Improved Linear Sieving Techniques with Applications to Step-Reduced LED-64

Itai Dinur[1], Orr Dunkelman[2,4],
Nathan Keller[3] and Adi Shamir[4]

[1]École normale supérieure, France

[2]University of Haifa, Israel

[3]Bar-Ilan University, Israel

[4]The Weizmann Institute, Israel

Summary

- We propose several new techniques in **MITM** attacks on block ciphers
- We apply the new techniques to the lightweight block cipher LED-64 (presented by Guo et al. at CHES'11)
- We improve the **best known attacks** on some step-reduced variants of this cipher in several models

Summary

Reference	Model	Steps	Time	Data	Memory
IS’12	Single-Key	2	2^{56}	2^{8} CP	2^{8}
New	**Single-Key**	**2**	$\mathbf{2^{48}}$	$\mathbf{2^{16}}$ **CP**	$\mathbf{2^{16}}$
DDKS’13	Single-Key	2	2^{60}	2^{49} KP	2^{60}
New	**Single-Key**	**2**	$\mathbf{2^{48}}$	$\mathbf{2^{48}}$ **KP**	$\mathbf{2^{48}}$
MRTV’12	Related-Key	3	2^{60}	2^{60} CP	2^{60}
New	**Related-Key**	**3**	$\mathbf{2^{49}}$	$\mathbf{2^{49}}$ **CP**	$\mathbf{2^{49}}$

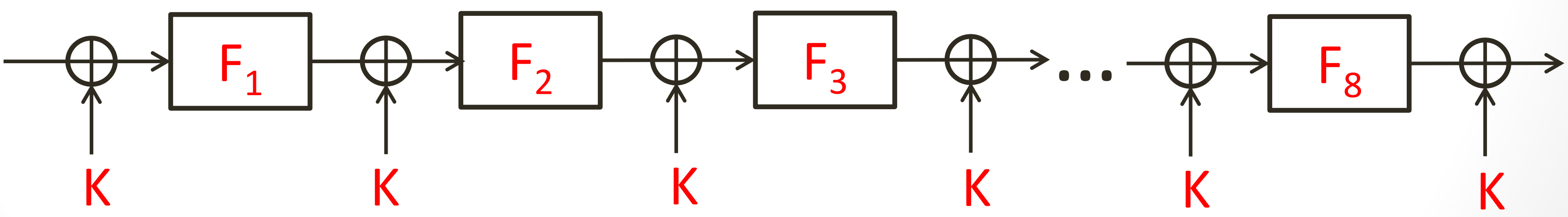
- Also note the theoretical attacks:
  - [DDKS’13] 3-step known plaintext attack
  - [MRTV’12] 4-step related-key attack

Summary

- Our main tool is called a **linear key sieve**
  - Exploits linear dependencies between key bits guessed in both sides of the attack
- We show for the first time that the **splice-and-cut** attack can be applied in the **known plaintext model**
- Our related-key attack in based on an extension of **differential MITM** on **AES-based designs**

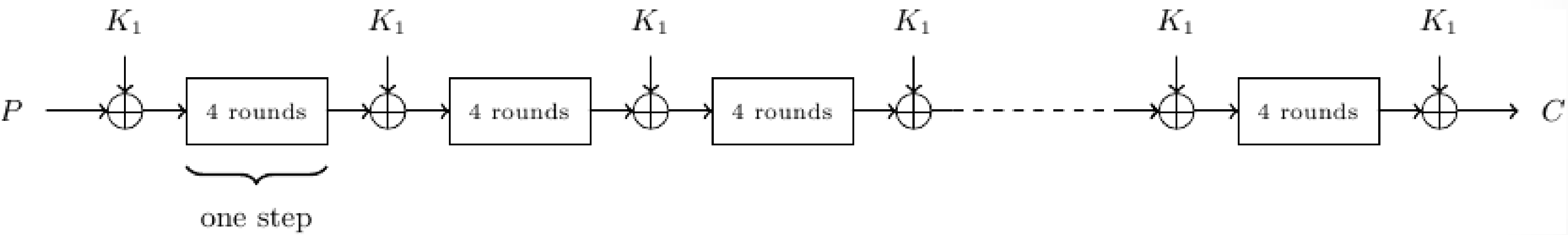
LED

- 64-bit lightweight block cipher presented by Guo, Peyrin, Poschmann, and Robshaw at CHES'11
- Two main versions: LED-64 and LED-128
- LED-64 is an 8-step **EM** scheme with 1 key

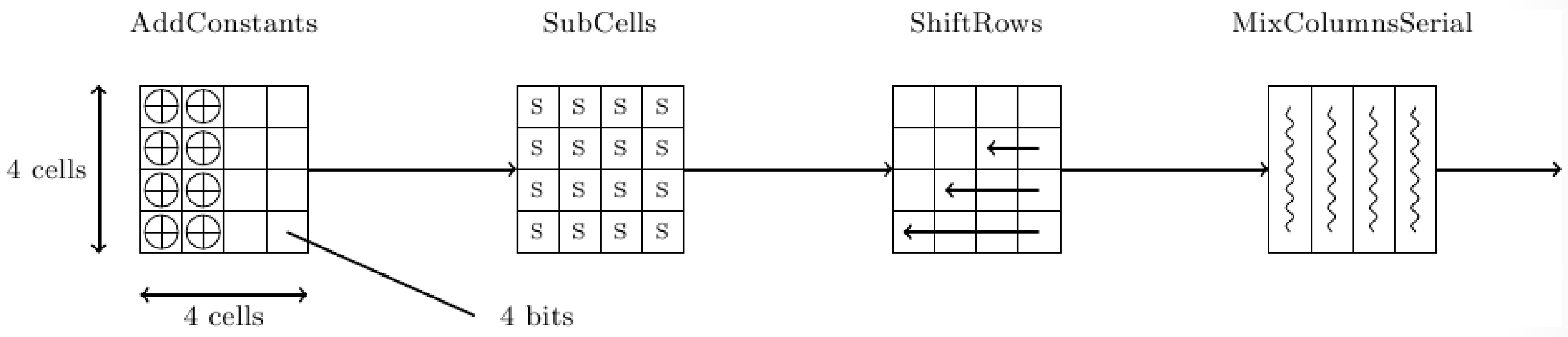


The LED Step Function

- LED uses an AES-like **design**
- Each **step** (F_1, F_2,…,F_8) applies 4 AES-like **rounds**

