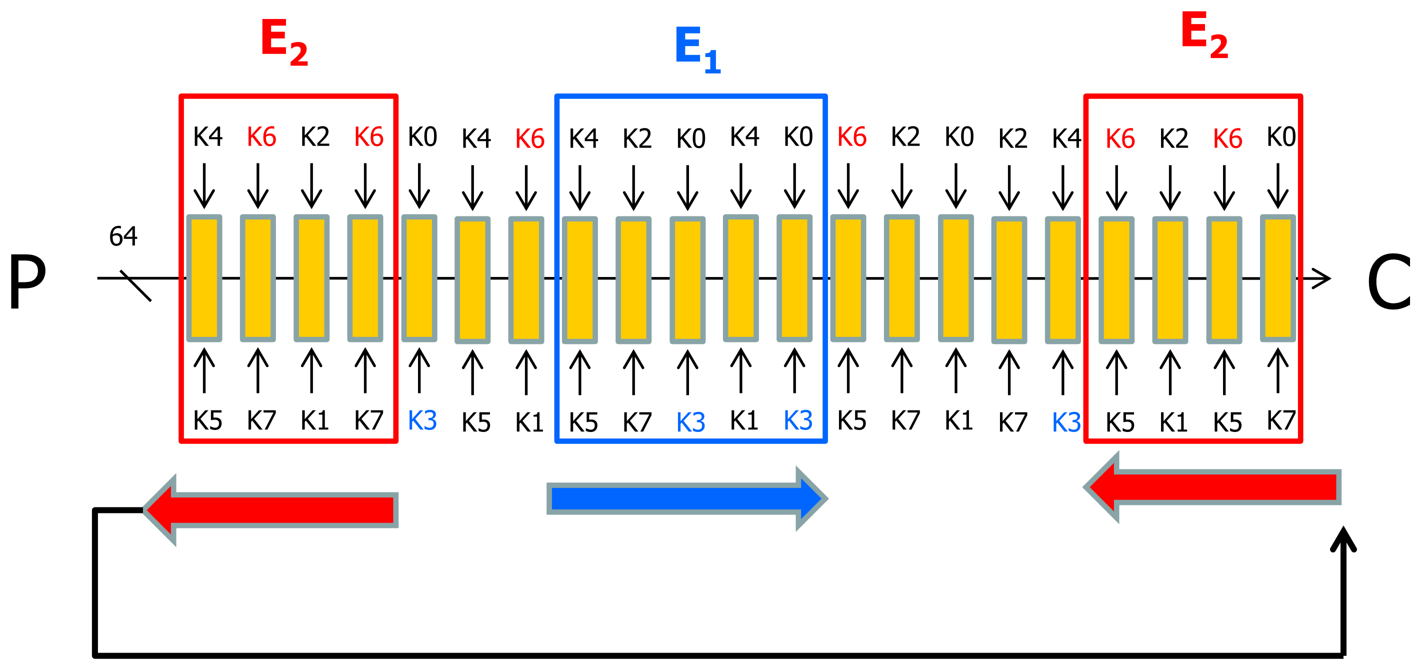
21 round Piccolo-128

- Piccolo's Key scheduling => word permutation



Attacking 21-round Piccolo-128

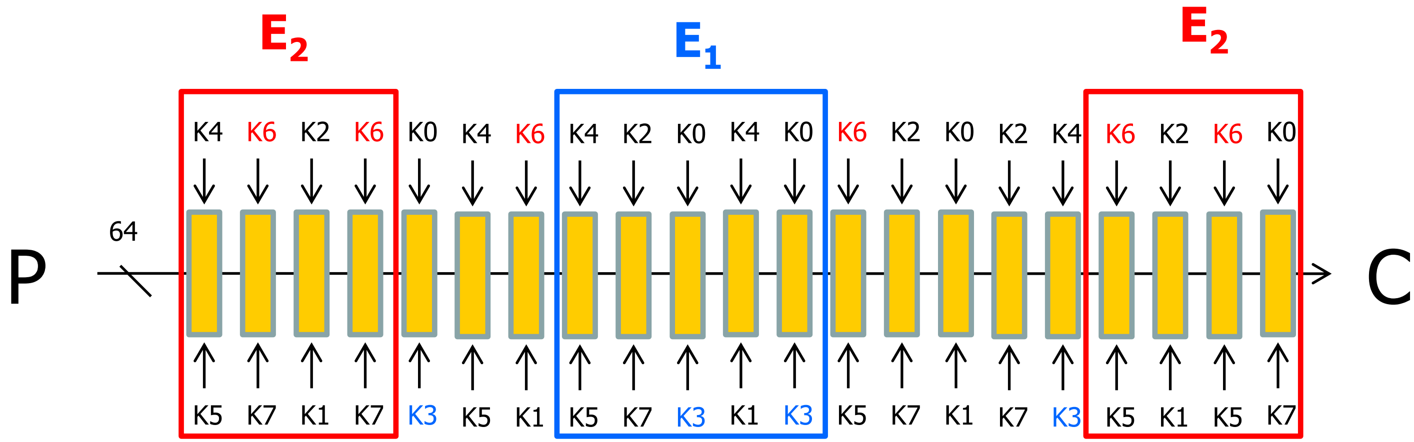
- Construct two chunks E_1 and E_2 by using Spice and Cut technique

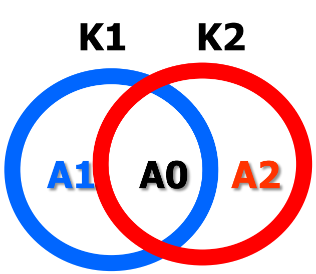


Spice and Cut technique

Attacking 21-round Piccolo-128

- Construct two chunks E_1 and E_2 by using Spice and Cut technique
  - Neutral word of E_1 : K3
  - Neutral word of E_2 : K6



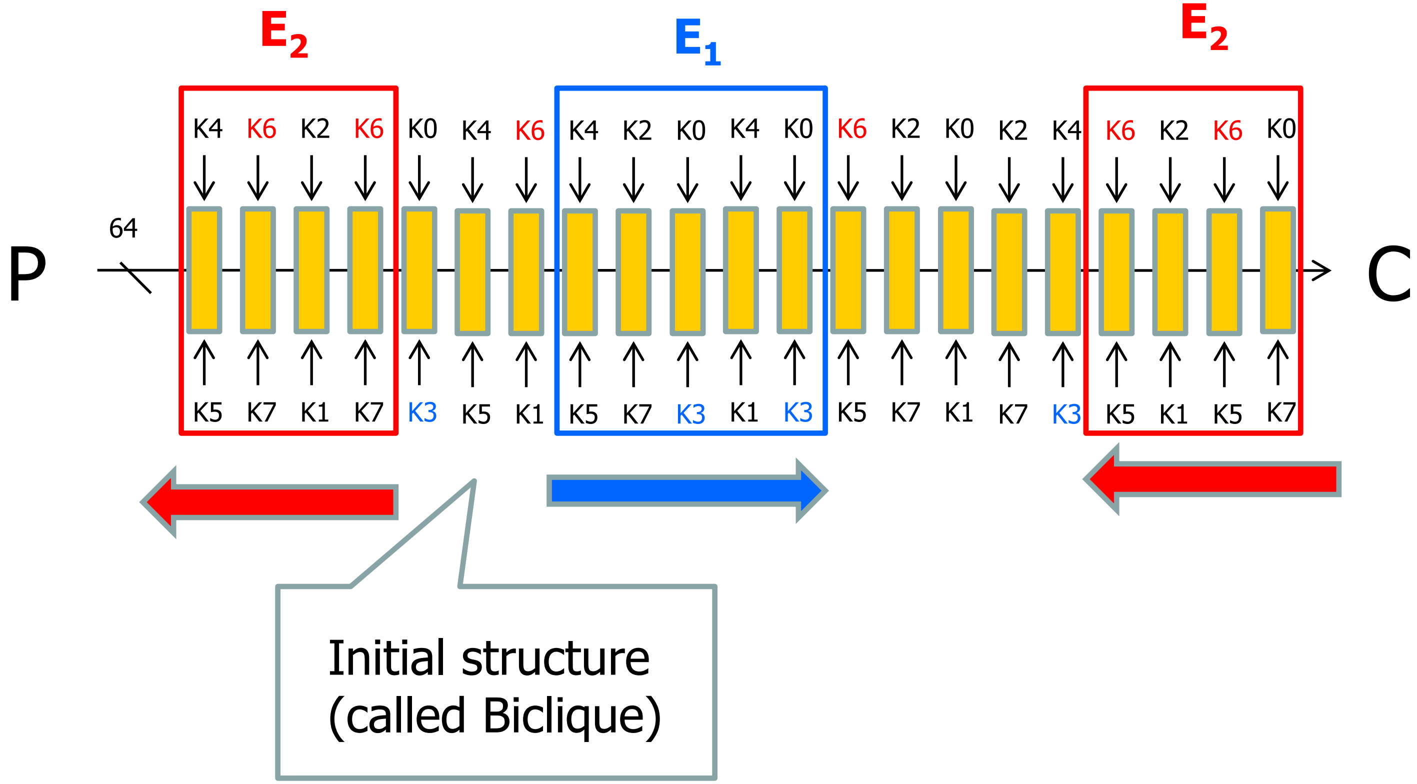


A0 = K1∩K2 = K0, K1, K2, K4, K5, K7
A1 = K1/(K1∩K2) = K3
A2 = K2/(K1∩K2) = K6

|A0| = 112, |A1| = |A2| = 8

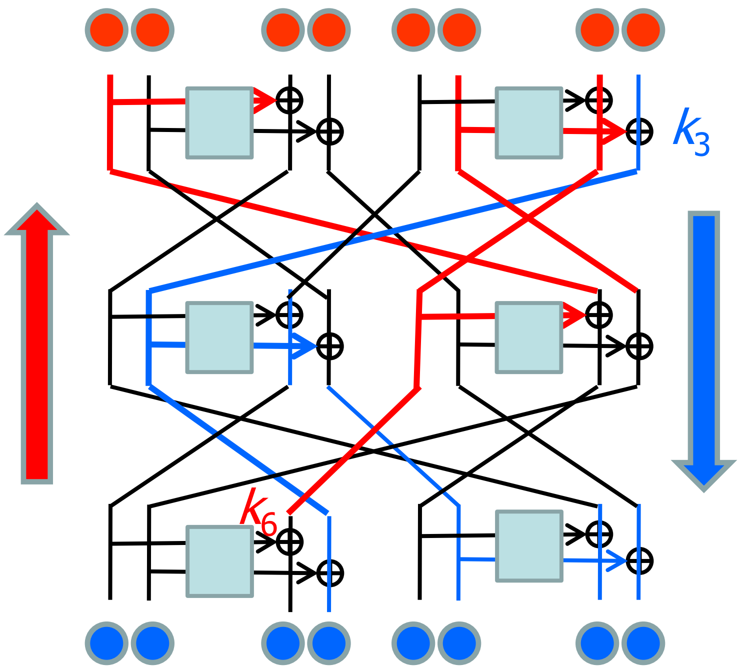
Attacking 21-round Piccolo-128

- Construct two chunks E_1 and E_2 by using Spice and Cut technique
  - Neutral word of E_1 : K3
  - Neutral word of E_2 : K6

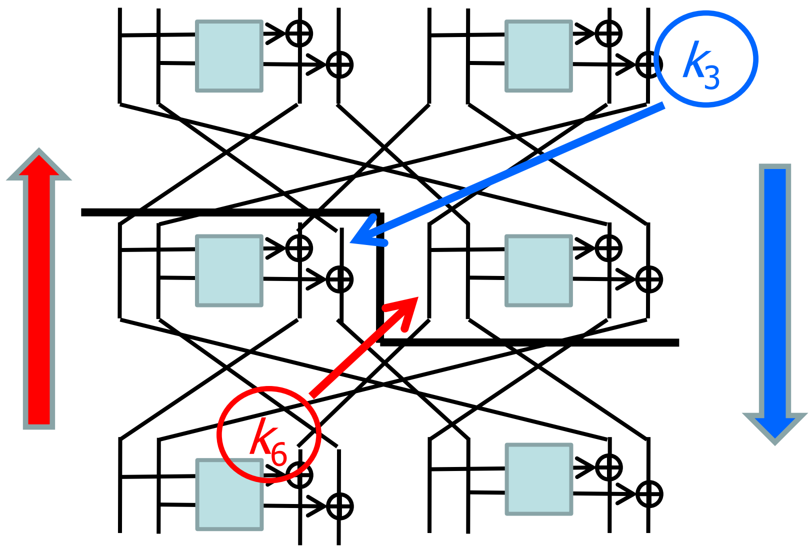


Initial Structure (biclique)

- K3 and K6 are exchangeable/movable
  - These differential trails do not share nonlinear component (formally called Biclique)

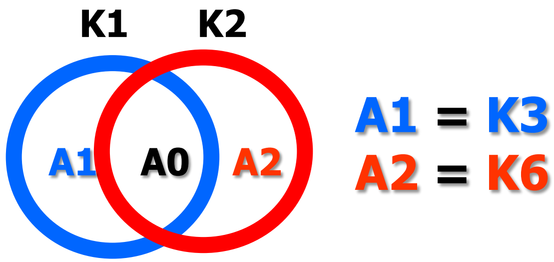


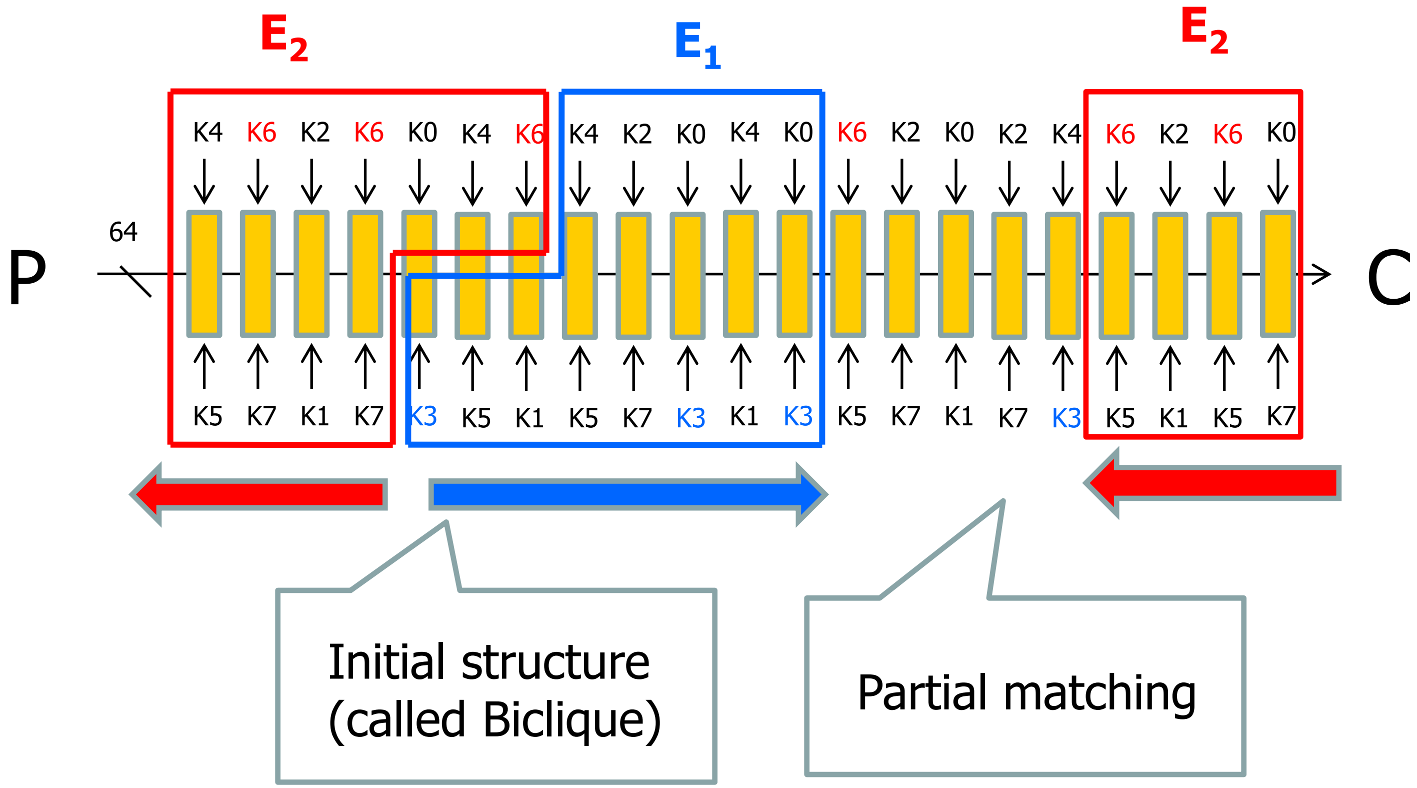
Do not share nonlinear function



Attacking 21-round Piccolo-128

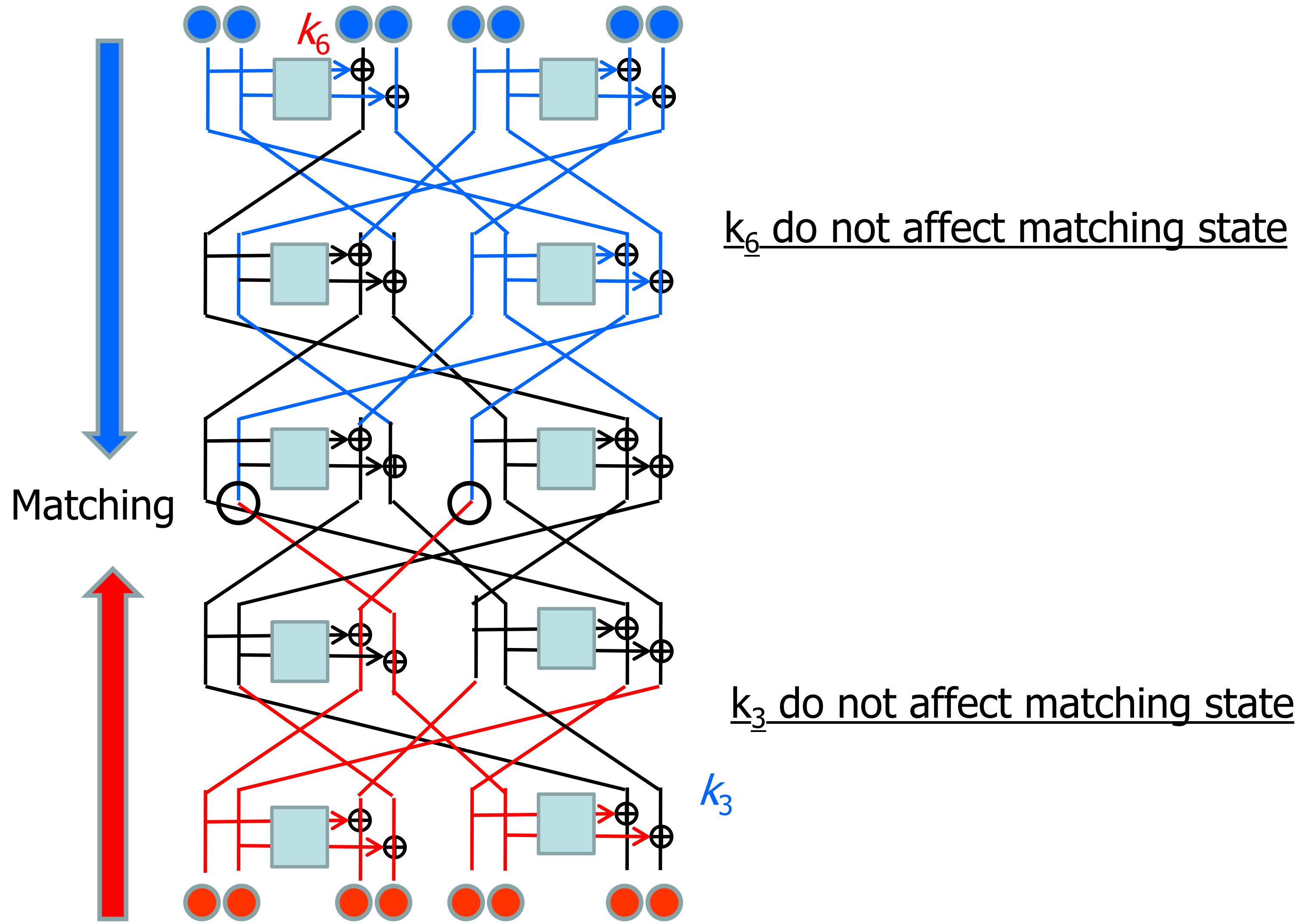
- Neutral word of forward process : K3
- Neutral word of backward process : K6





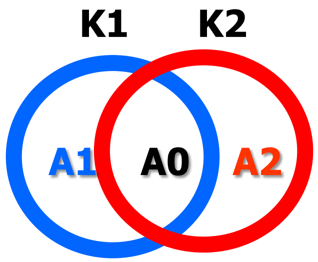
Partial Matching

- some key bits around the matching point can be omitted.