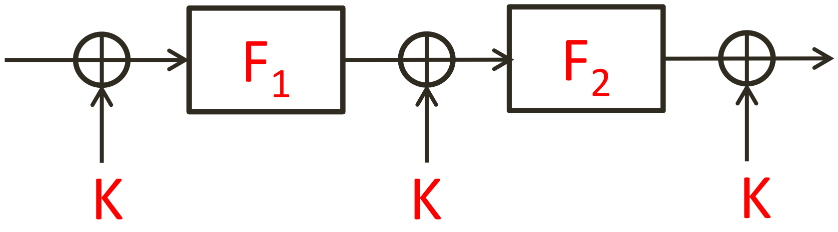


The LED Round



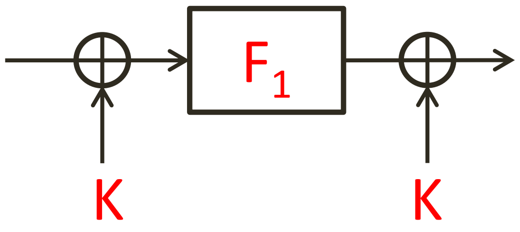
Previous Attacks on 2-Step LED

- Several previous attacks [MRTV'12,NWW'13,DDKS'13] require about 2^{60} **time** and **memory** and a lot of **data**
- [IS'12] requires 2^{56} **time** and 2^{8} and chosen plaintexts and a small amount of memory



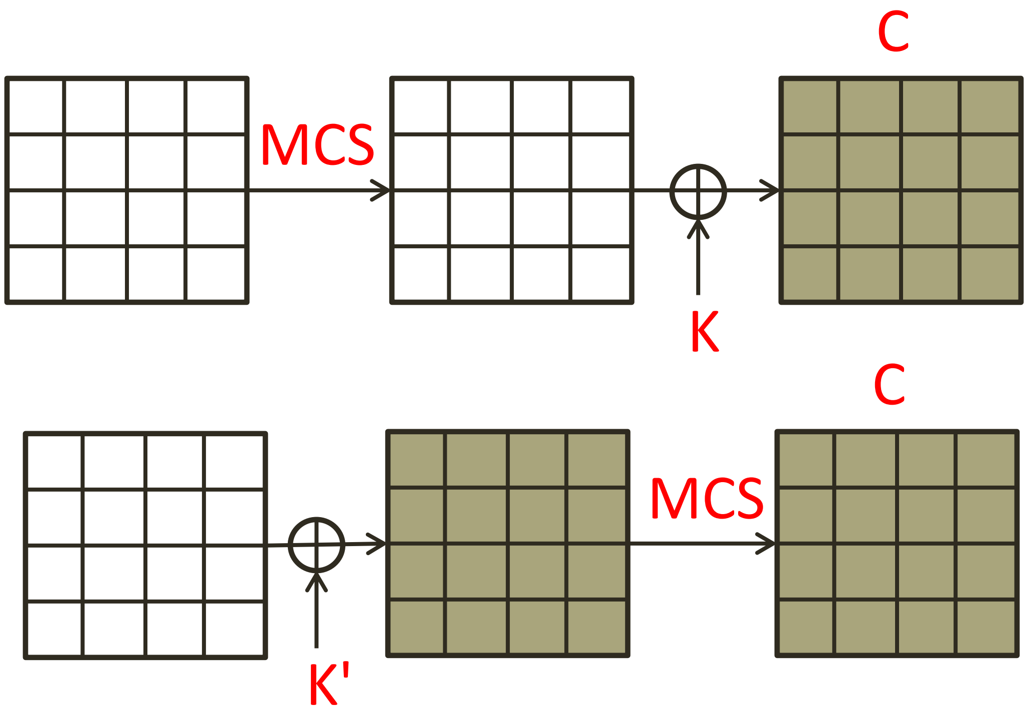
A MITM Attack on 1-Step LED [IS'12]

- [IS'12] is based on a **MITM** attack on 1-step LED-64 given a **single known** plaintext-ciphertext pair
- A similar attack **MITM** attack published by Sasaki in 2011
- Exploits a few well-known observations regarding the **structure** of AES-like ciphers



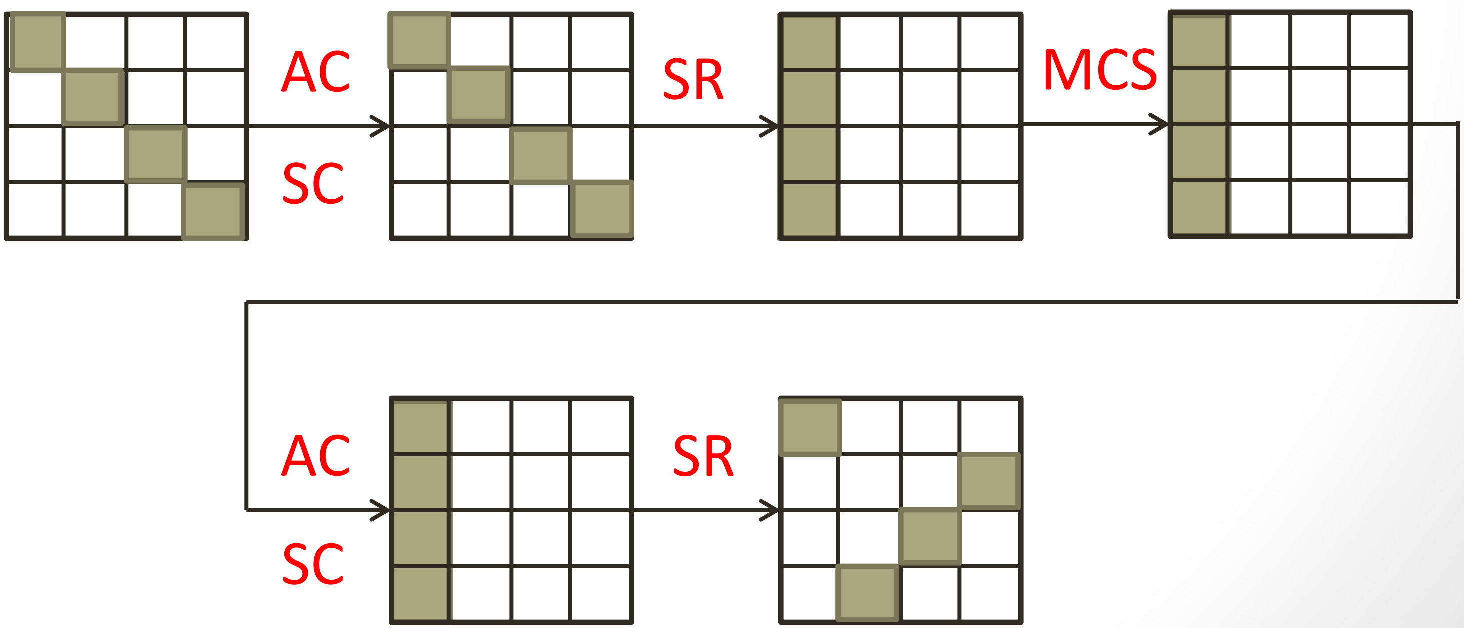
A MITM Attack on 1-Step LED [IS'12]