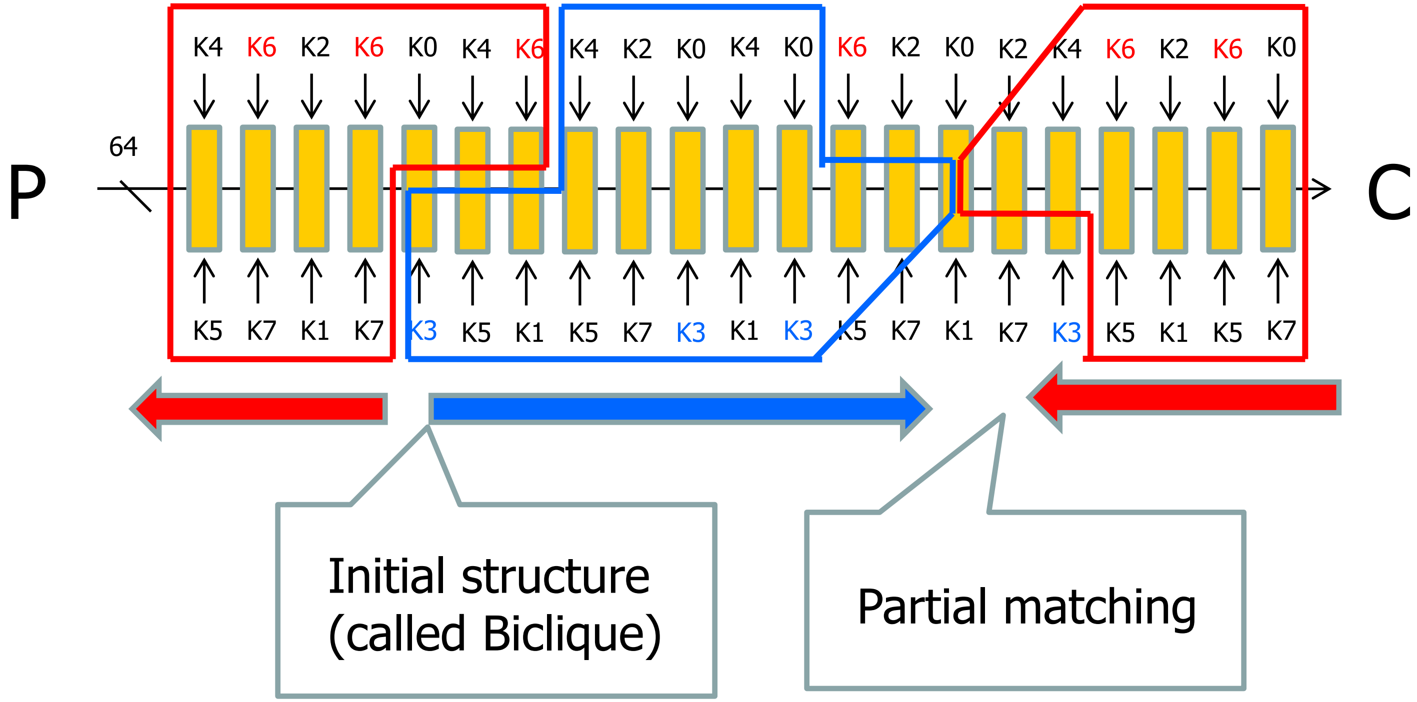


Attacking 21-round Piccolo-128

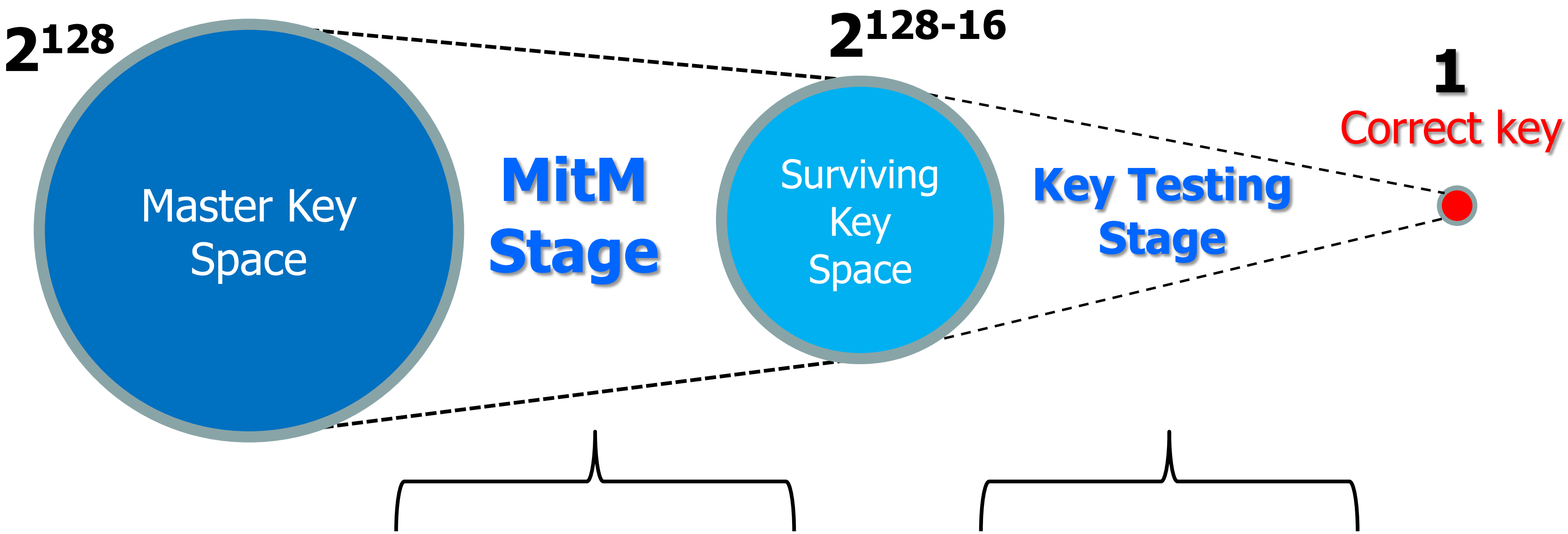
- Neutral word of forward process : K3
- Neutral word of backward process : K6



A1 = K3
A2 = K6



Evaluation



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

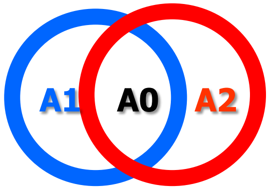
Data = max (1 , l/b)

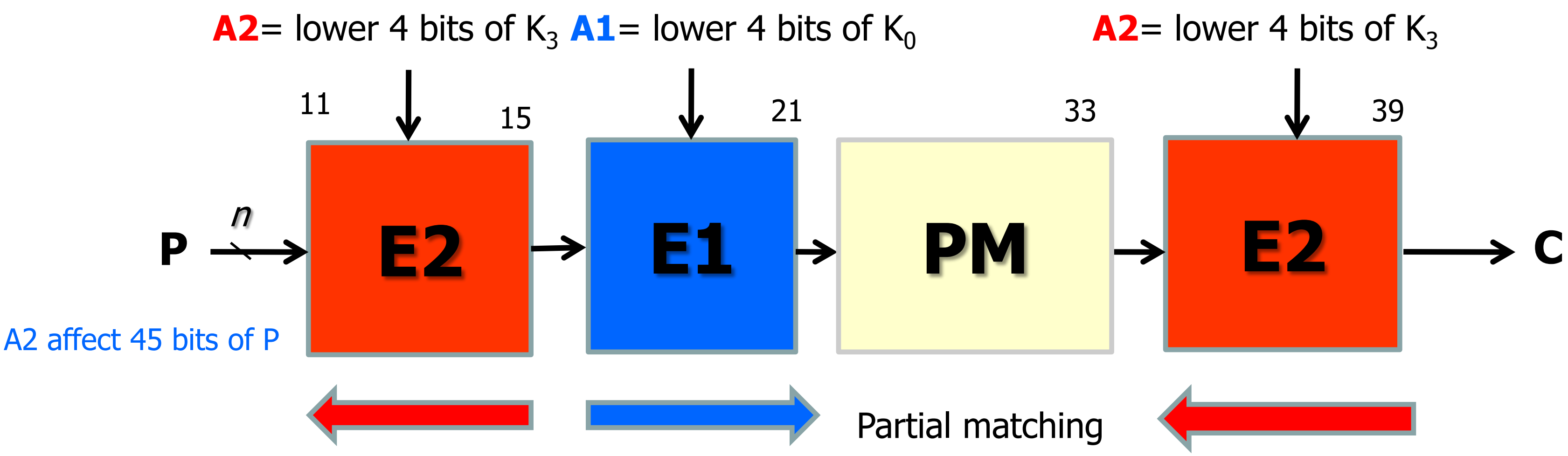
- 21-round Attack on Piccolo-128 $|A0| = 112, |A1| = |A2| = 8$
  - time complexity : $2^{112}(2^8+2^8) + 2^{112} = 2^{121}$
  - Data : 2^{64} (code book)
  - memory 2^8

29-round XTEA Attack

- round 11 to 39 (29 round)
  - Neutral word of forward process : K0
  - Neutral word of backward process : K3

128 bit key = $K_0 || .. || K_3$



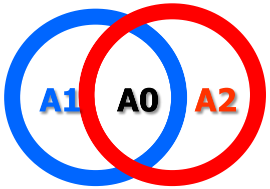


- Evaluation of 29-round Attack on XTEA-128
  - Time complexity : $2^{120} (2^4+2^4) + 2^{124} = 2^{124}$
  - Data : 2^{45} KP
  - Memory 2^4

$|A0| = 120, |A1| = |A2| = 4$

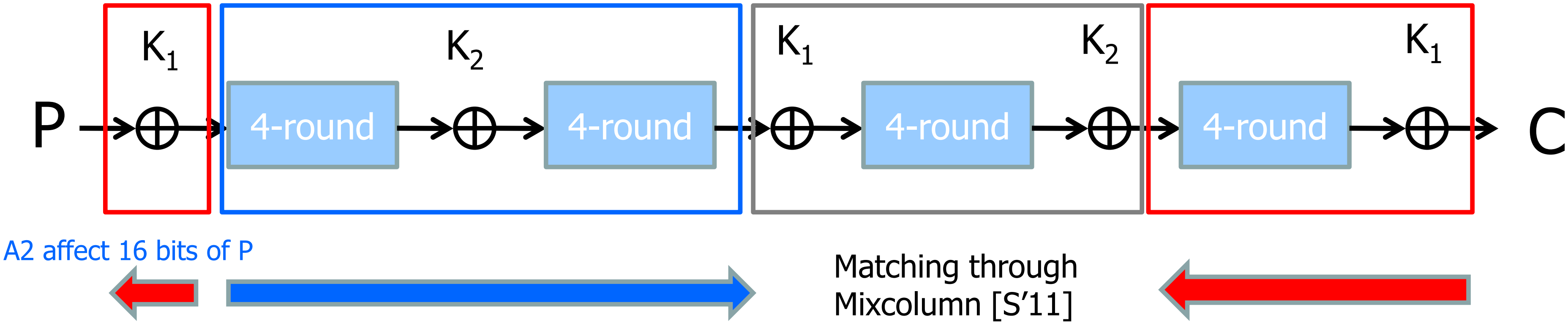
16 round LED-128 Attack

- round 1 to 16 (16 round)



128 bit key = $K_1 || K_2$

A2= lower 16 bits of K_2 **A1**= lower 16 bits of K_1



- Evaluation of 16-round Attack on LED-128
  - Time complexity : $2^{96} (2^{16}+2^{16}) + 2^{96} = 2^{112}$
  - Data : 2^{16} KP
  - Memory 2^{16}

$|A0| = 96, |A1| = |A2| = 16$

Results

- MitM attack on Block Cipher having Simple KSF
  - Update best attack of target ciphers

Algorithm	#Full round	Type of Attack	#attacked round	Paper
XTEA	64	Meet-in-the-Middle	23	[SMWP11]
		Impossible differential	23	[CWP12]
		zero correlation Linear	27	[BW12]
		Meet in the Middle	28	[SWS+12]
		Meet in the Middle	29	Our
LED-64	32	Differential/Linear	8	[GPPR11]
		Meet in the Middle	8	Our
LED-128	48	Differential/Linear	8	[GPPR11]
		Meet in the Middle	16	Our
Piccolo-64	25	Differential/Linear	9	[SIH+11]
		Meet in the Middle	14	Our
Piccolo-128	31	Differential/Linear	9	[SIH+11]
		Meet in the Middle	21	Our

2.MitM attack on Block Cipher having Simple KSF

- XTEA, LED, Piccolo (ACISP 2012, w/ K.Shibutani)
- GOST (FSE 2011, JoC)

GOST Bloch Cipher

GOST Block Cipher

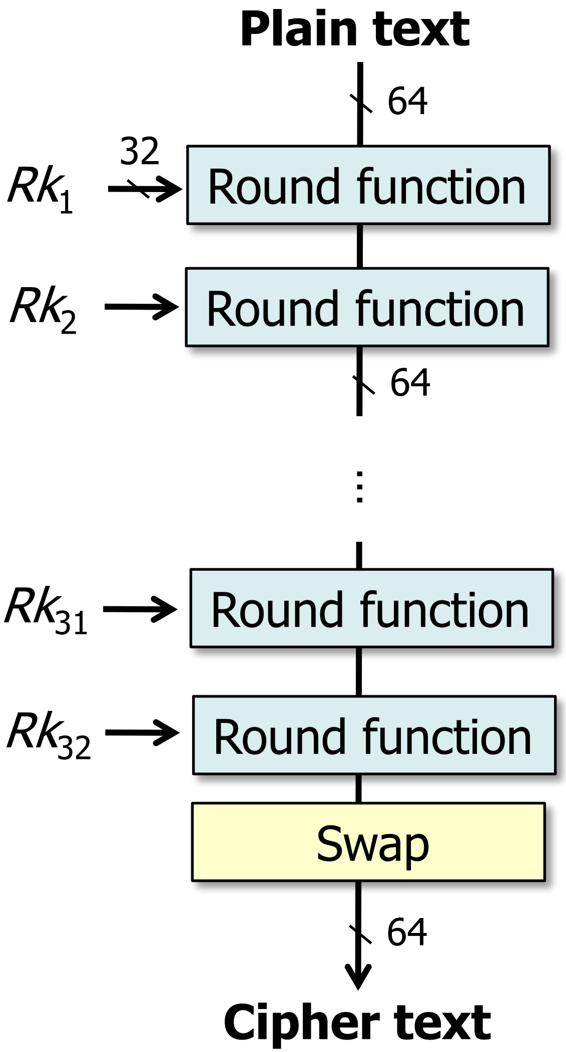
- Soviet Encryption Standard “GOST 28147-89”.
- Standardized in 1989 as the Russian Encryption Standard.
- Called *Russian DES* .
- No Single key attack on Full round GOST until 2011.

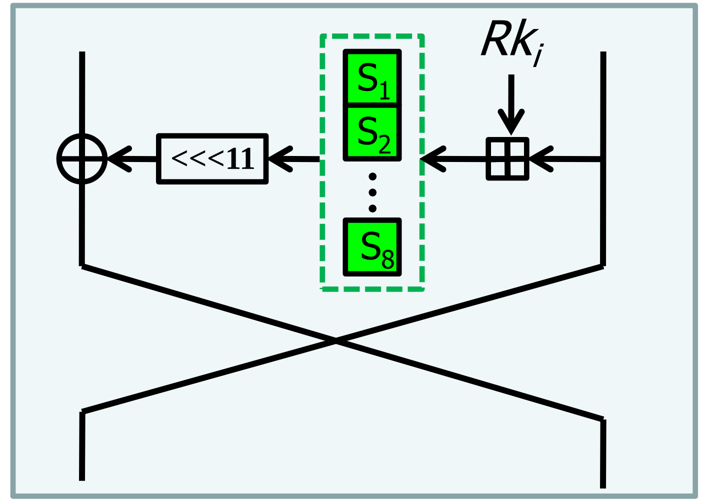
Implementation Aspect

- A. Poschmann et.al. show the 650-GE H/W implementation @CHES 2010
- Considered as Ultra light weight Block cipher.