- Observation 1: The order of the **linear** operations ARK and MCS is interchangeable
- $MCS^{-1}(ARK^{-1}(C))=ARK'^{-1}(MCS^{-1}(C))$, where ARK' adds the key $K'=MCS^{-1}(K)$



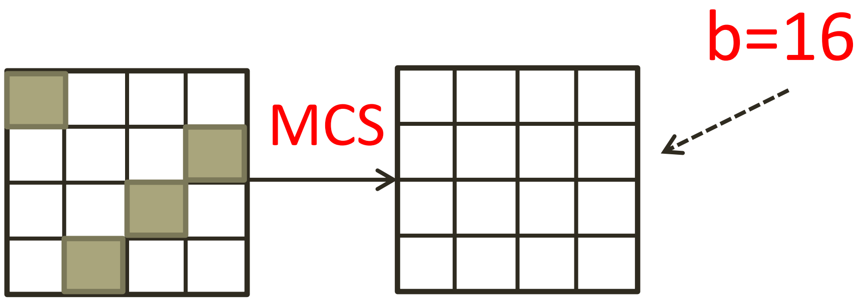
A MITM Attack on 1-Step LED [IS'12]

- Observation 2: Given an **inverse-diagonal** we can fully compute the **diagonal** of the state after the 7 operations (and vise-versa)
- This 4 nibble to 4 nibble mapping is called a **super-Sbox**

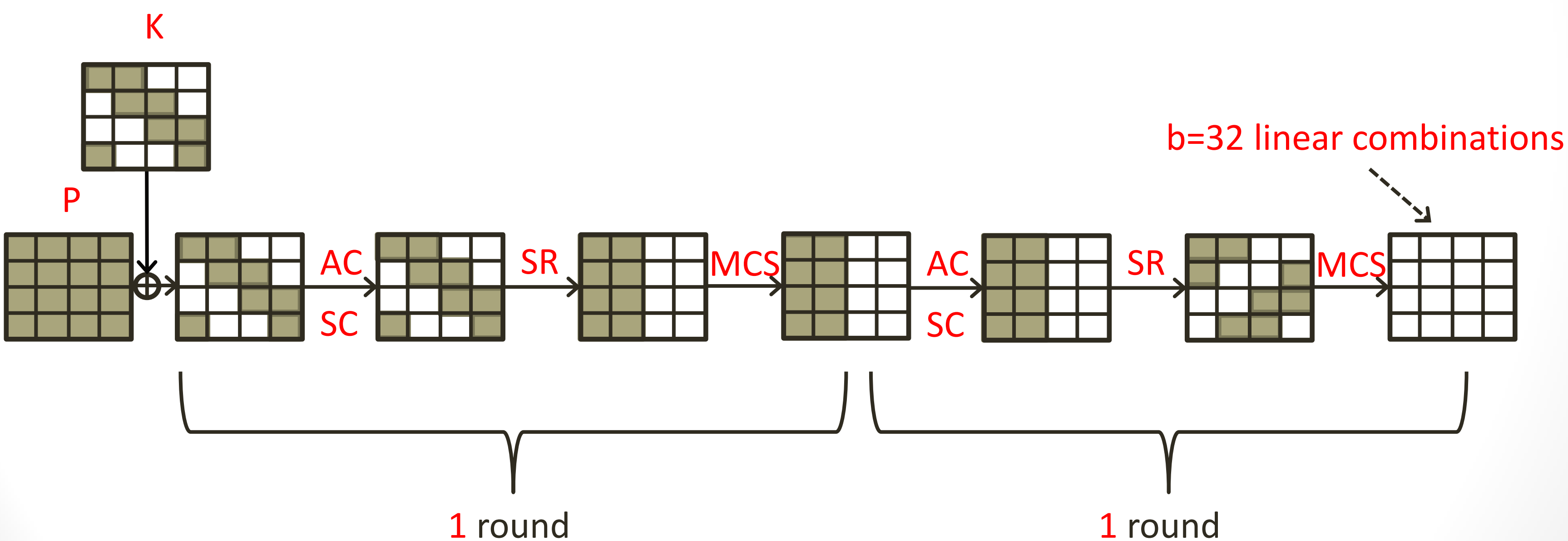


A MITM Attack on 1-Step LED [IS'12]

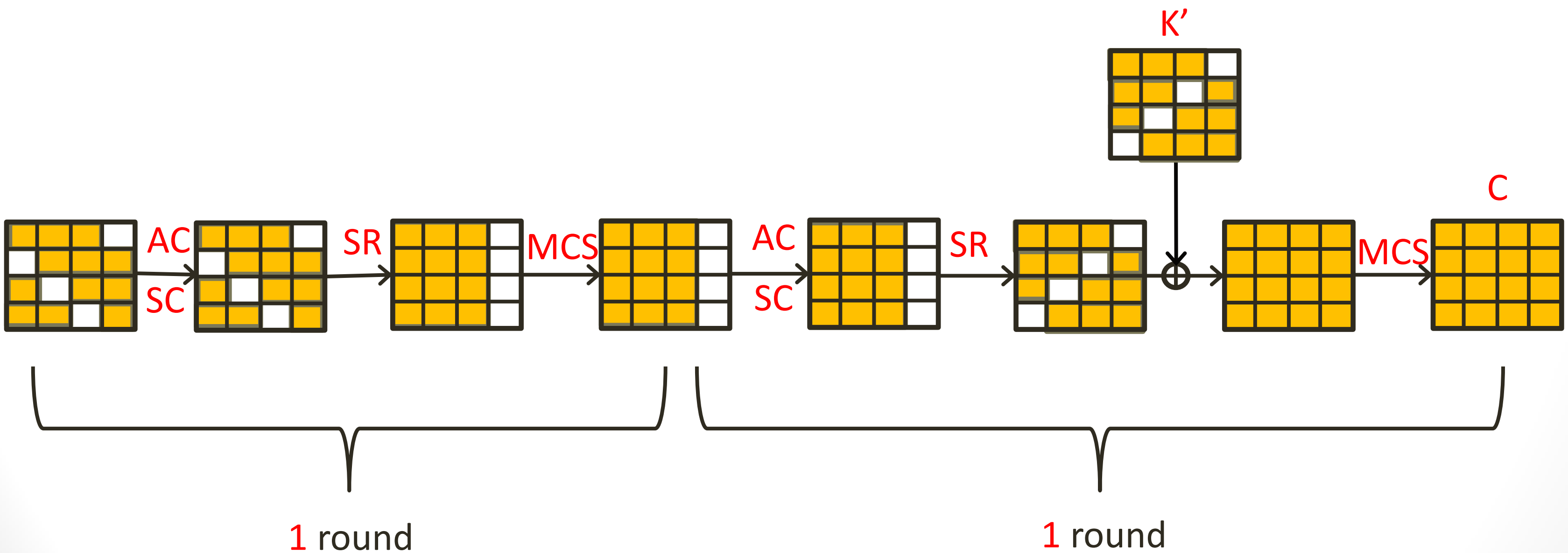
- Observation 3: Given knowledge of any b bits of the state X, we can compute the values of b **linear combinations** (over GF(2)) on the state MCS(X)



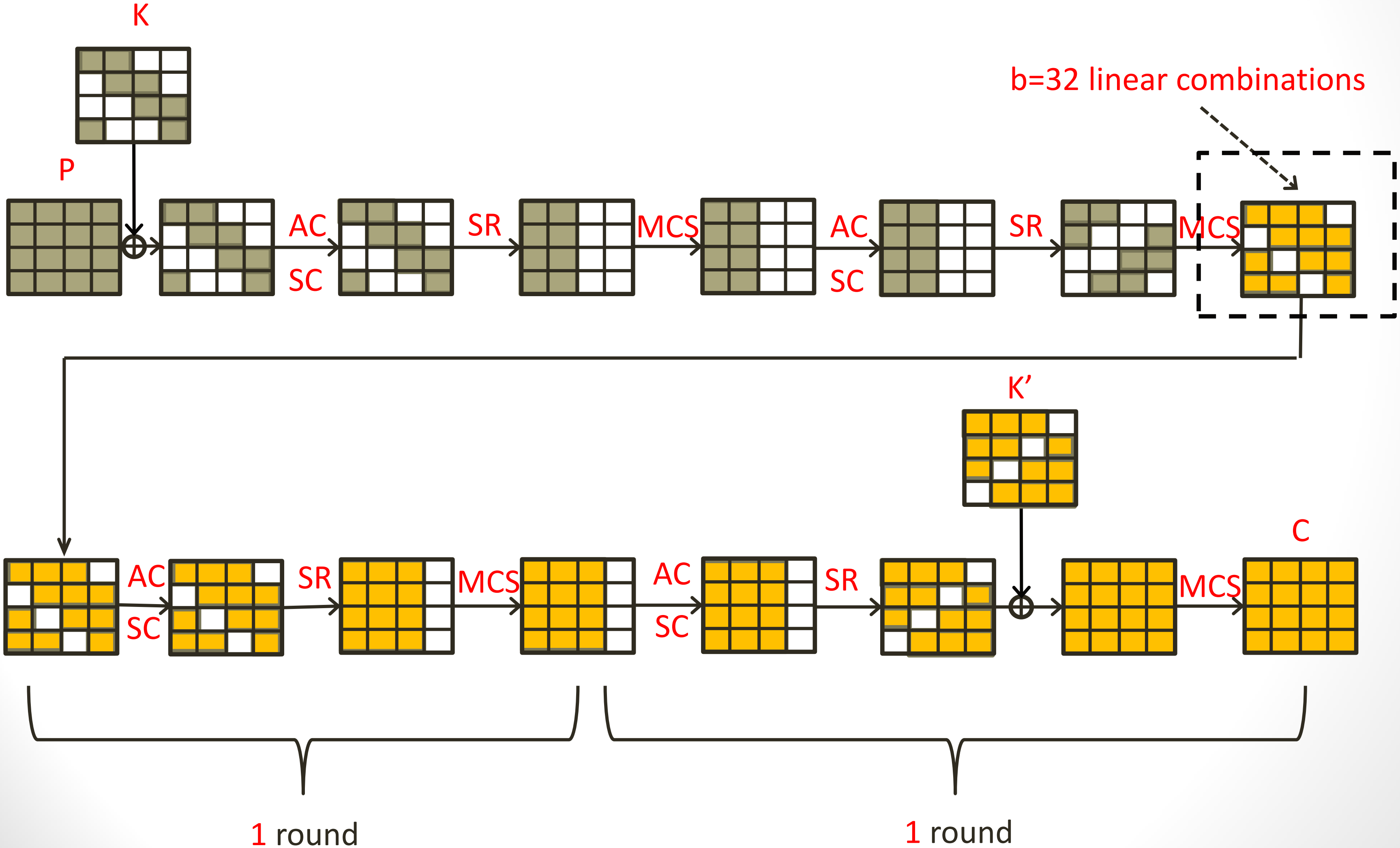
A MITM Attack on 1-Step LED [IS'12]



A MITM Attack on 1-Step LED [IS'12]

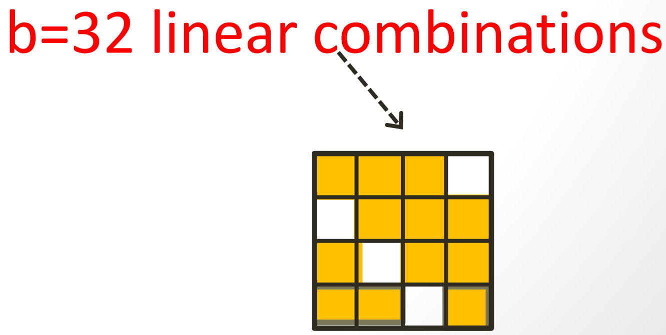


A MITM Attack on 1-Step LED [IS'12]



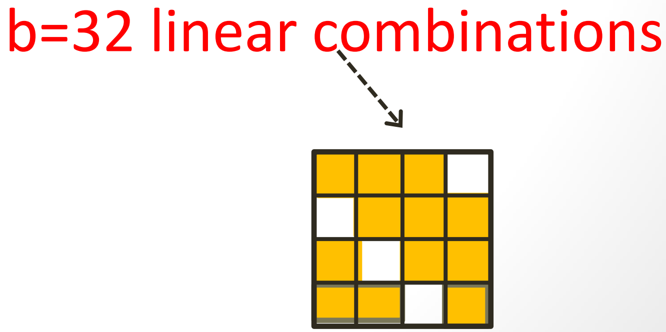
A MITM Attack on 1-Step LED [IS'12]

- From the **encryption** side we calculate 32 linear combinations on the state after 2 rounds
- From the **decryption** side we calculate 48 bits
- The linear subspaces **intersect** on a linear subspace of dimension 32+48-64=16
- 16 combinations of a **basis** for the intersection subspace are computable **independently** from both sides
- Typically called **indirect partial matching**



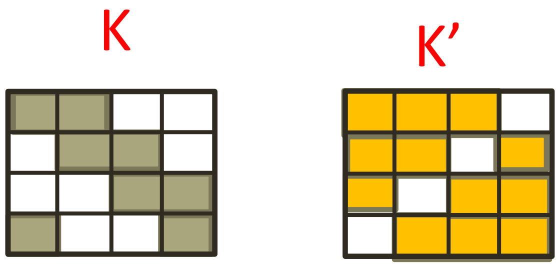
A MITM Attack on 1-Step LED [IS'12]