Structure of GOST

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





Key schedule : 32-bit word permutation

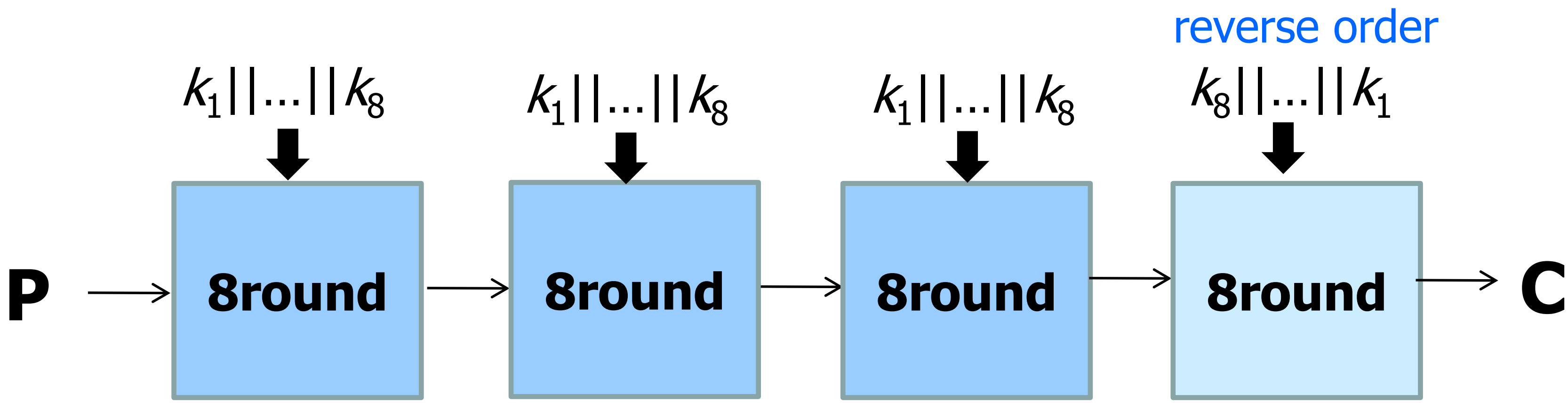
$$Master\ key = K_1 || K_2 || \ldots || K_8$$

256 bit   32 bit × 8

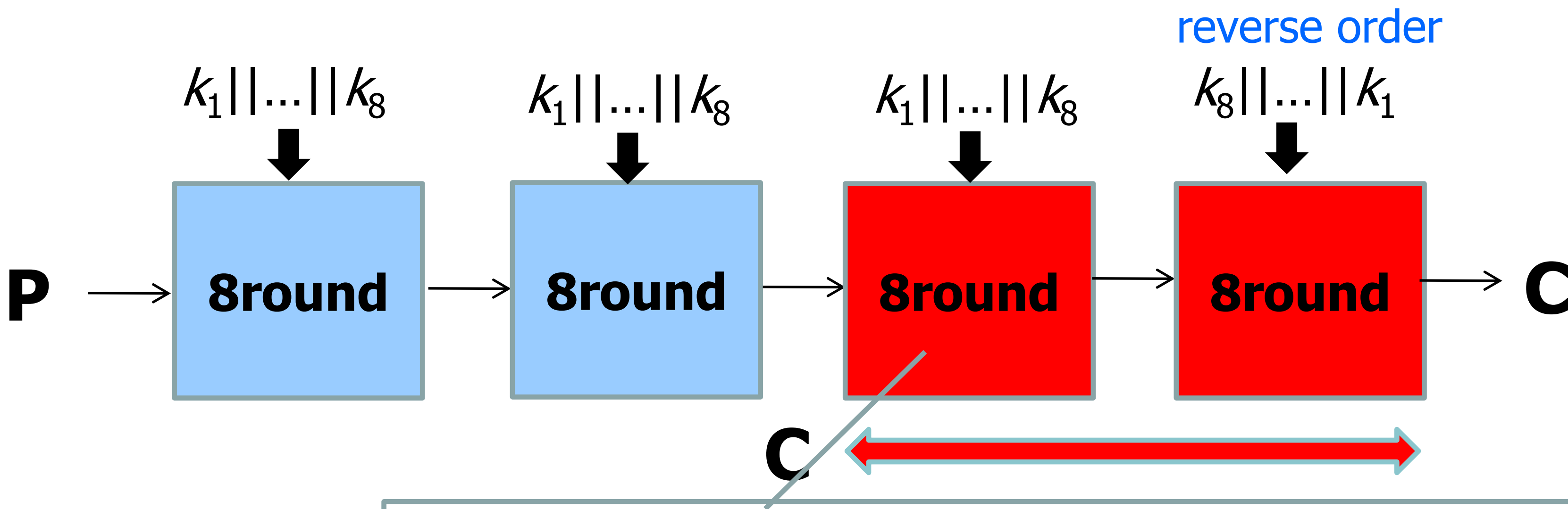
Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Key	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8

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

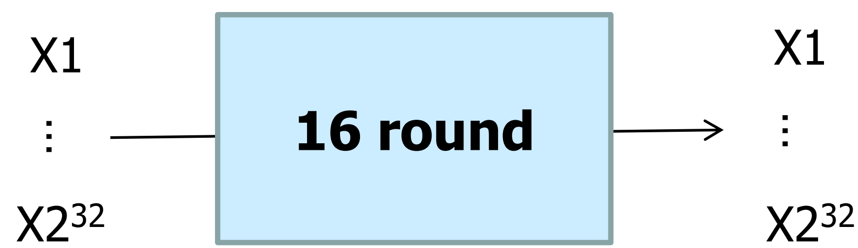
Application to Full GOST



Reflection Property [KM07]

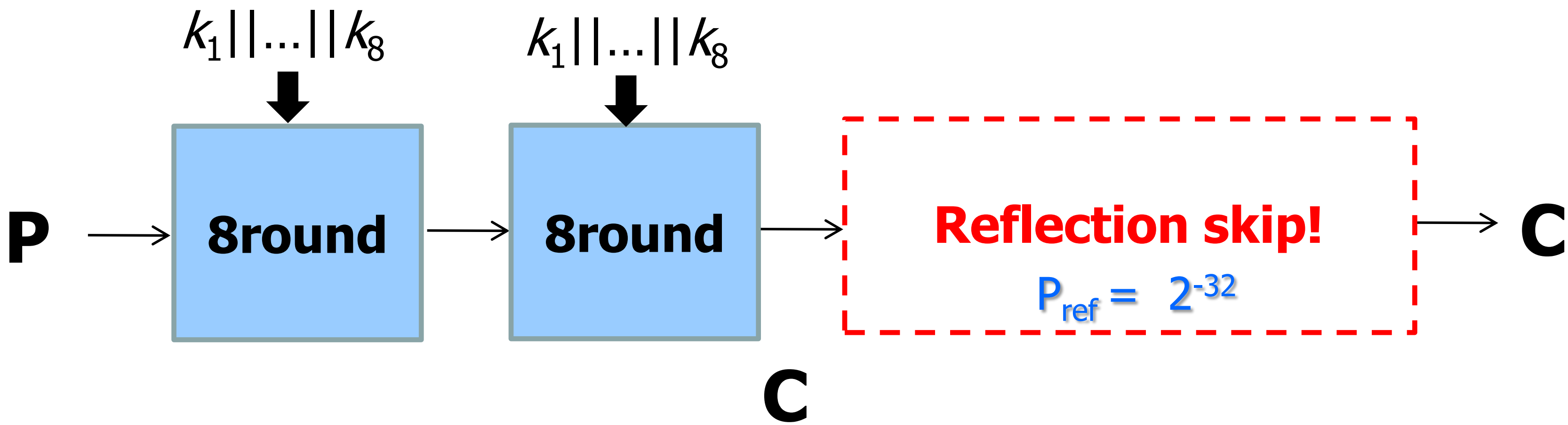


GOST's Reflection property was shown by Kara [K'08].
- # of fixed points of last 16 round is 2^{32}



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

Reflection Skip



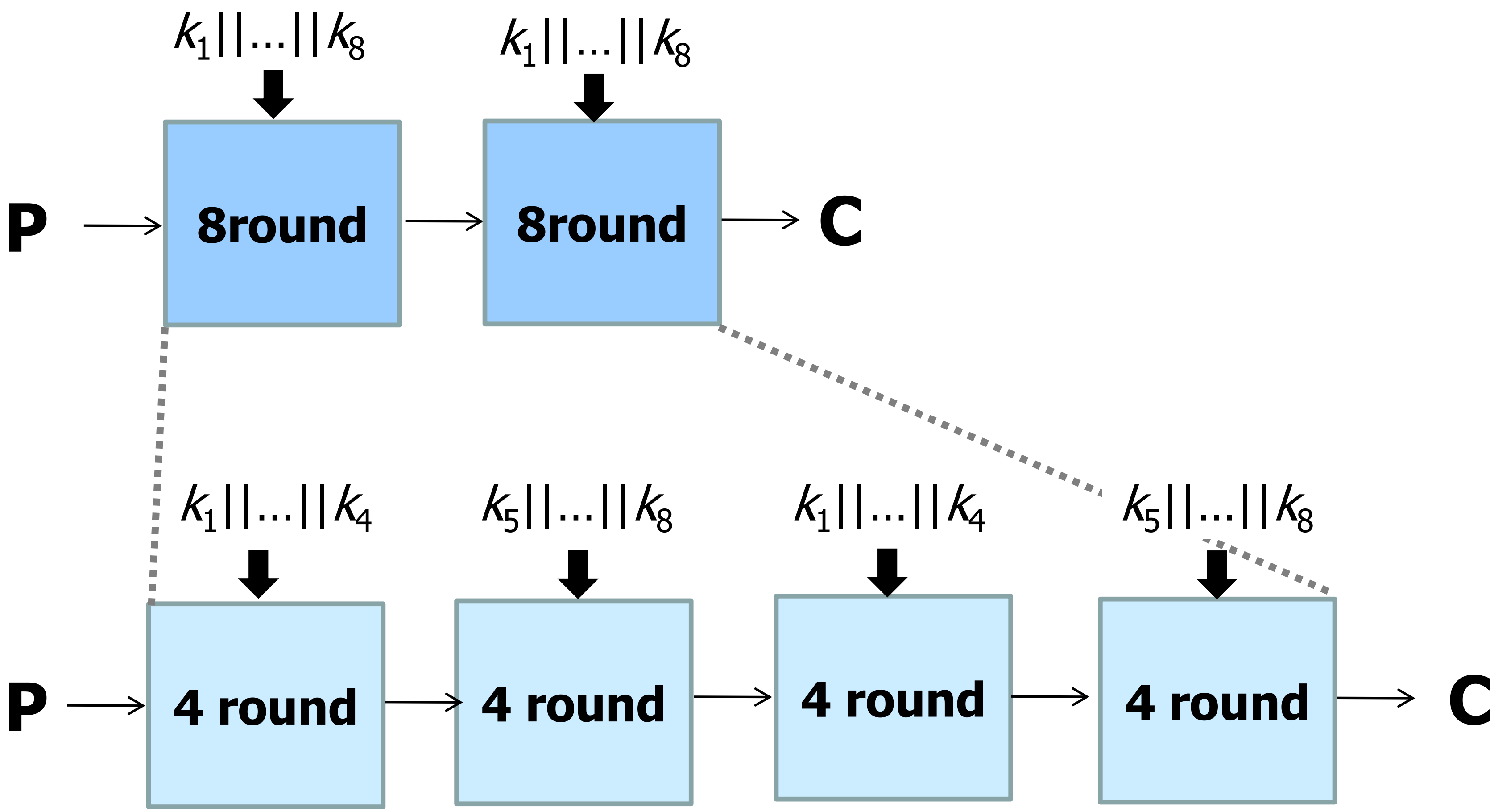
@Data collection stage:

Collect 2^{32} known plaintext/ciphertext pairs

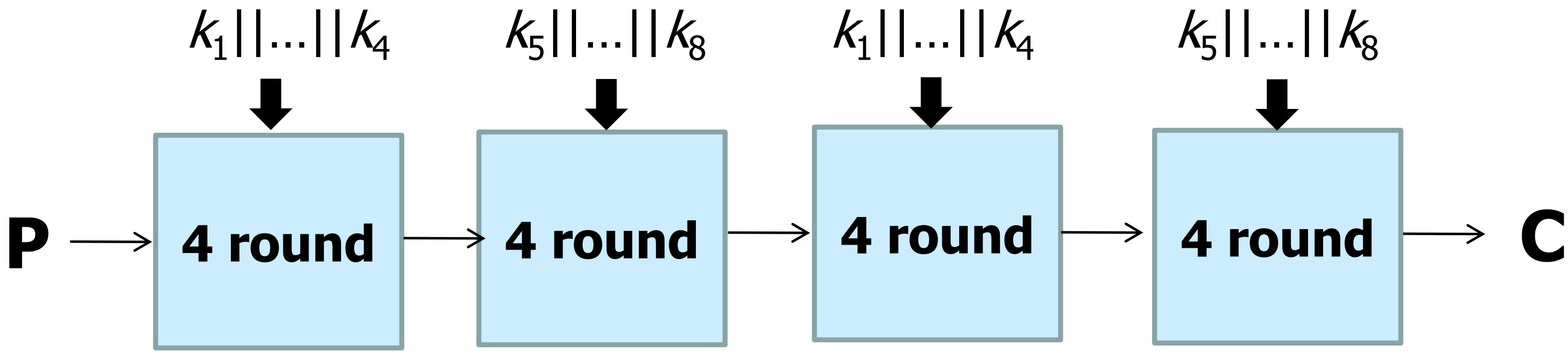
=> There is one pair in which reflection skip occur

R-MITM Stage

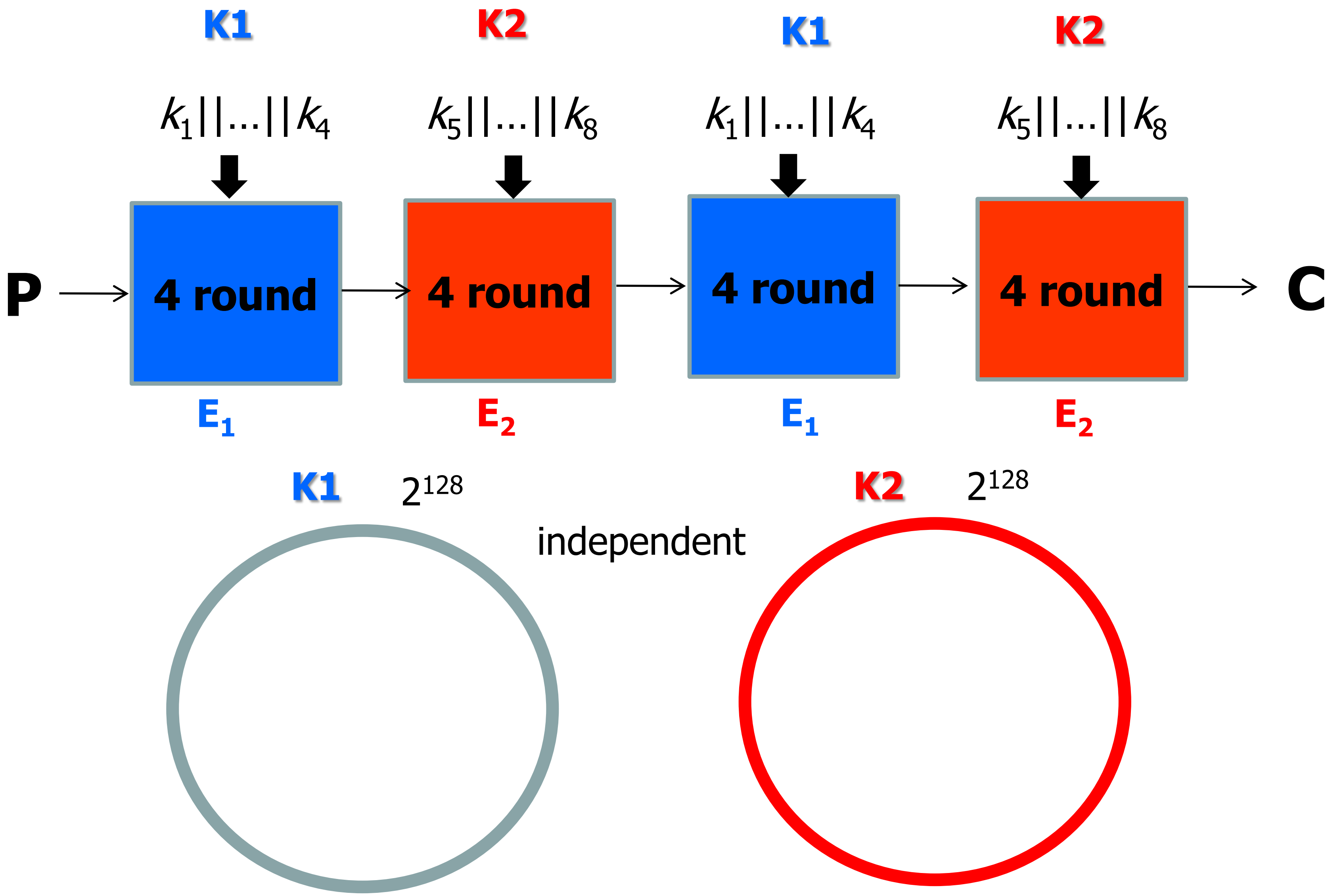
- For all 2^{32} Plaintext/Ciphertext, we mount MitM approach, assuming that the reflection skip occurs.



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
E_1
E_2
E_1
E_2
K1
2^{128}
independent
K2
2^{128}

MITM Stage

K1 K2 K1 K2

$k_1||\dots||k_4$ $k_5||\dots||k_8$ $k_1||\dots||k_4$ $k_5||\dots||k_8$

P → C

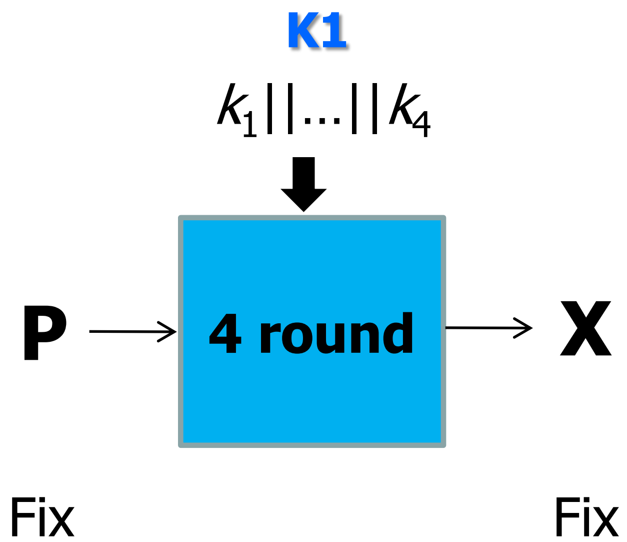
In the straightforward method,
It is difficult to mount the MITM attack.