- We have 16 bits of the **sieving** on the **state**
- We guess 32 key bits from the **encryption** side
- We guess 48 key bits from the **decryption** side
- After **filtering** we remain with about $2^{32+48-16}=2^{64}$ **keys**
- The current form of the attack is not faster than **exhaustive search**



The New Linear Key Sieve

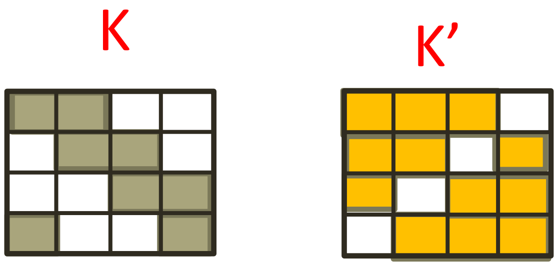
- We can add more filtering conditions by using more data, but this is **not required**
- We guess 32 bits of K from the **encryption** side and 48 bits of K' from the **decryption** side
- Since K and K' are related by a **linear function** we can factor out 32+48-64=16 linear combintations on the **key** computable **independently** from both sides
- We call these expressions a **linear key sieve**



The New Linear Key Sieve

- Similar techniques exploited linear message schedules of **hash functions** in **MITM** attacks [Aoki and Sasaki, CRYPTO'09]

- This is the **first time** that such **sieving** techniques are used on **block ciphers**

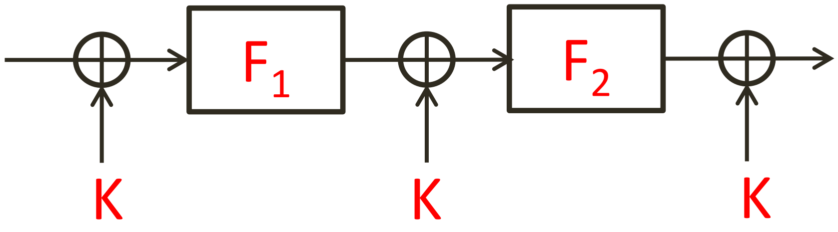


An Improved MITM Attack on 1-Step LED

- We have 16 bits of **sieve** on the **state**
- We have 16 bits of the **linear key sieve**
- Guess 32 key bits from the **encryption** side
  - Compute the 32 bits of filtering and store the suggestions in a sorted list L
- Guess 48 key bits from the **decryption** side
  - Compute the 32 bits of filtering, search L, and obtain a suggestion for the full key
- After **filtering** we need to test about $2^{32+48-16-16}=2^{48}$ **keys**
- We obtain an attack with time complexity 2^{48}

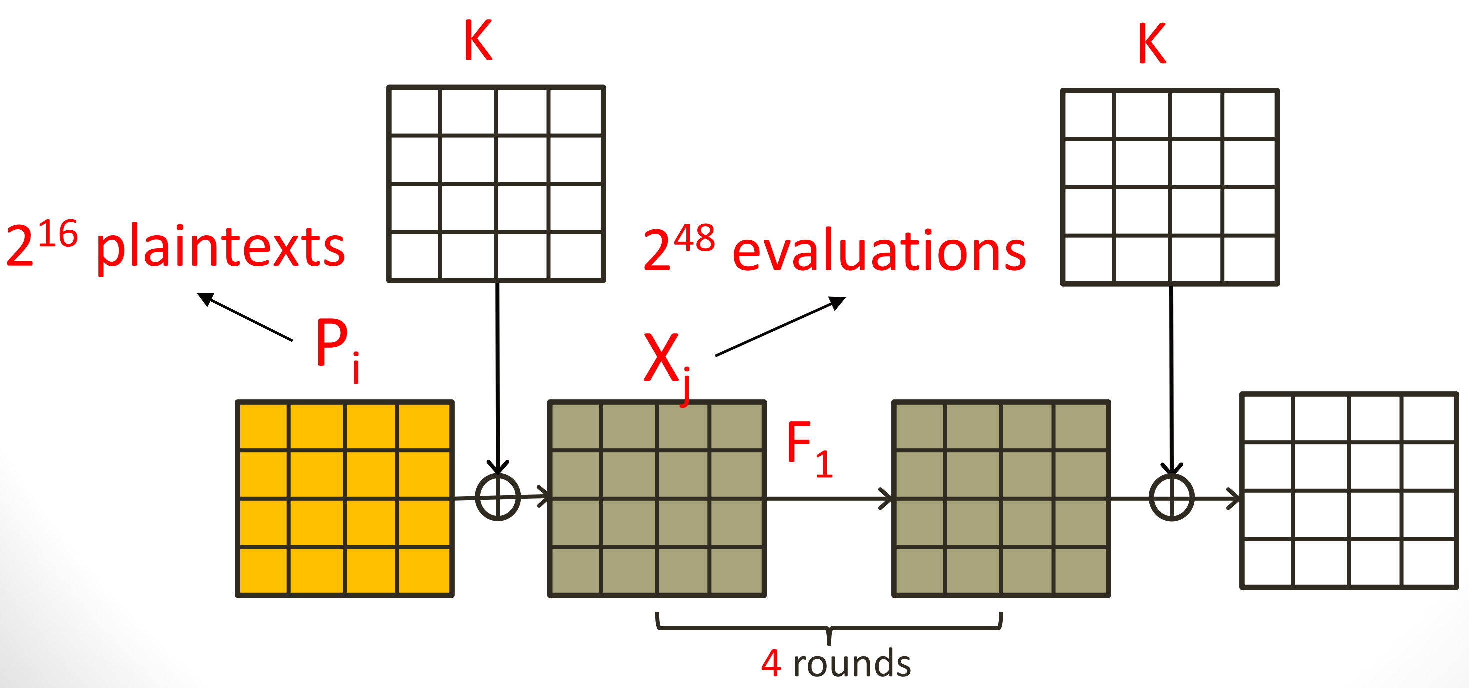
Splice-and-Cut (Aoki and Sasaki, 2008)

- In order to attack 2-step LED, we use the splice-and-cut technique (as the previous attack of [IS'12])