Because there are 4 chunks.

Equivalent-key technique

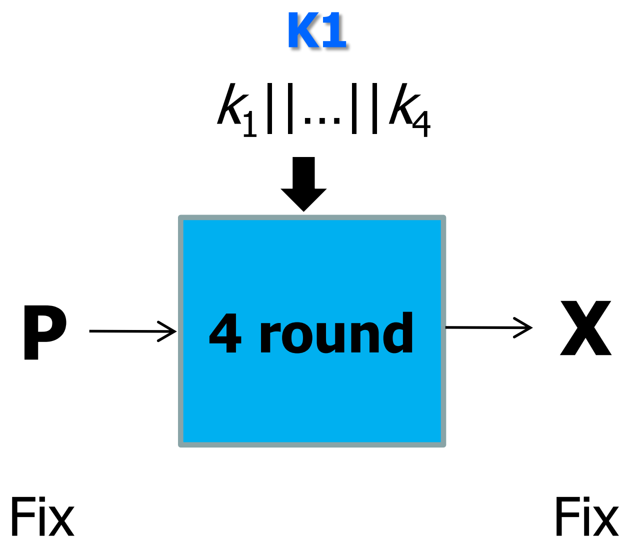
Equivalent Keys

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

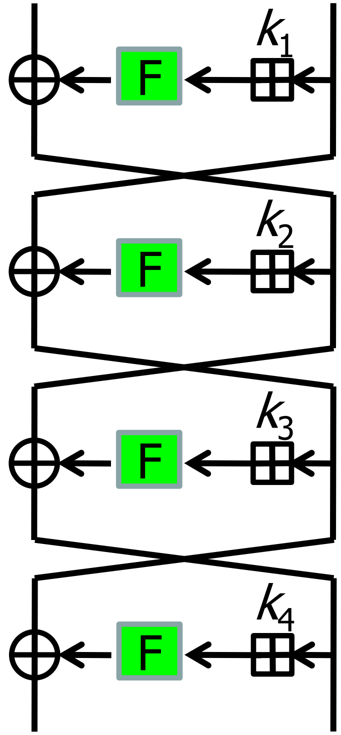


Equivalent Keys

- 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 (fixed) P and X, It is easy to find such set.



k_1 : Guess
32 bit

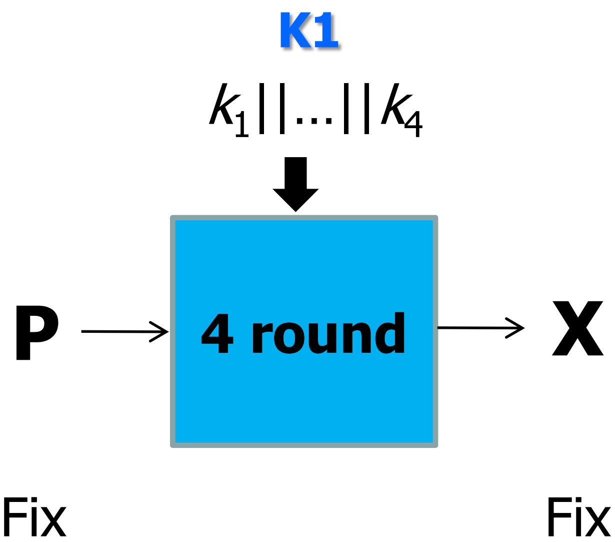
k_2 : Guess
32 bit

k_3 : Determine
32 bit

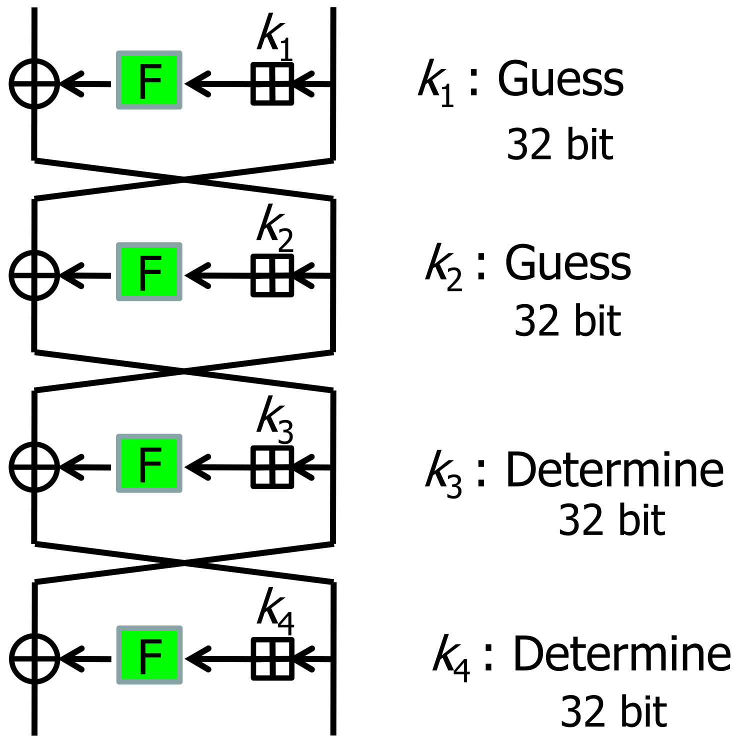
k_4 : Determine
32 bit

Equivalent Keys

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



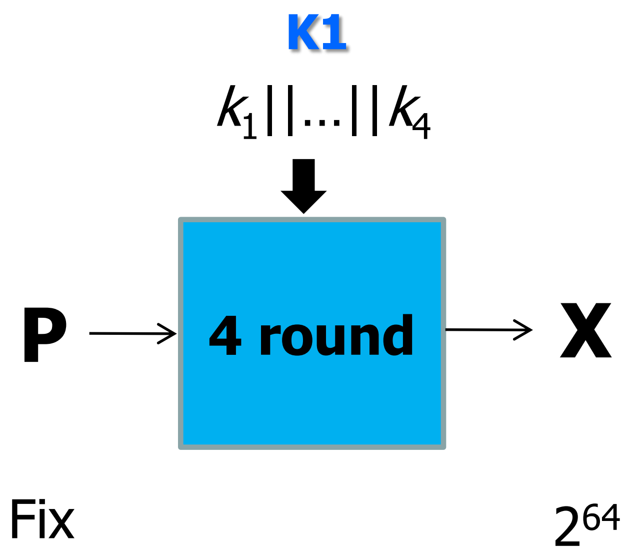
Given the values of (fixed) P and X,
It is easy to find such set.

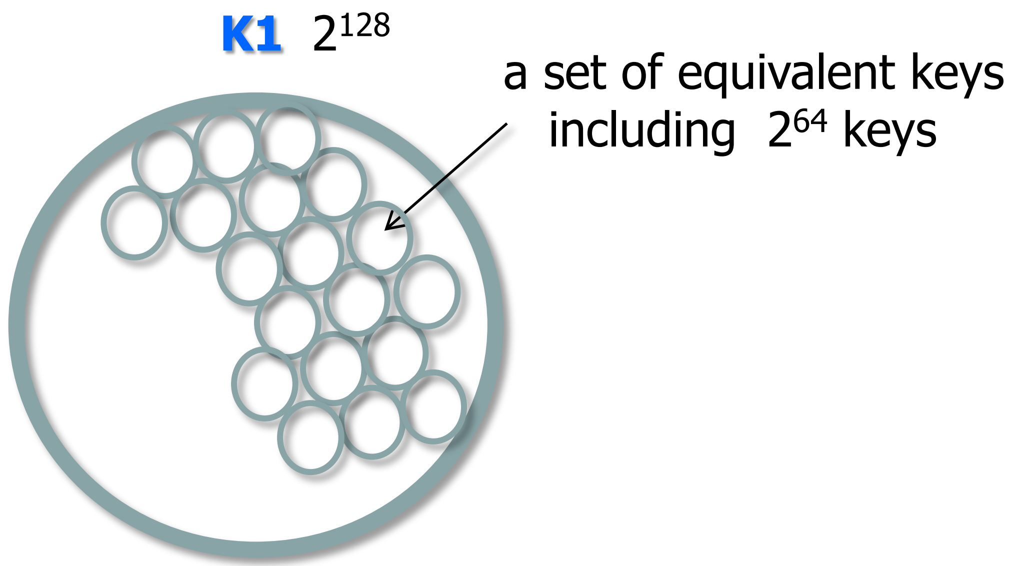


For each (P, X) pair,
there are 2^{64} such equivalent keys

Equivalent Keys

- Categorize K1 into sets of equivalent keys depending on X, where P is fixed one value and X has 2^{64} values

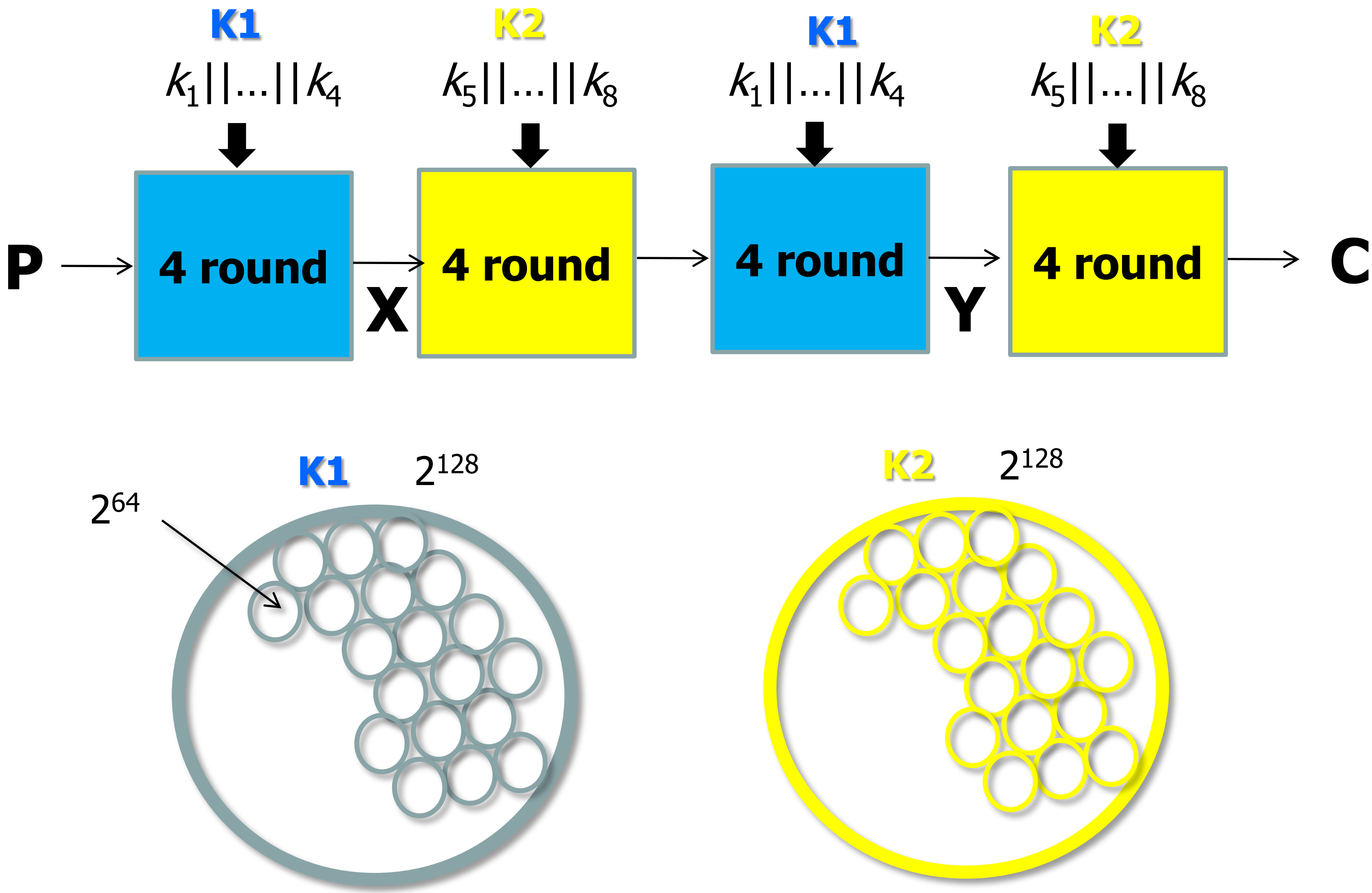




2^{64} sets of 2^{64} keys

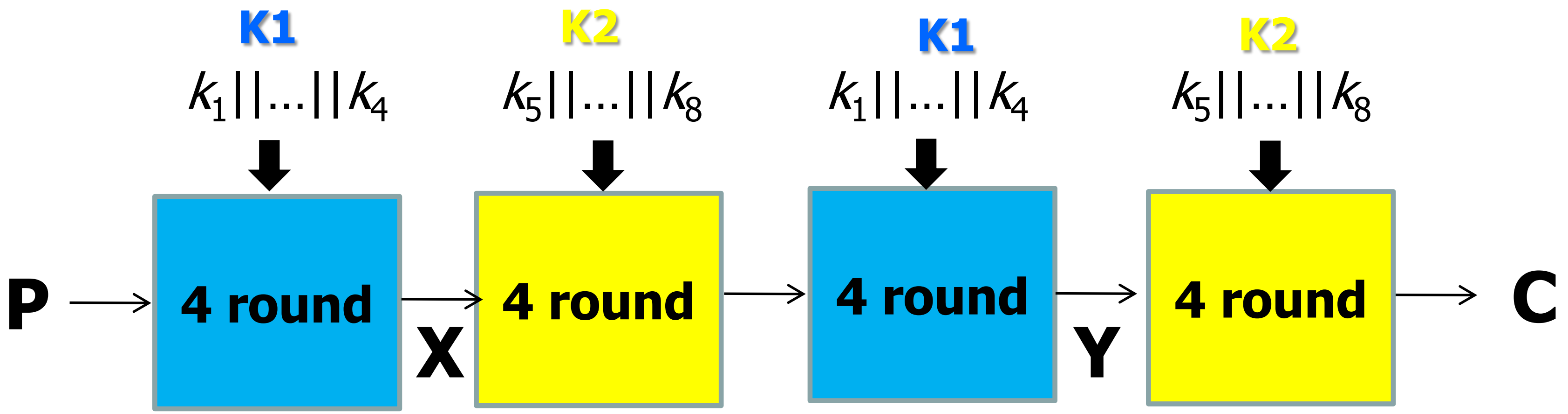
(cover 2^{128} key space)