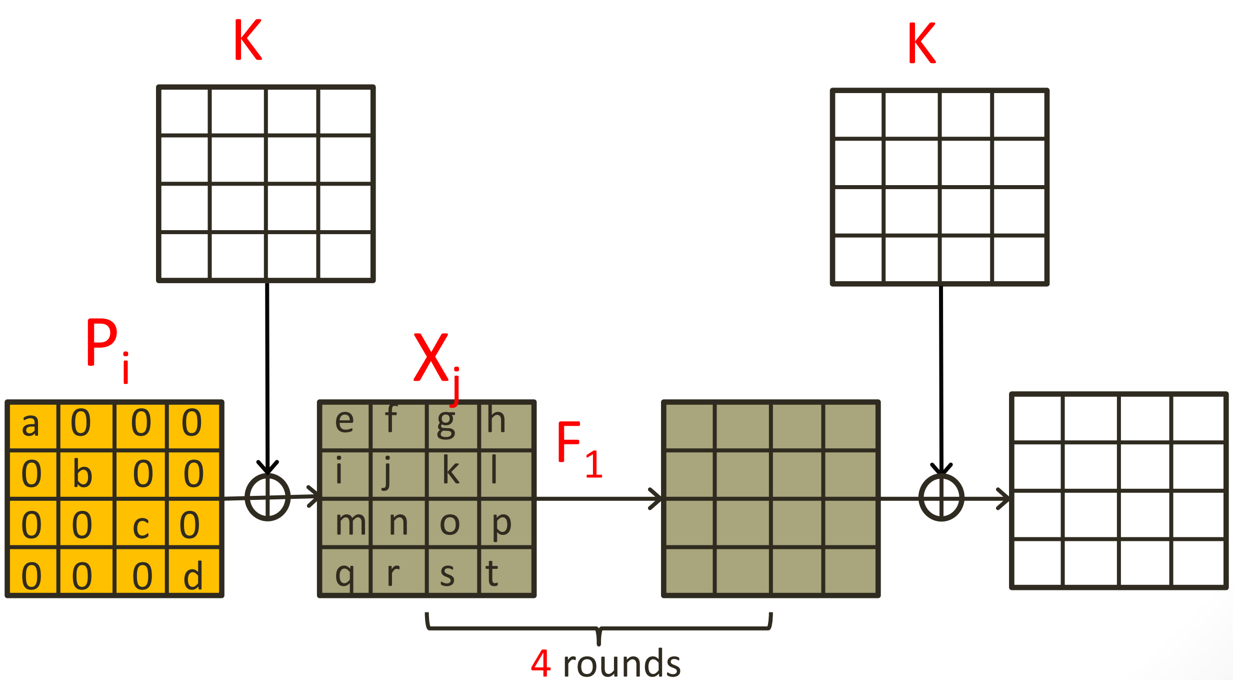
Splice-and-Cut on 2-Step LED-64

- We **choose** 2^{16} **plaintexts** P_i and evaluate F_1 on 2^{48} values X_j
- Each of the 2^{64} keys is covered by a **unique** (i,j) such that $P_i+X_j=K$