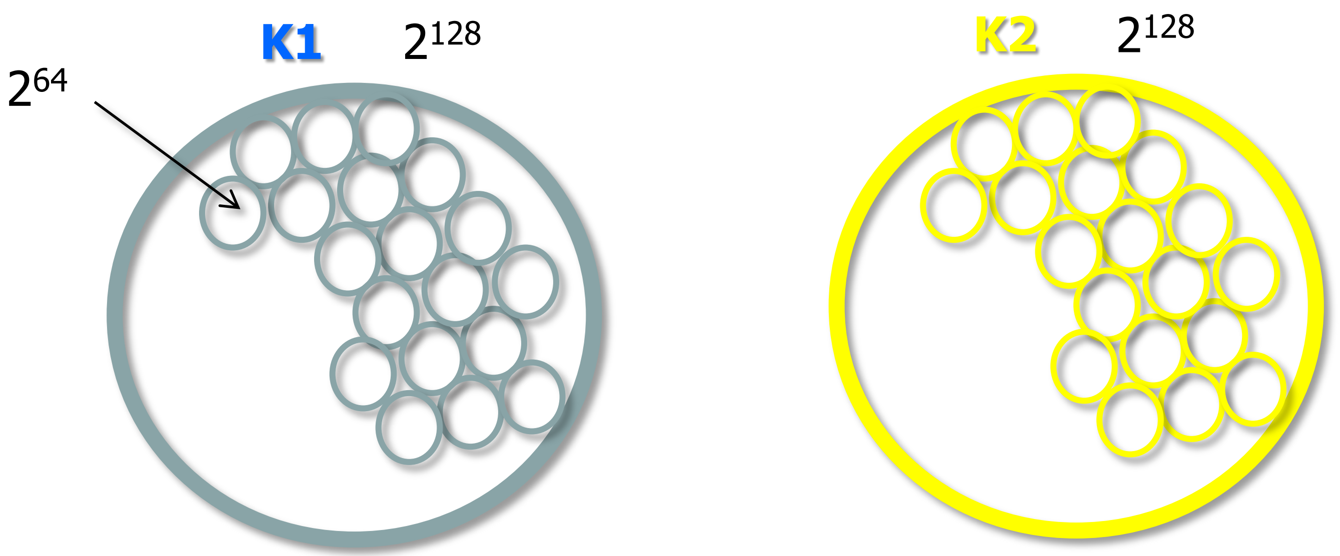
Equivalent Keys



Effective MITM approach

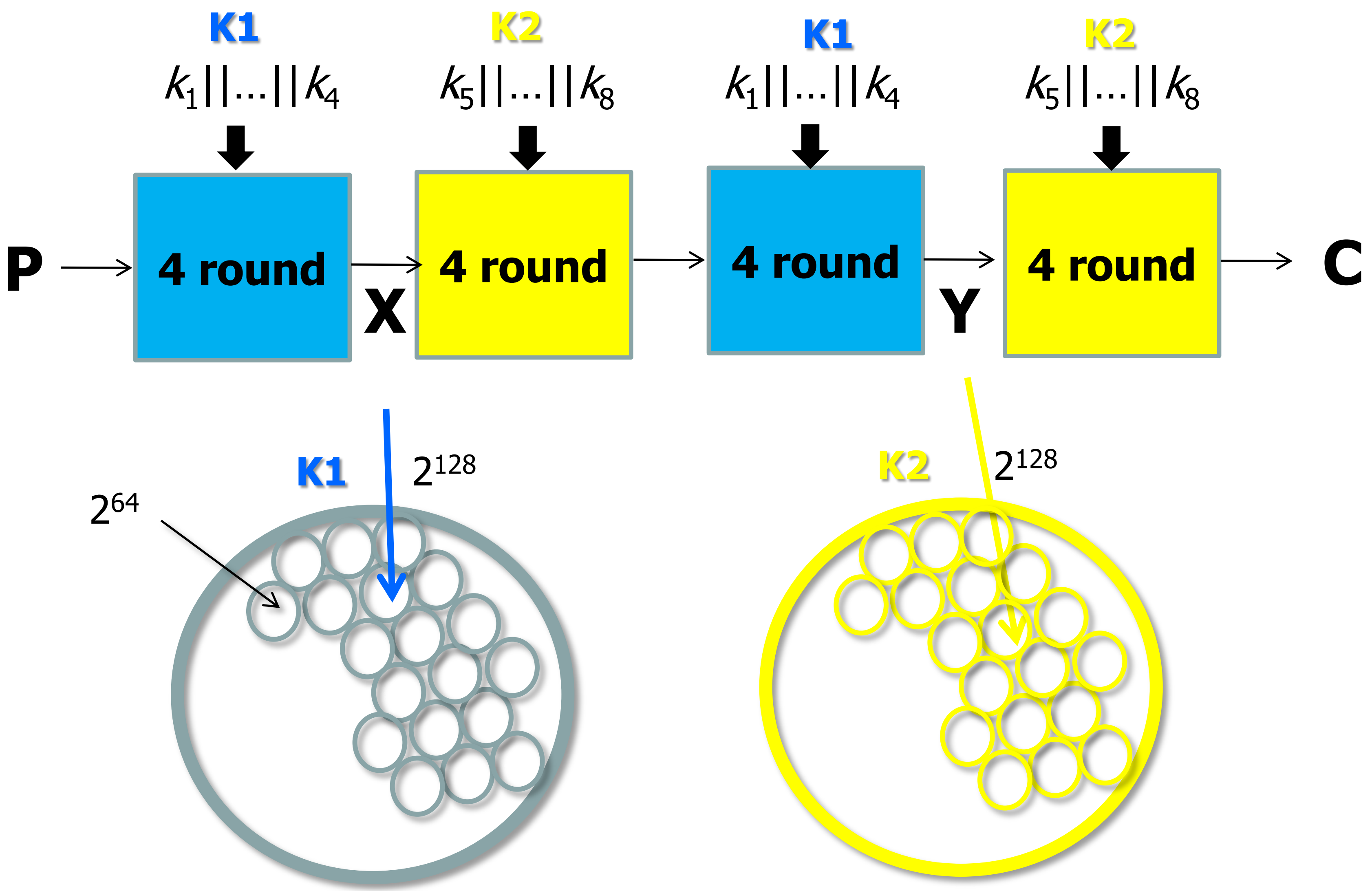
- Guess values of X and Y.





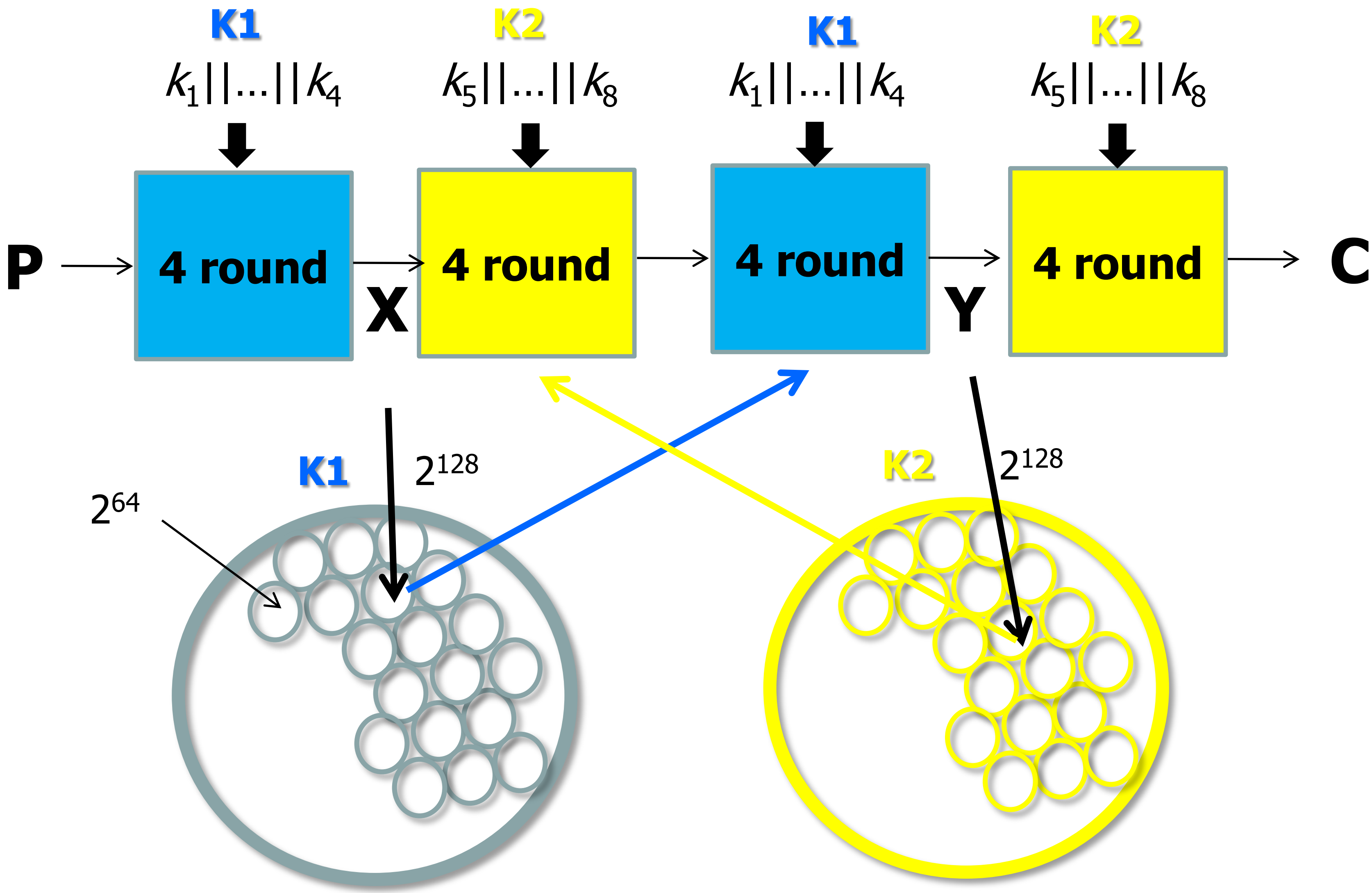
Effective MITM approach

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



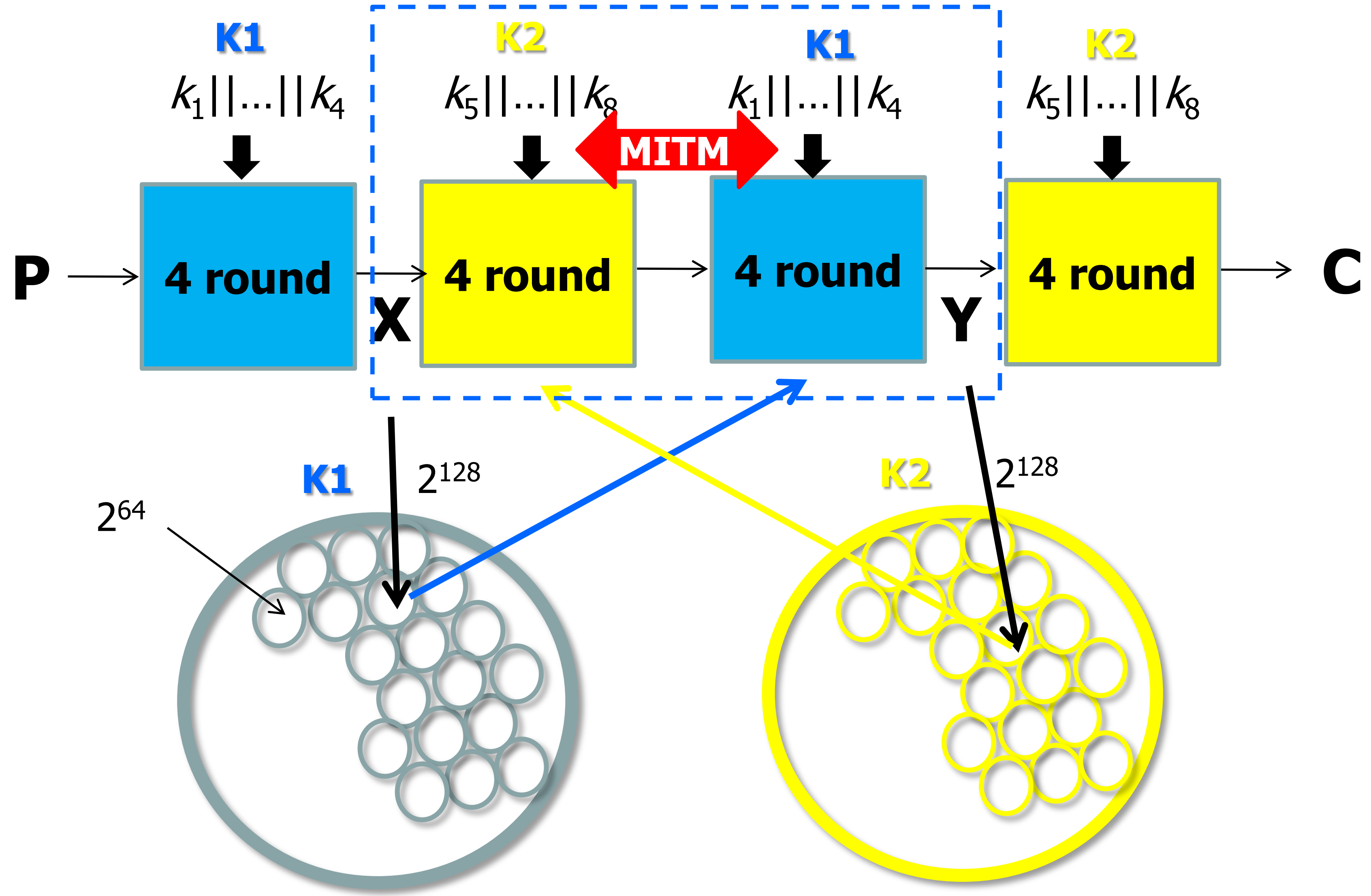
Effective MITM approach

- Mount MITM approach in only intermediate 8 round.



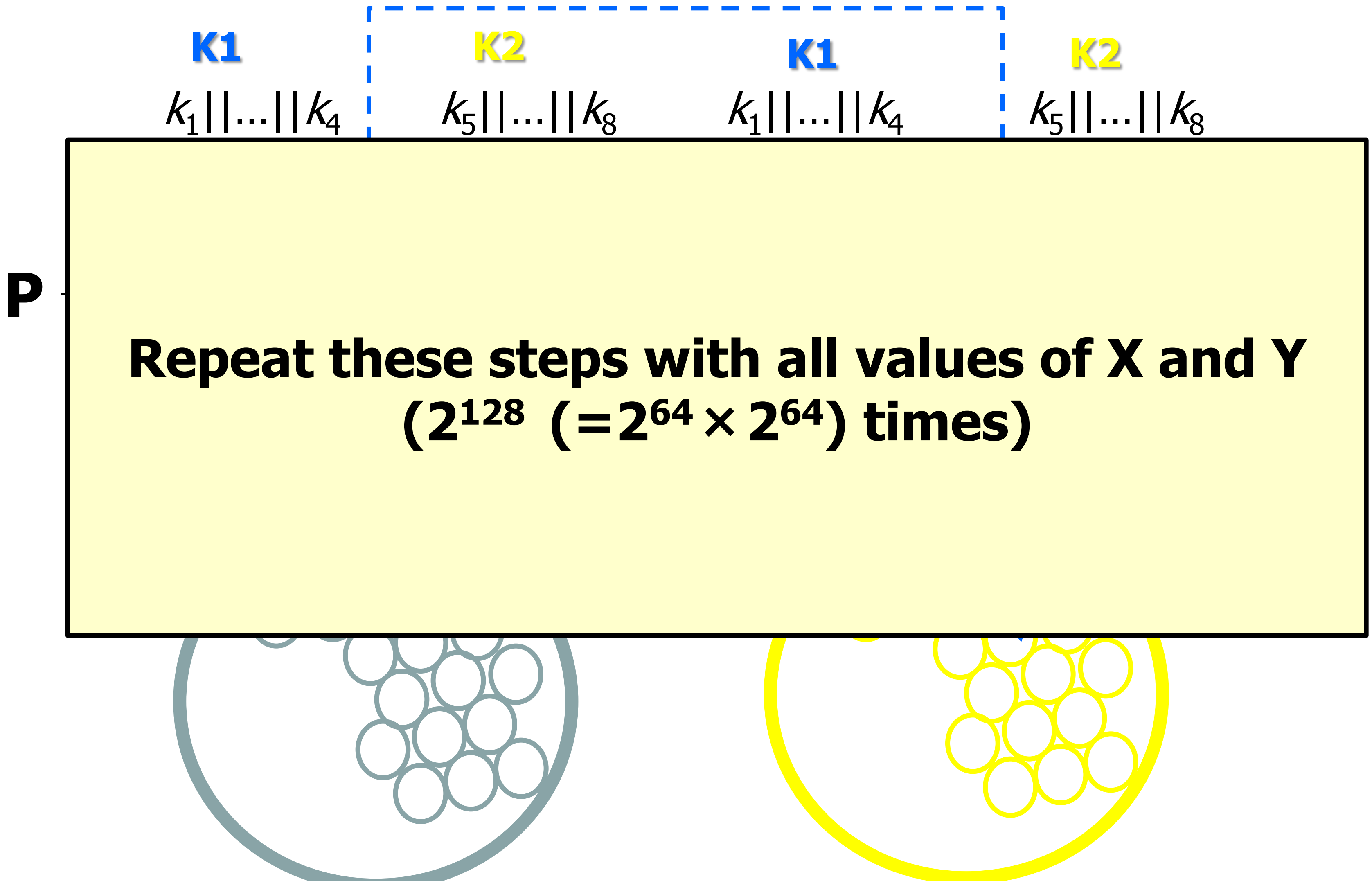
Effective MITM approach

- Mount MITM approach in only intermediate 8 round.

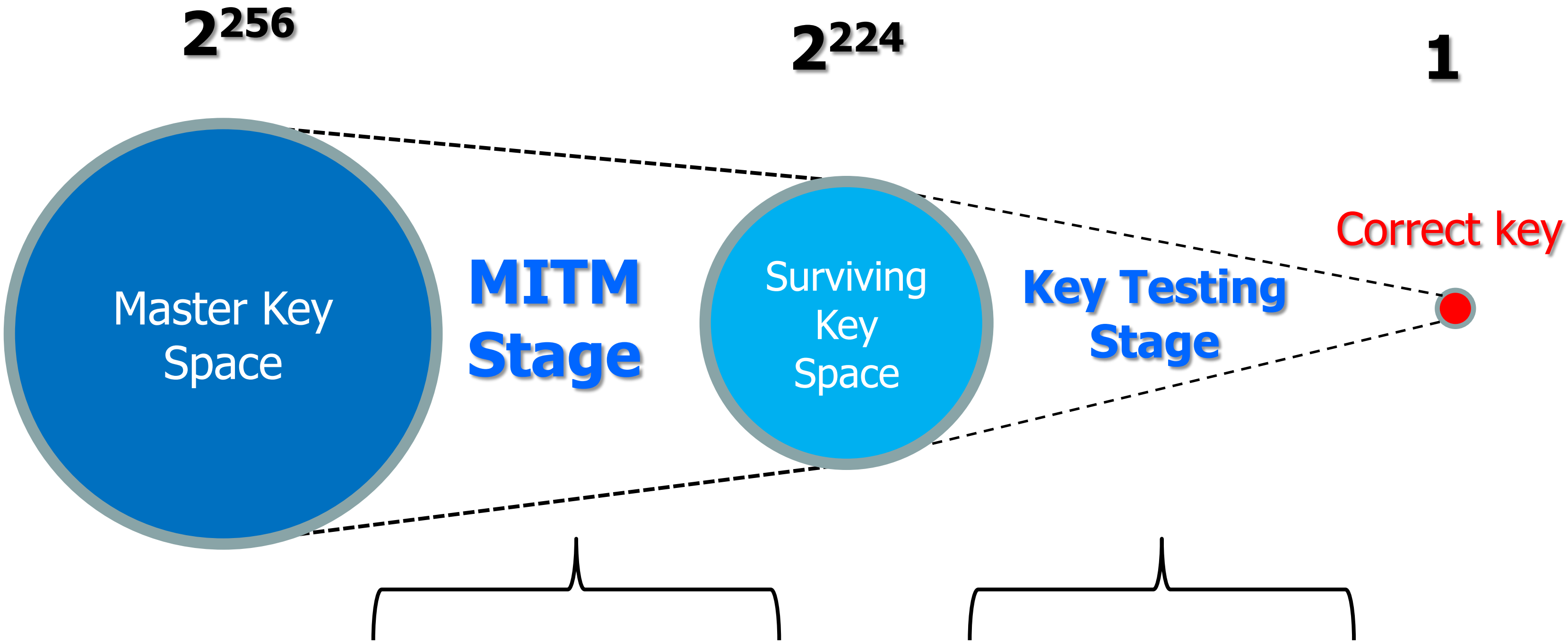


Effective MITM approach

- **Mount MITM approach in only intermediate 8 round.**



Evaluation



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

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

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

Result

- First Single Key Attack on GOST block cipher
  - Applicable to any S-box even including not bijective.[Joc ver.]
  - Several Improvements have been proposed so far.

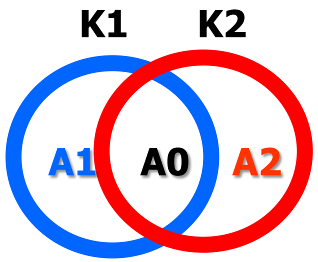
Key Setting	Type of Attack	Round	Complexity	Data	Paper
Single Key	Differential	13	-	2^{51} (CP)	[SK00]
	Slide	24	2^{63}	$2^{64}-2^{18}$ (KP)	[BDK07]
	Slide	30	2^{254}	$2^{64}-2^{18}$ (KP)	[BDK07]
	Reflection	30	2^{224}	2^{32} (KP)	[K08]
	Reflection-MITM	**32 (Full)**	**2^{225}**	**2^{32} (KP)**	**Ours**
	MitM attack	32(Full)	2^{192}	2^{64} (KP)	[DDS12]

3.MitM attack on Block Cipher having Complex KSF

- All Subkeys Recovery Attack on Block cipher
(SAC 2012 w/ K. Shibutani)

MitM Attack on Block Cipher

- Mainly Exploits low key dependency of KSF.
  - Work well for simple key scheduling.
    - Recent Attacks : KTANATAN, GOST, XTEA, IDEA, LED, Piccolo
      => (permutation base KSF)
  - Complex KSF is difficult to analyze or evaluate
    - Only AES attack (complicated and specific)



hard to find independent key bits….

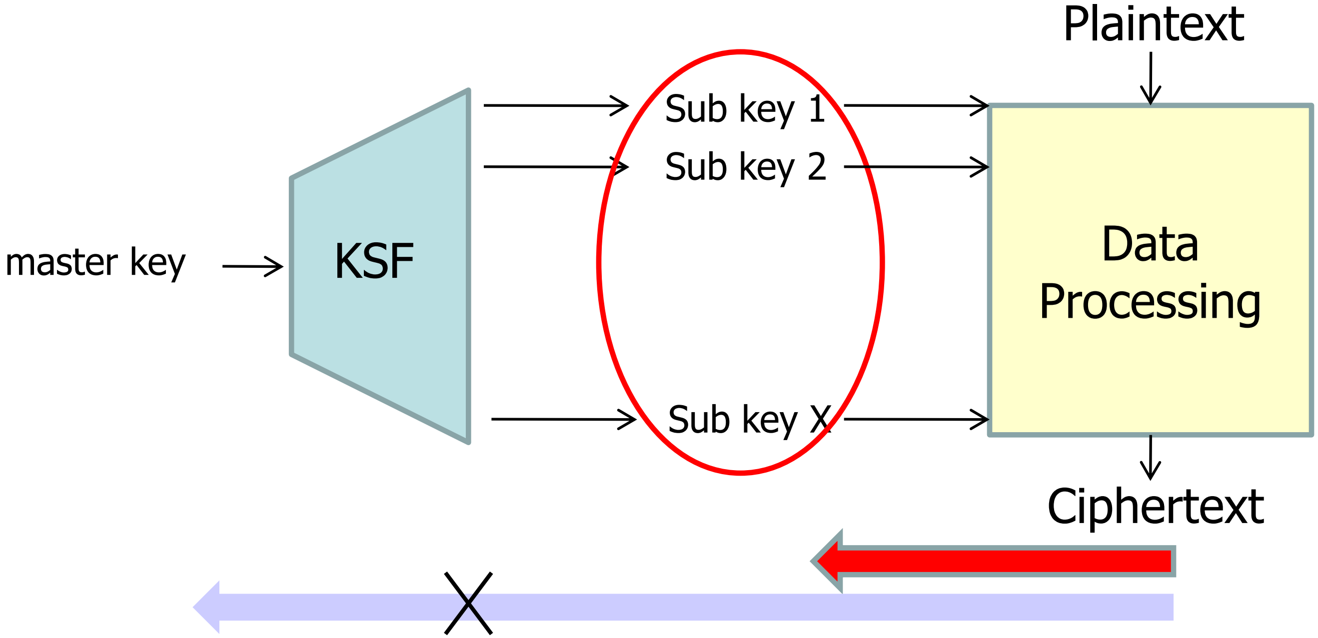
- Our Questions
  - How do we evaluate block cipher having complex KSF against MitM attack?
  - How secure is complex KSF against MitM attack?

Our Approach

Extend MitM attack so that it can be applied to wider class of block cipher

Give a general method for evaluating MitM attack
=> All Subkey Recovery (ASR) Attack

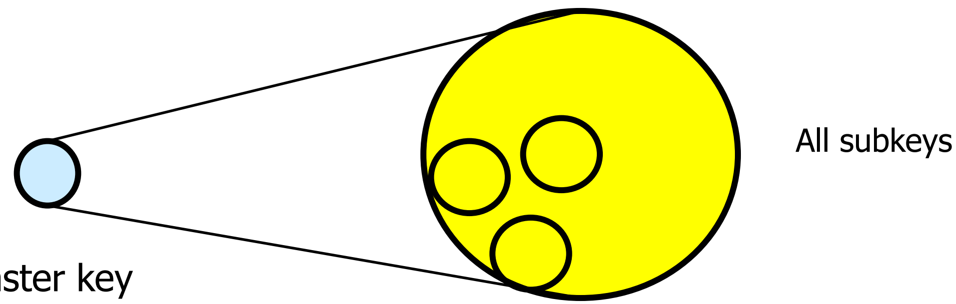
- Our Approach
  - Finding “all subkeys” instead “master key”



Assumption

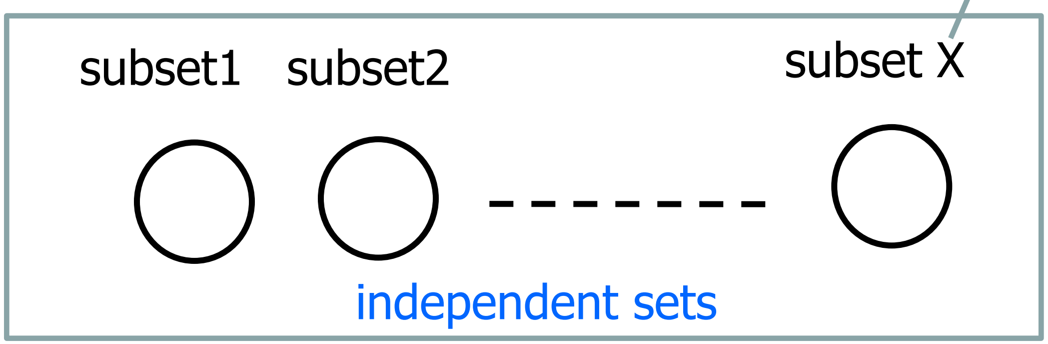
- All subkey are considered as independent variables
  - do not use any relation between subkey bits

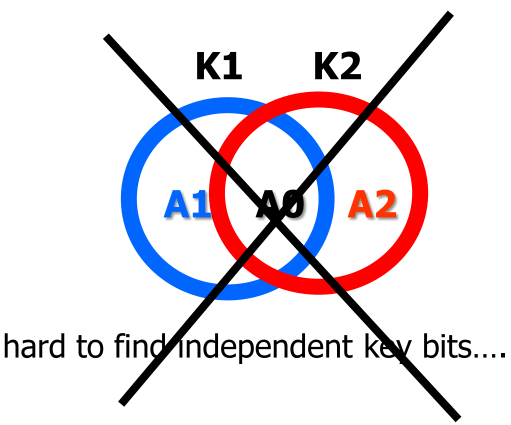
• Search Space => increase : All subkey space (larger than master key)



•Easily mount MitM attack!!

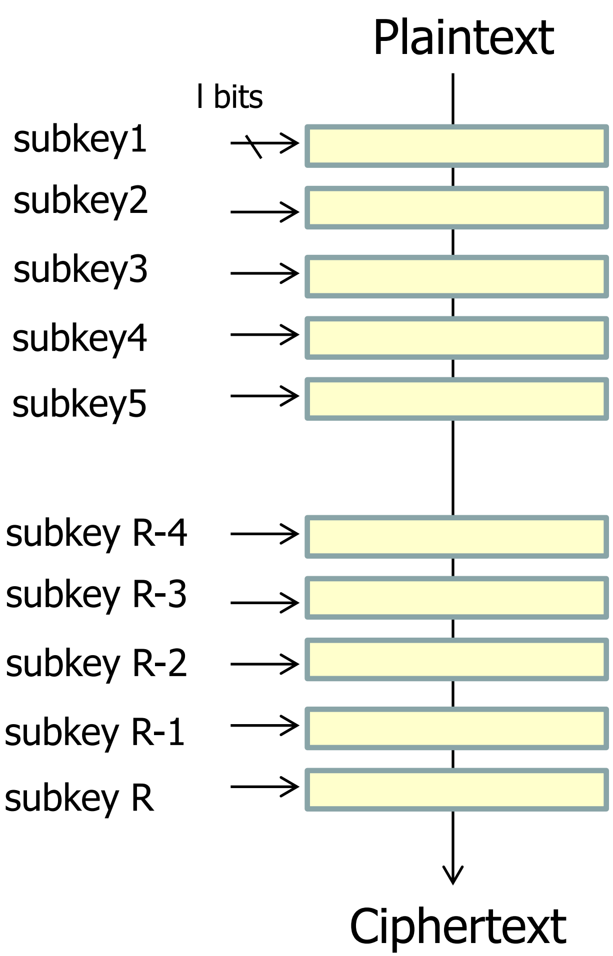
=> All subkey bits are independent bits





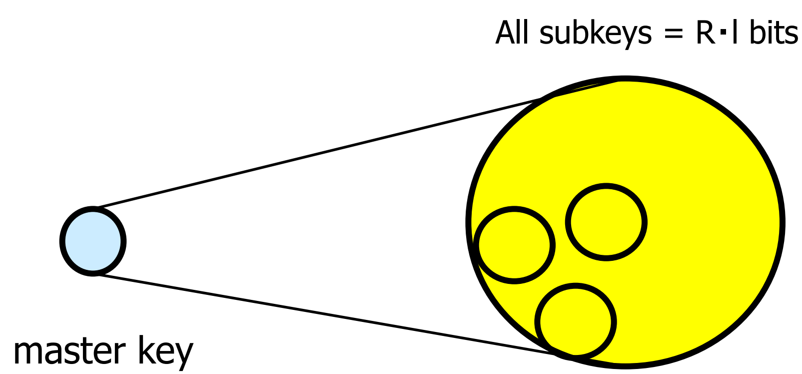
How to recover all subkeys

- Meet in the Middle Approach



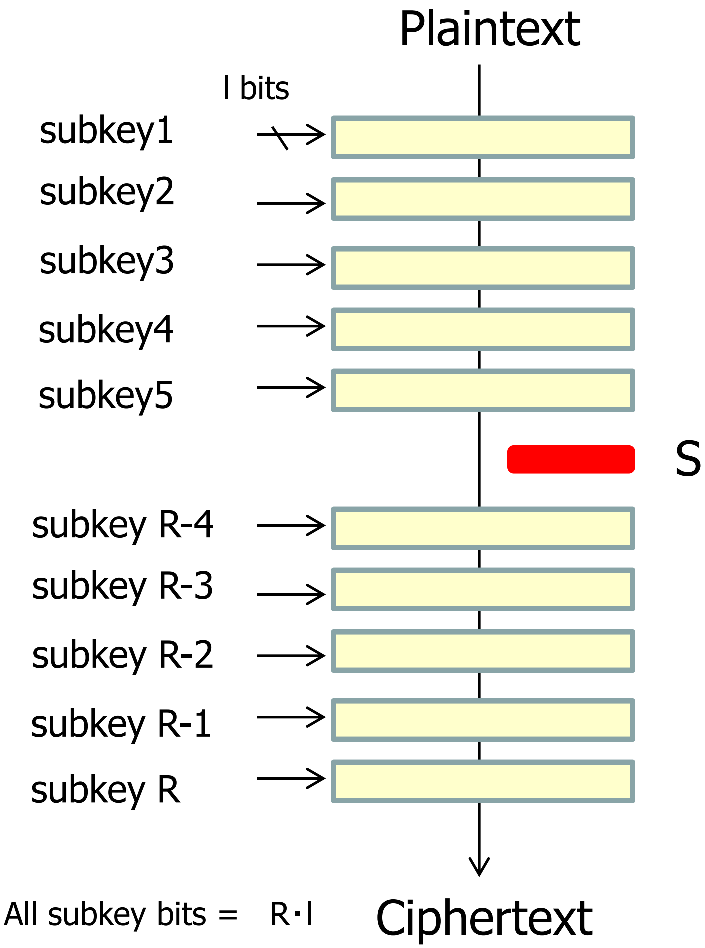
Assumption : All subkeys are independent variables

All subkey : R･l bits (> master key bits)
-R is round number
- l is subkey bits per round



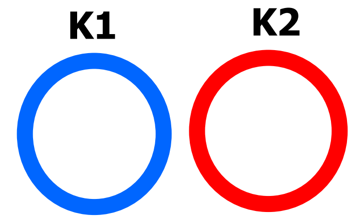
How to recover all subkeys

- Meet in the Middle Approach

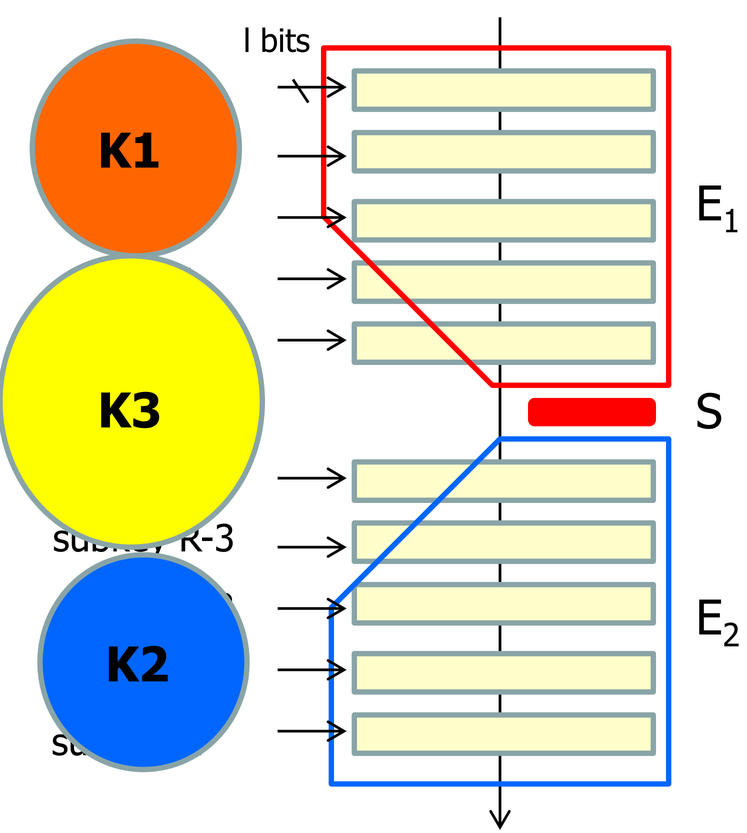


1. Choose *s*-bit matching state *S*

How to recover all subkeys



- Meet in the Middle Approach



Ciphertext

All subkey bits = $R \cdot l$
$= |K_{(1)}| + |K_{(2)}| + |K_{(3)}|$

1. Choose *s*-bit matching state *S*
2. Construct sets of **K1** and **K2** such that
   s = E_1(**K1**, P) and s = E_2(**K2**, C)
3. Compute s = E_1(**K1,** P) with all **K1**
   and Make Table of (s, **K1**) pairs
4. Compute s′ = E_2(**K2**, C) with all **K2**
5. If s = s′, regard it as key candidate

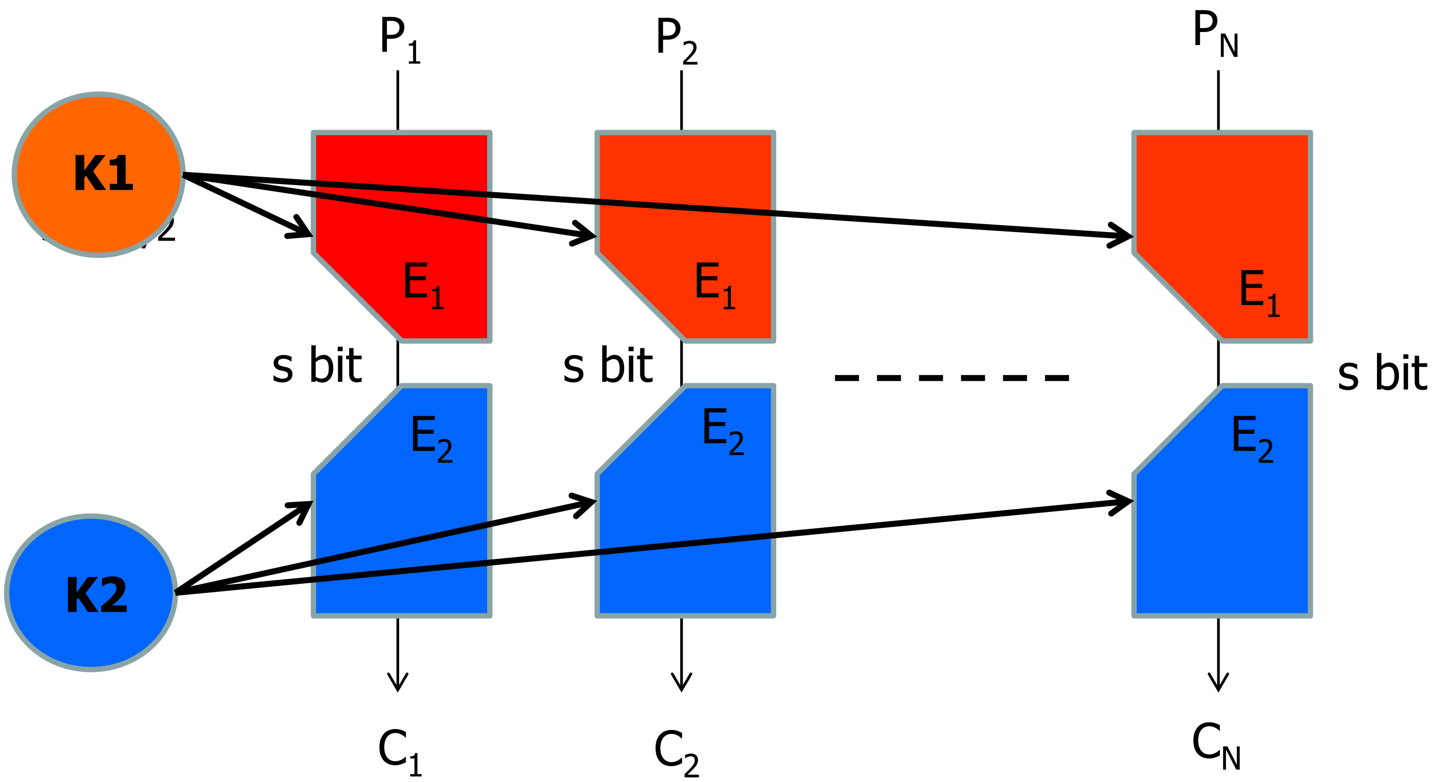
surviving key candidate : $2^{R \cdot l - s}$