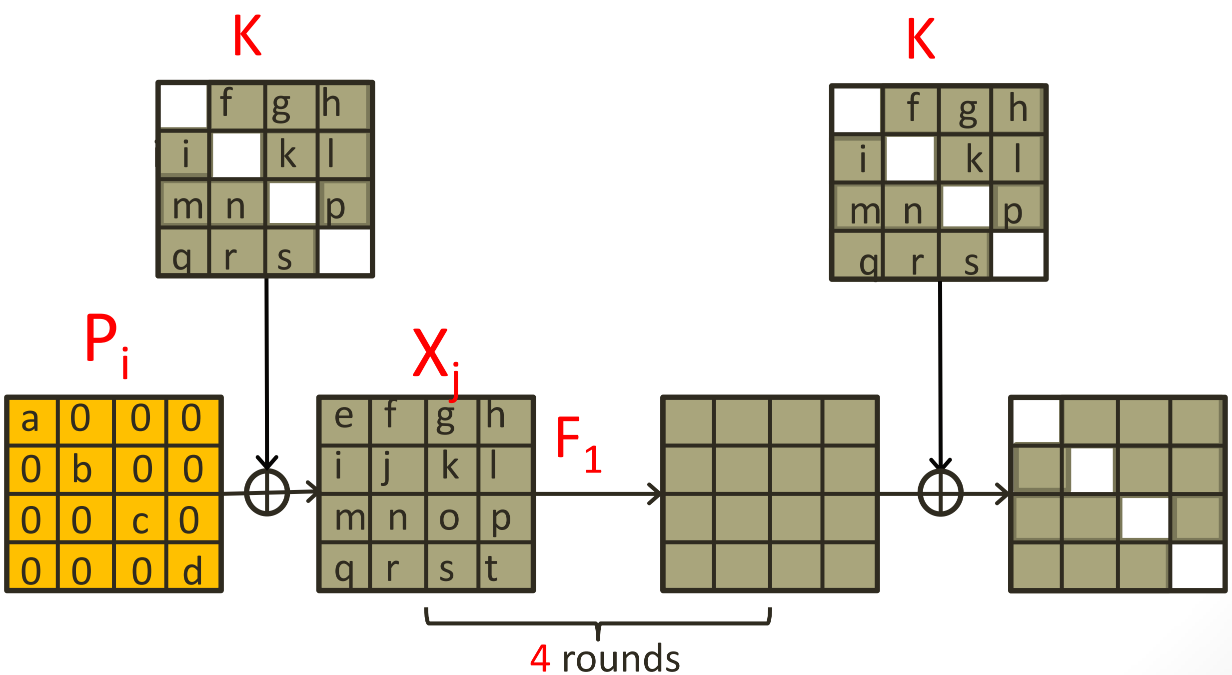
Splice-and-Cut on 2-Step LED-64

- Ask for **chosen plaintexts** P_i in which 3 inverse-diagonals are 0



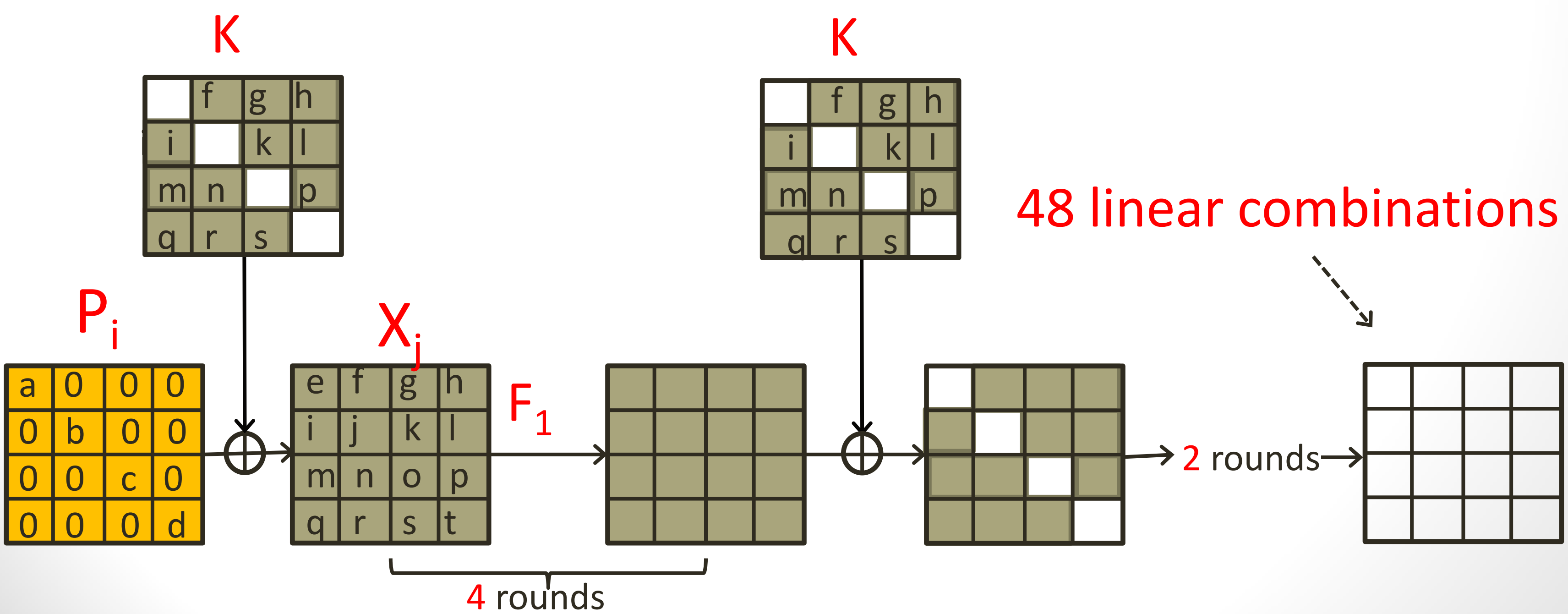
Splice-and-Cut on 2-Step LED-64

- $P_i+X_j=K$ implies that **for any** P_i: $K=X_j$ on the 3 inverse-diagonals
- Each X_j is **associated** with a value of K on the 3 inverse-diagonals

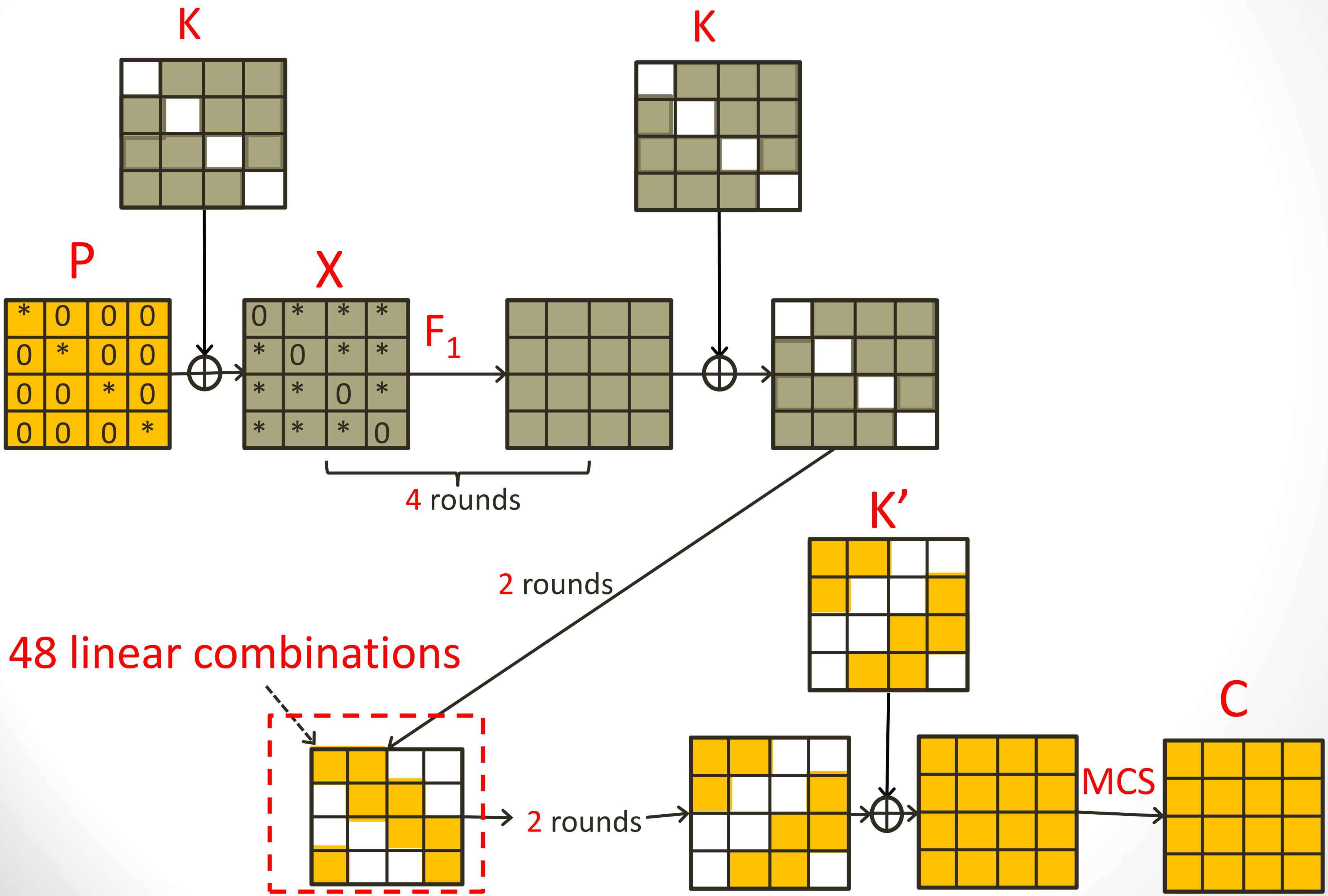


Splice-and-Cut on 2-Step LED-64

- For each X_j we can continue the evaluation and calculate 48 linear expression on the state after 6 rounds