↓ additionally use N plaintext
Parallel MitM

surviving key candidate : $2^{R \cdot l - N \cdot s}$

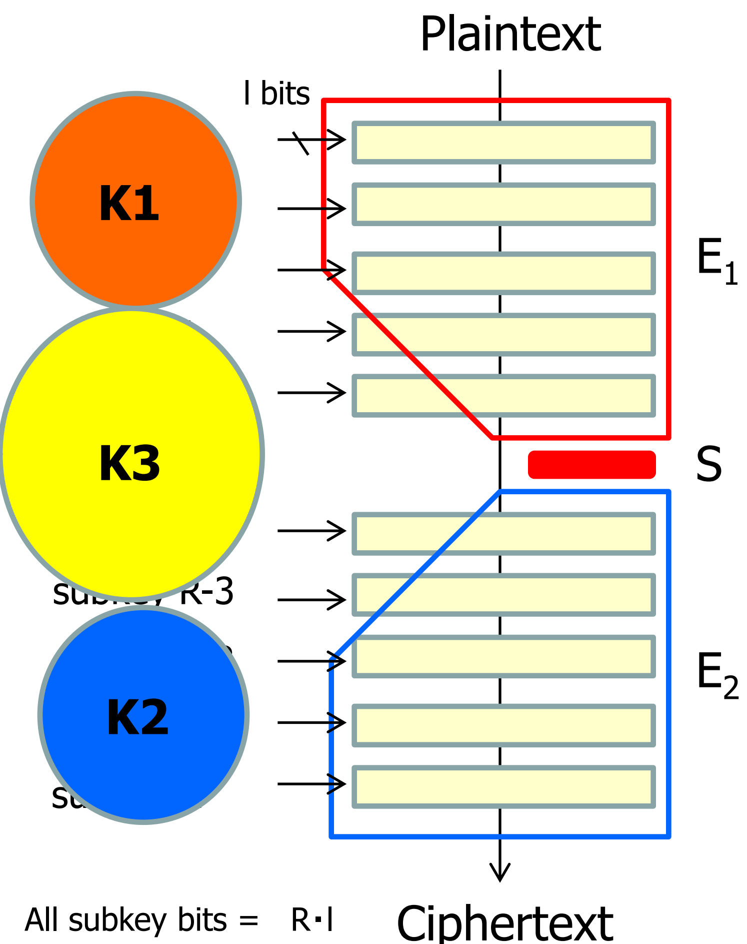
Parallel MitM attack

- Given N Plaintext/Ciphertext



Filter out wrong keys by using N matching state

surviving key candidate : $2^{R \cdot l - N \cdot s}$



All subkey bits = R·l
= $|K_{(1)}| + |K_{(2)}| + |K_{(3)}|$

- **Time complexity**

  $$\underbrace{\max(2^{|K(1)|}, 2^{|K(2)|}) \times N}_{\text{MitM filtering}} + \underbrace{2^{R \cdot l - N \cdot s}}_{\text{brute force of surviving key}}$$

- **Data**

  max (N, (R·l-N·s)/n) KP

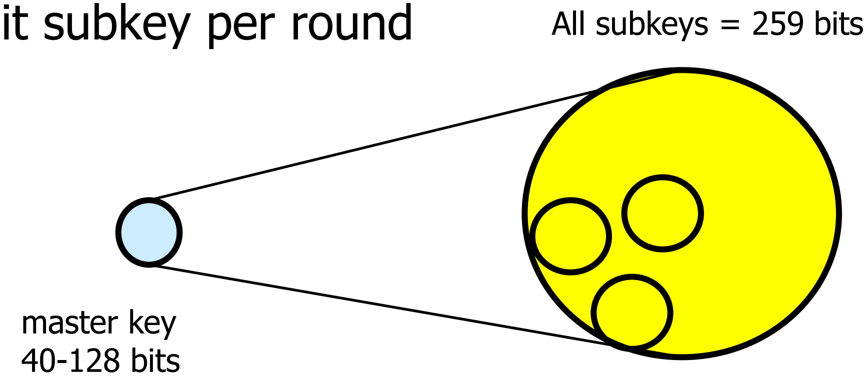
- **Memory**

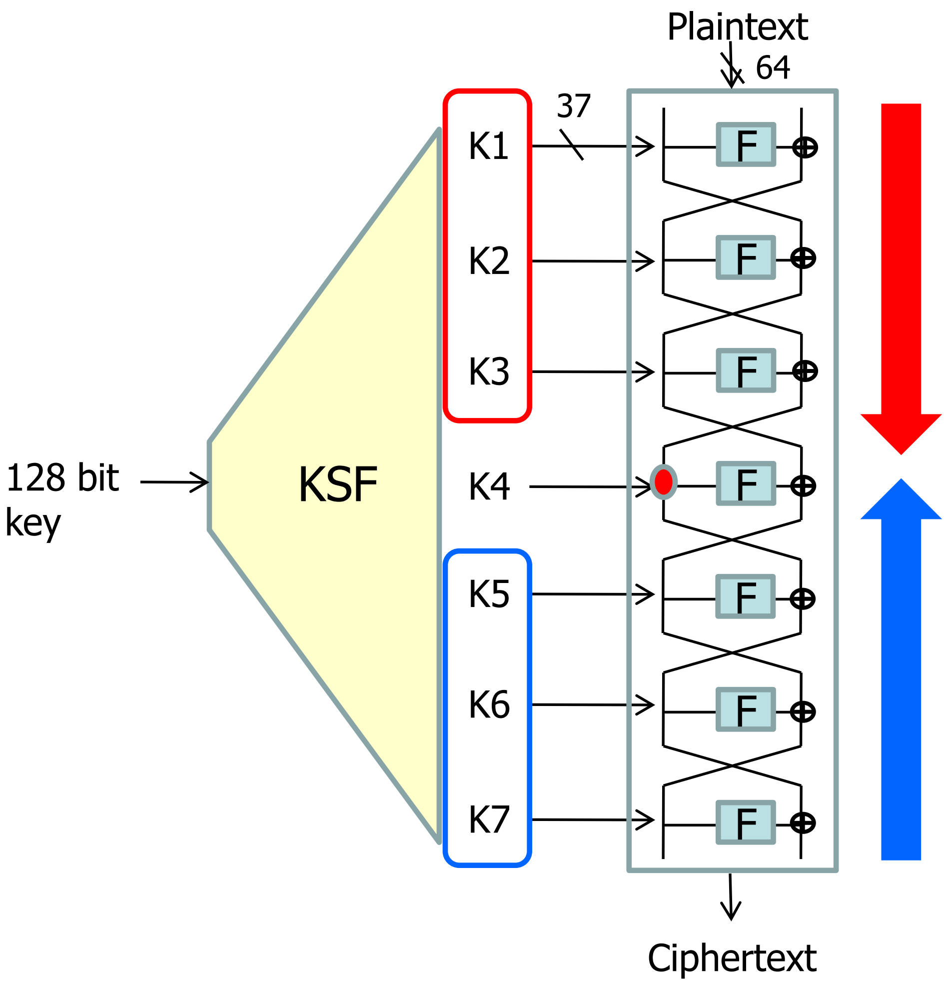
  $\min(2^{|K(1)|}, 2^{|K(2)|}) \times N$

  used for matching

Example : 7-round CAST

- 7-round Balanced Feistel Network
  - 40 - 128-bit key, 64-bit block, 37-bit subkey per round





$s = F_{(1)}(K1, K2, K3)$ $|K_{(1)}| = 111$

$|s| = 32$

$s = F_{(2)}(K5, K6, K7)$ $|K_{(2)}| = 111$

<6 parallel MitM>

Time complexity :

$\max(2^{|K(1)|}, 2^{|K(2)|}) \times N + 2^{R \cdot l - N \cdot s} = 2^{114}$

Data $\max(N, (R \cdot l - N \cdot s)/n)$ KP $= 6$

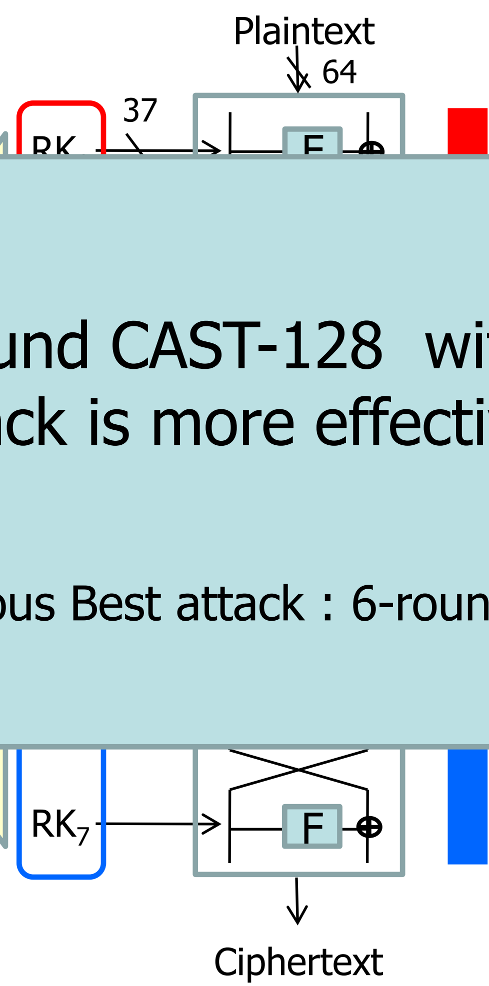
Memory $\min(2^{|K(1)|}, 2^{|K(2)|}) \times N = 2^{114}$

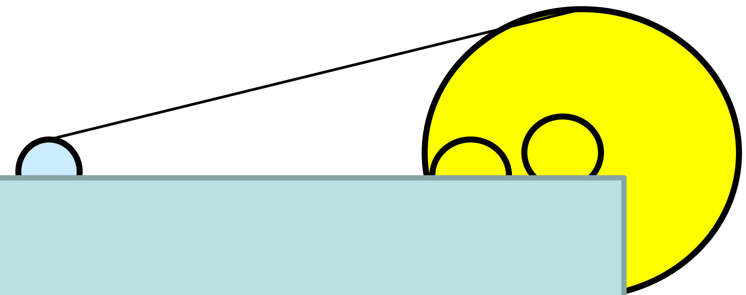
Example : 7-round CAST

- 7-round Balanced Feistel Network
  - 40 - 128-bit key, 64-bit block, 37-bit subkey per round

All subkeys = 259 bits

Best Attack





For 7 round CAST-128 with key size > 110,
ASR attack is more effective than brute force attack

Previous Best attack : 6-round linear attack [M.Wang+09]

128 bit key

1

1

All subkey = 259 bit

Time complexity :

max $(2^{|K(1)|}, 2^{|K(2)|}) \times N + 2^{R \cdot l - N \cdot s} = 2^{114}$

Data max $(N, (R \cdot l - N \cdot s)/n)$ KP = 6

Memory min$(2^{|K(1)|}, 2^{|K(2)|}) \times N = 2^{114}$

Key of ASR Attack

- Complexity
  - max $(2^{|K(1)|}, 2^{|K(2)|}) \times N + 2^{R \cdot l - N \cdot s}$

    => dominated by |K(1)| and |K(2)|

Smaller |K(1)| and |K(2)| leads to more efficient attack

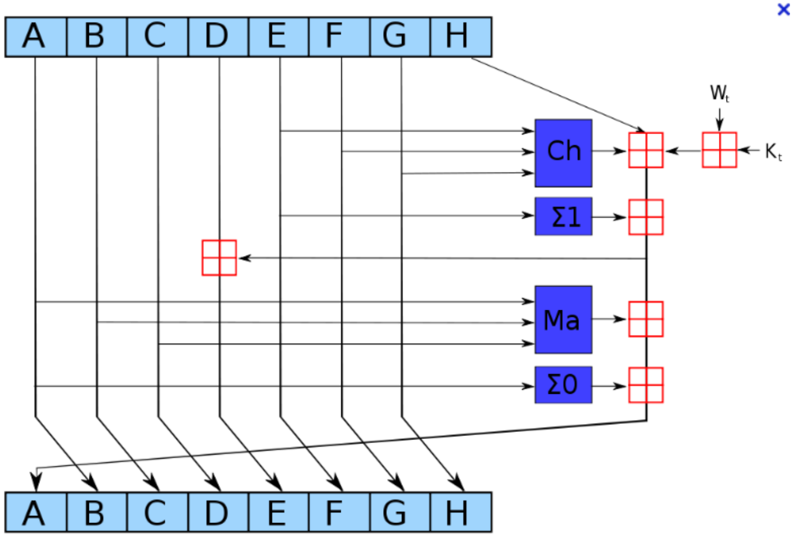
- Point

Finding the matching state S that can be computed by the smallest max(|K(1)|, |K(2)|)

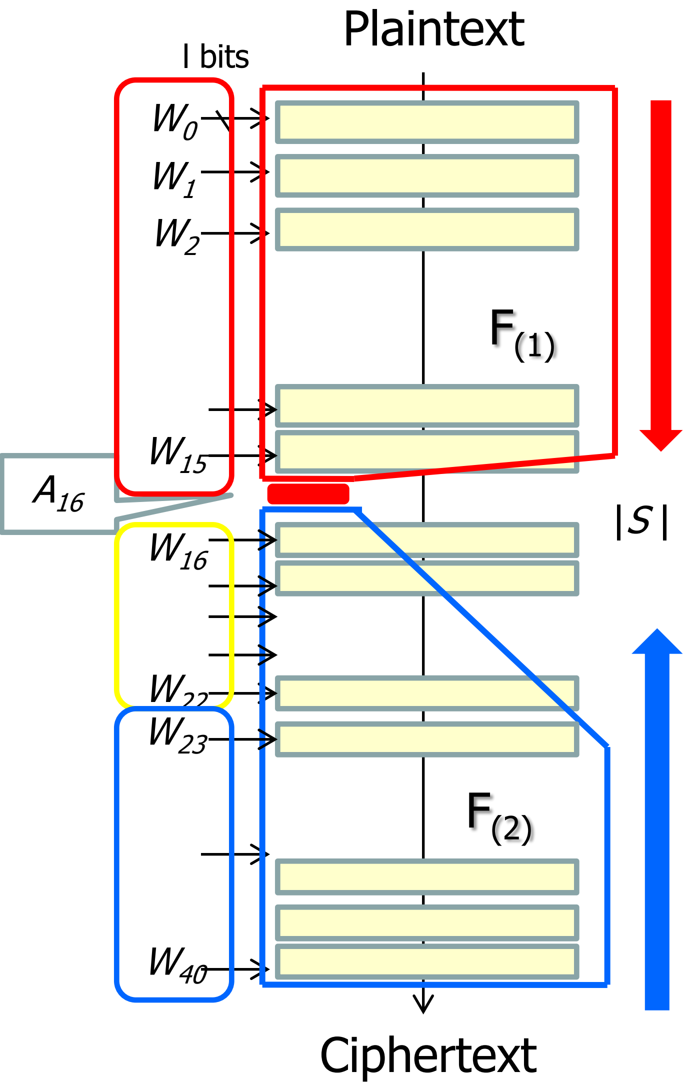
SHACAL-2

- Selected by NESSIE portfolio
- block size : 256 bits, key size : <= 512 bits
- Based on SHA-256 compression function
  - 64-round GFN like construction
- Current Attack : 32 round (Differential-linear)

[Y. Shin+04]



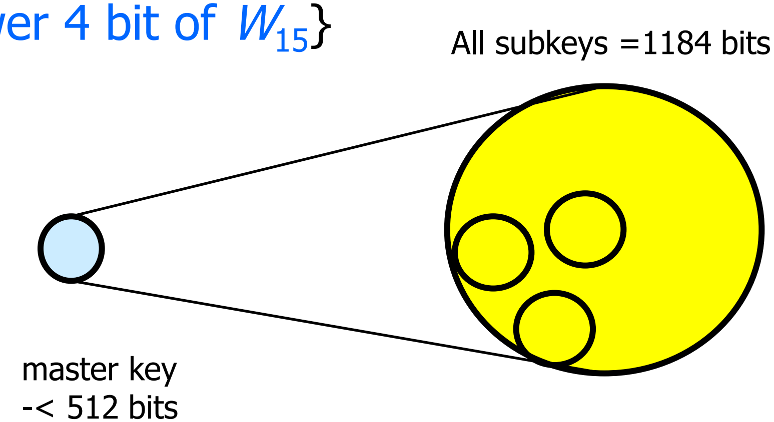
41-round Attack



$K_{(1)} \in \{W_0 - W_{14}$, lower 4 bit of $W_{15}\}$

$|K_{(1)}| = 484$

$|S| = 32$



$K_{(2)} \in \{W_{26} - W_{40}$, lower 4 bit of $W_{23}, W_{24}, W_{25}\}$

$|K_{(2)}| = 492$

Best Attack

N = 244

Time complexity :

$\max(2^{|K(1)|}, 2^{|K(2)|}) \times N + 2^{R \cdot l - N \cdot s} = 2^{500}$

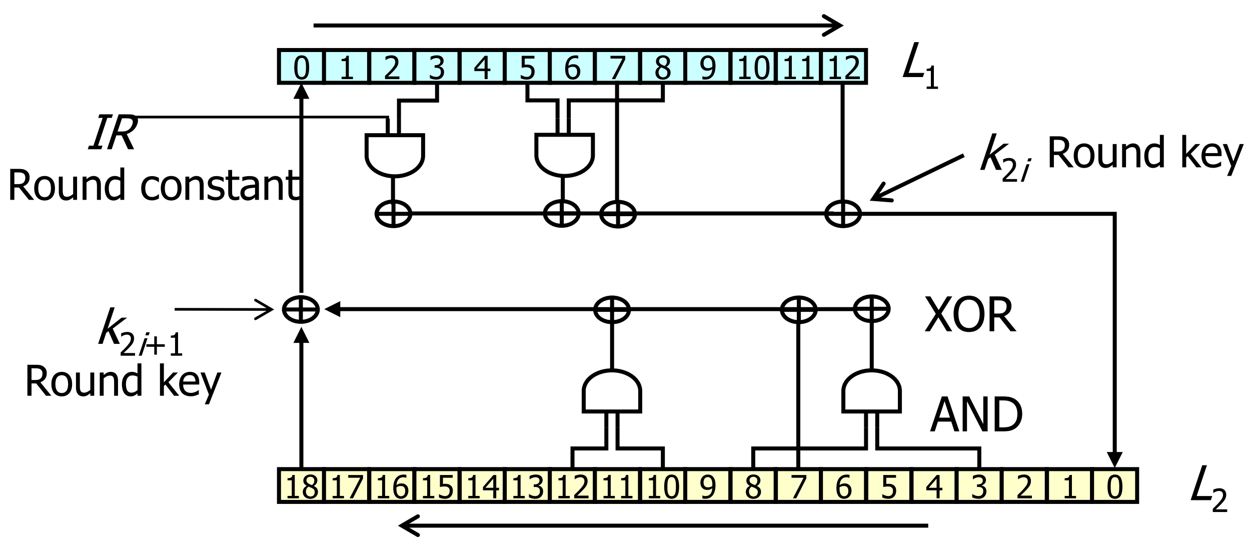
Data $\max(N, (R \cdot l - N \cdot s)/n)$ KP = 224

Memory $\min(2^{|K(1)|}, 2^{|K(2)|}) \times N = 2^{500}$

KATAN Family

- Ultra lightweight block cipher (CHES 2010)
- block size : 32/48/64 bits, key size : 80 bits
- Based on Stream cipher Trivium
  - 254 round LFSR-type construction
- Best Attack 78/70/68 round on KATAN32/48/64

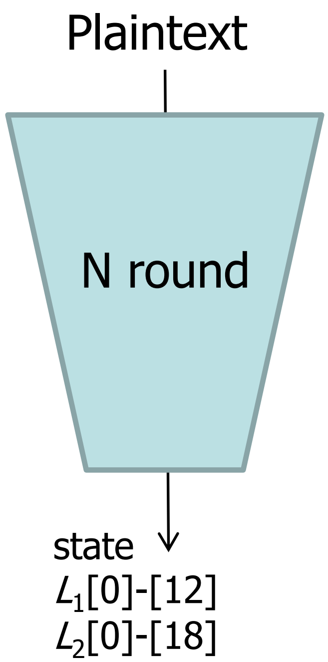
[K+10]



Attack Strategy

- Point of Attack :
  - Finding the state S computed by the smallest max(|K(1)|, |K(2)|)

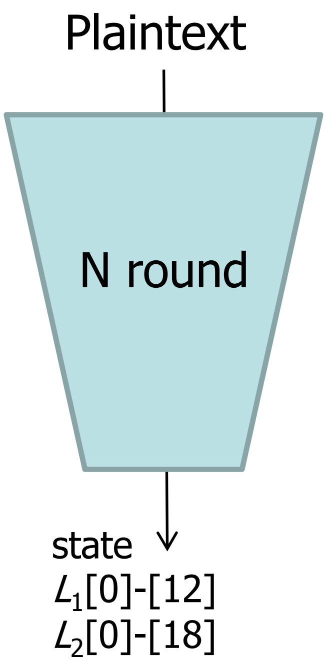
In order to find "good state", we exhaustively observe the number of key bits involved in each state per round



Attack Strategy

- Point of Attack :
  - Finding the state S computed by the smallest max(|K(1)|, |K(2)|)

In order to find “good state”, we exhaustively observe the number of key bits involved in each state per round



After 63 round

L_1

0	1	2	3	4	5	6	7	8	9	10	11	12
108	104	102	100	98	98	96	94	92	88	86	84	84

L_2

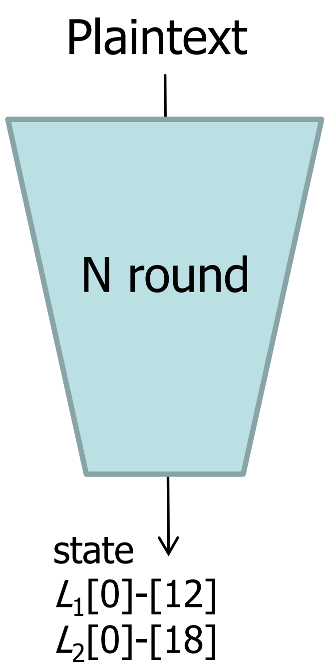
0	1	2	3	4	5	6	7	8	9	10	11	12
104	102	100	98	100	93	92	90	88	90	87	85	83

13	14	15	16	17	18
77	75	75	75	74	68

Attack Strategy

- **Point of Attack :**
  - Finding the state S computed by the smallest max(|K(1)|, |K(2)|)

In order to find "good state", we exhaustively observe the number of key bits involved in each state per round



After 63 round

L_1

0	1	2	3	4	5	6	7	8	9	10	11	12
108	104	102	100	98	98	96	94	92	88	86	84	84

L_2

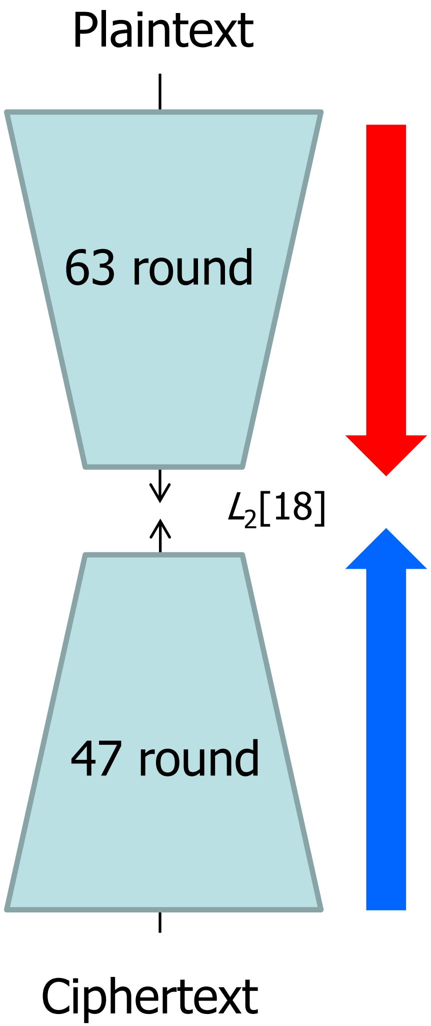
0	1	2	3	4	5	6	7	8	9	10	11	12
104	102	100	98	100	93	92	90	88	90	87	85	83

13	14	15	16	17	18
77	75	75	75	74	68

L_2[18] can be computed by 68 bit of subkey and Plaintext

110 round Attack on KATAN32

All subkey = 220 bit



$K_{(1)} \in \{k_0\text{-}k_{54}, k_{56}, k_{57}, k_{58}, K_{60}\text{-}k_{64}, k_{68}, k_{71}, k_{73}, k_{77}, k_{88}\}$

$|K_{(1)}| = 68$

$|S| = 1$

$K_{(2)} \in \{k_{126}, k_{138}, k_{142}, k_{146}, k_{148}, k_{150}, k_{153}, k_{154}, k_{156}, k_{158}, k_{160}\text{-}k_{219}\}$

$|K_{(2)}| = 70$

$N = 138$

Time complexity :

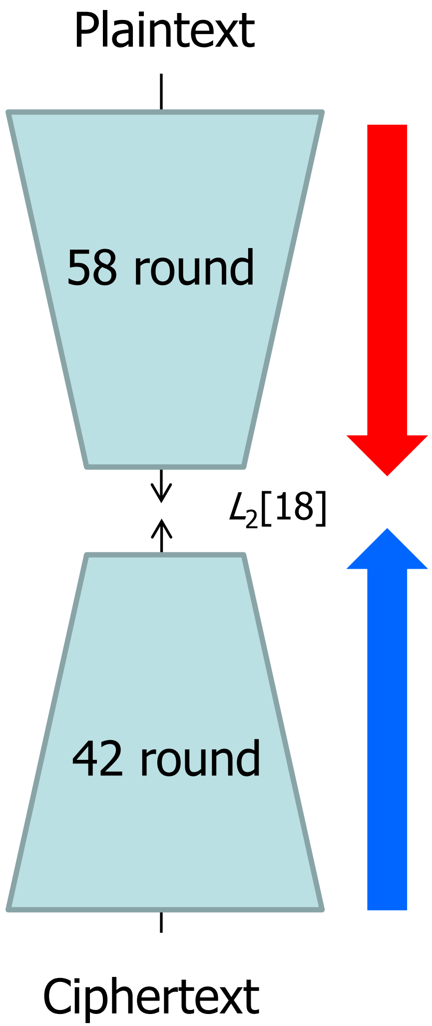
$\max(2^{|K(1)|}, 2^{|K(2)|}) \times N + 2^{R \cdot l - N \cdot s} = 2^{77.1}$

Data max (N, $(R \cdot l - N \cdot s)/n$) KP $= 138$

Memory $\min(2^{|K(1)|}, 2^{|K(2)|}) \times N = 2^{75.1}$

100 round Attack on KATAN48

All subkey = 200 bit



$K_{(1)} \in \{k_0\text{-}k_{60}, k_{62}, k_{64}, k_{65}, k_{66}, k_{69}, k_{71}, k_{72}, k_{75}, k_{79}, k_{86}\}$

$|K_{(1)}| = 71$

$|S| = 1$

$K_{(2)} \in \{k_{116}, k_{122}, k_{124}, k_{128}, k_{130}, k_{132}, k_{134}, k_{135}, k_{136}, k_{138}\text{-}k_{199}\}$

$|K_{(2)}| = 71$

Best Attack

$N = 128$

Time complexity :

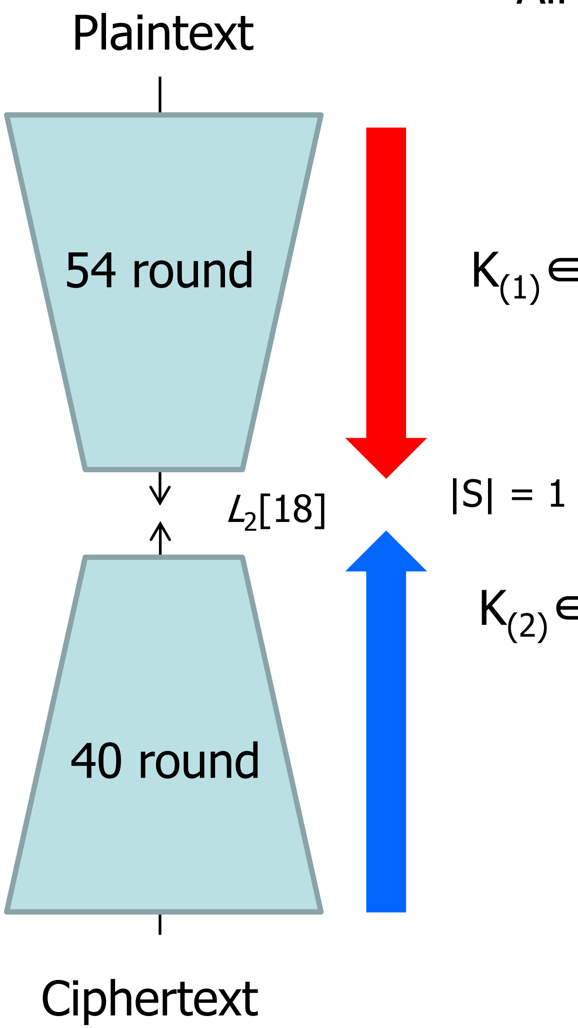
$\max(2^{|K(1)|}, 2^{|K(2)|}) \times N + 2^{R \cdot l - N \cdot s} = 2^{78}$

Data $\max(N, (R \cdot l - N \cdot s)/n)$ KP $= 128$

Memory $\min(2^{|K(1)|}, 2^{|K(2)|}) \times N = 2^{78}$

94 round Attack on KATAN64

All subkey = 188 bit



$K_{(1)} \in \{k_0\text{-}k_{61}, k_{63}, k_{64}, k_{65}, k_{66}, k_{68}, k_{69}, k_{71}, k_{75}, k_{82}\}$

$|K_{(1)}| = 71$

$K_{(2)} \in \{k_{108}, k_{110}, k_{114}, k_{116}, k_{118}, k_{120}, k_{122}, k_{124}\text{-}k_{187}\}$

$|K_{(2)}| = 71$

Best Attack

$N = 116$

Time complexity :

$\max(2^{|K(1)|}, 2^{|K(2)|}) \times N + 2^{R \cdot l - N \cdot s} = 2^{77.68}$

Data $\max(N, (R \cdot l - N \cdot s)/n)$ KP $= 116$

Memory $\min(2^{|K(1)|}, 2^{|K(2)|}) \times N = 2^{77.68}$

Our Results

- We can update best attack w.r.t. #attacked round

Algorithm	#attacked round	Time	Memory	Data	reference
CAST-128	6	$2^{88.5}$	-	$2^{53.96}$ KP	[MXC09]
	7	2^{114}	2^{114}	6KP	Our
SHACAL-2	32	$2^{504.2}$	$2^{48.4}$	$2^{43.4}$CP	[S+04]
	41	2^{500}	2^{492}	244 KP	Our
KATAN32	78	2^{76}	-	2^{16} CP	[KMN10]
	110	2^{77}	$2^{75.1}$	138 KP	Our
	115	-	-		[AL12]
KATAN48	70	2^{78}	-	2^{31} CP	[KMN10]
	100	2^{78}	278	128 KP	Our
KATAN64	68	2^{78}	-	2^{32} CP	[KMN10]
	94	$2^{77.68}$	$2^{77.68}$	116 KP	Our
FOX128	5	$2^{205.6}$	-	2^{9} CP	[WZF05]
	5	2^{228}	2^{228}	14 KP	Our
Blowfish*	16	2^{292}	2^{260}	9 KP	Our
Blowfish-8R*	8	2^{160}	2^{131}	5 KP	Our

＊ : Known F function setting

CAST, SHACAL, Blowfish support variable key length, our attacks are applicable to restricted parameter

Advantage and Limitation

- Advantage
  - Our attack works any KSF even if Ideal function.
    - Generic and simple attack
  - Thanks to MitM attack w/o Spice and Cut,
    Data complexity is very low.

- Limitation
  - When Key size is smaller, ASR attack is less effective.
    => bound of attack complexity = key size (not all subkeys)
  - Huge memory requirement

Conclusion

- Introduced several results w.r.t MitM attack of Block Cipher
  - MitM on Block cipher having *simple* KSF
    - XTEA, LED, Piccolo (@ ACISP 2012 w/ K. Shibutani)
    - GOST (@ FSE 2011 and JoC)
  - MitM on Block cipher having *complex* KSF
    - All subkeys recovery attack (@ SAC 2012 w/ K. Shibutani)
    - KATAN-32/48/64, SHACAL-2, CAST-128

Thank you for your attention

Reference

[AS08] : K. Aoki and Y. Sasaki, "Preimage Attacks on One-Block MD4, 63-Step MD5 and More", SAC 2008

[SA09] :Y. Sasaki and K. Aoki, "Finding Preimages in Full MD5 Faster Than Exhaustive Search", EUROCRYPT2009

[GLRW10] : J. Guo, S. Ling, C. Rechberger and H. Wang, "Advanced Meet-in-the-Middle Preimage Attacks: First Results on Full Tiger, and Improved Results on MD4 and SHA-2", ASIACRYPT2010

[KK12] : S. Knellwolf and D. Khovratovich, "New Preimage Attacks against Reduced SHA-1", CRYPTO2012

[KRS12] : D. Khovratovich and C. Rechberger and A. Savelieva, "Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 family" FSE2012

[LIS12] : L, Ji, T. Isobe and K.Shibutani,"Converting Meet-in-the-Middle Preimage Attack into Pseudo Preimage : Application to SHA-2", FSE2012,

[D'12] : : D. Khovratovich, "Bicliques for permutations: collision and preimage attacks in stronger settings", ePrint2012/141

[BR09] : A. Bogdanov and C. Rechberger, "A 3-Subset Meet-in-the-Middle Attack: Cryptanalysis of the Lightweight Block Cipher KTANTAN", SAC 2010

[I'11] : T. Isobe, "A Single Key Attack on the Full GOST Block Cipher", FSE2011

[BKR11] : A. Bogdanov and D. Khovratovich and C. Rechberger, "Biclique Cryptanalysis of the Full AES", ASIACRYPT2011

[KLR11] : D. Khovratovich, G.Leurent and C. Rechberger, "Narrow-Bicliques: Cryptanalysis of Full IDEA", EUROCRYPT2011

Reference

[IS12] : T. Isobe and K. Shibutani, "Security Analysis of the Lightweight Block Ciphers XTEA, LED and Piccolo" ACISP2012

[NW97] : R.M. Needham and D.J. Wheeler, "Tea Extentions"

[GPPR11] : J. Guo,T. Peyrin, A. Poschmann and M. J. B. Robshaw, "The LED Block Cipher" CHES2011

[SIHMAS11] : K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita and T. Shirai, "Piccolo: An Ultra-Lightweight Blockcipher"CHES2011

[BW12] : A. Bogdanov and M. Wang "Zero Correlation Linear Cryptanalysis with Reduced Data Complexity " FSE2012

[SWS+] : Y. Sasaki, L. Wang, Y. Sakai, K. Sakiyama and K. Ohta, "Three-Subset Meet-in-the-Middle Attack on Reduced XTEA" Africacrypt2012

[SK00] : H. Seki and T. Kaneko, "Differential Cryptanalysis of Reduced Rounds of GOST", SAC2000

[BDK07] : E. Biham and O. Dunkelman and N. Keller, "Improved Slide Attacks", FSE2007

[KM07] : O. Kara and C. Manap, "A New Class of Weak Keys for Blowfish", FSE2007

[K08] O.Kara, "Reflection Cryptanalysis of Some Ciphers", INDOCRYPT2008

[DDS] : I. Dinur, O. Dunkelman and A. Shamir, "Improved Attacks on Full GOST", FSE2012

Reference

[MXC09] : M.Wang and X. Wang and C. Hu, "New Linear Cryptanalytic Results of Reduced-Round of CAST-128 and CAST-256", SAC2009

[S+04] : Y.Shin, J. Kim, G. Kim, S. Hong and S. Lee, "Differential-Linear Type Attacks on Reduced Rounds of SHACAL-2", ACISP2004

[SMN10] : S. Knellwolf, W. Meier and M. Naya-Plasencia, "Conditional Differential Cryptanalysis of NLFSR-based Cryptosystems" ASIACRYPT2010.

[WZF05] : W. Wu and W. Zhang and D. Feng, "Integral Cryptanalysis of Reduced FOX Block Cipher", ICISC 2005