Splice-and-Cut on LED-64

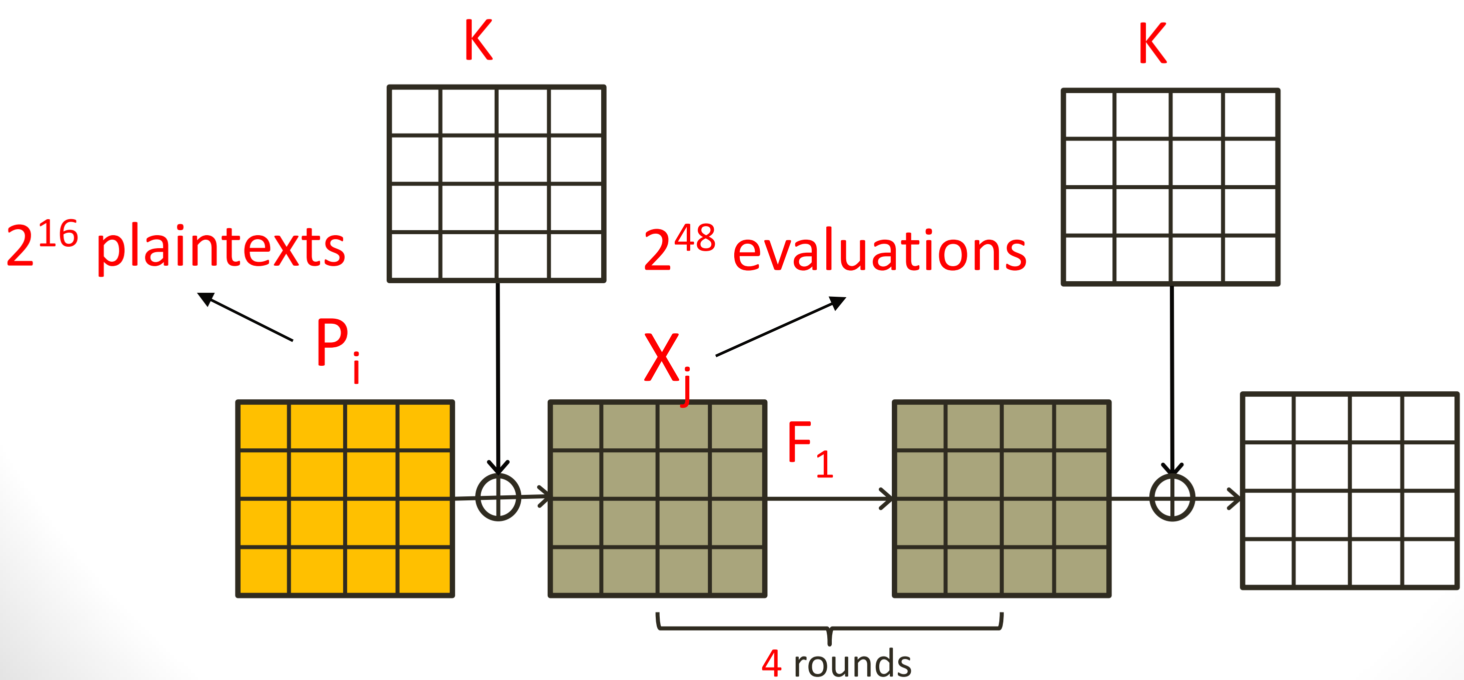


Splice-and-Cut on LED-64

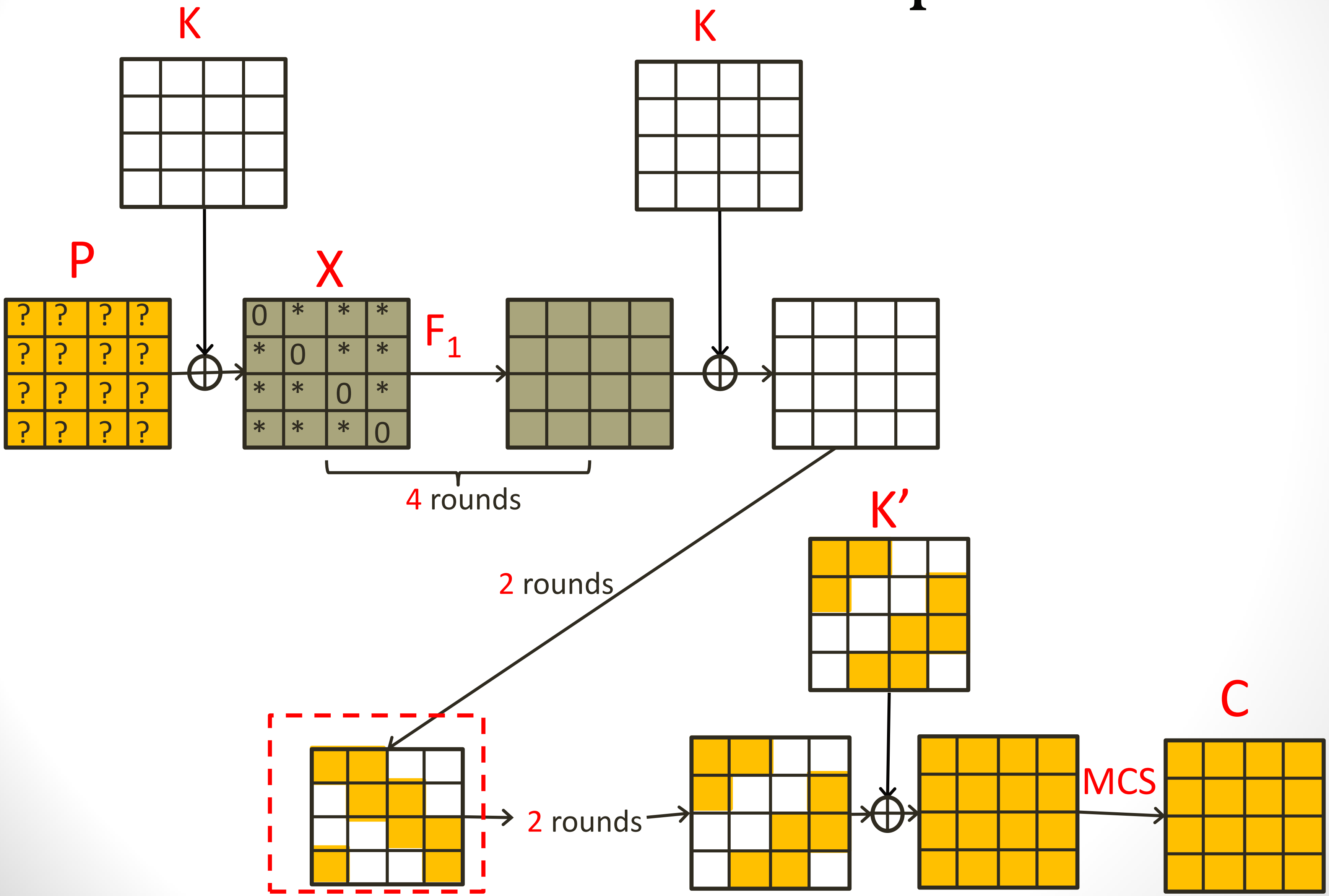
- Using the **sieve** on the state and the **linear key sieve**, we obtain an attack with **time** complexity 2^{48}
- The **data** complexity is 2^{16} chosen plaintexts
- The **memory** complexity is about 2^{16}

An Attempt to Obtain a Known Plaintext Attack on 2-Step LED-64

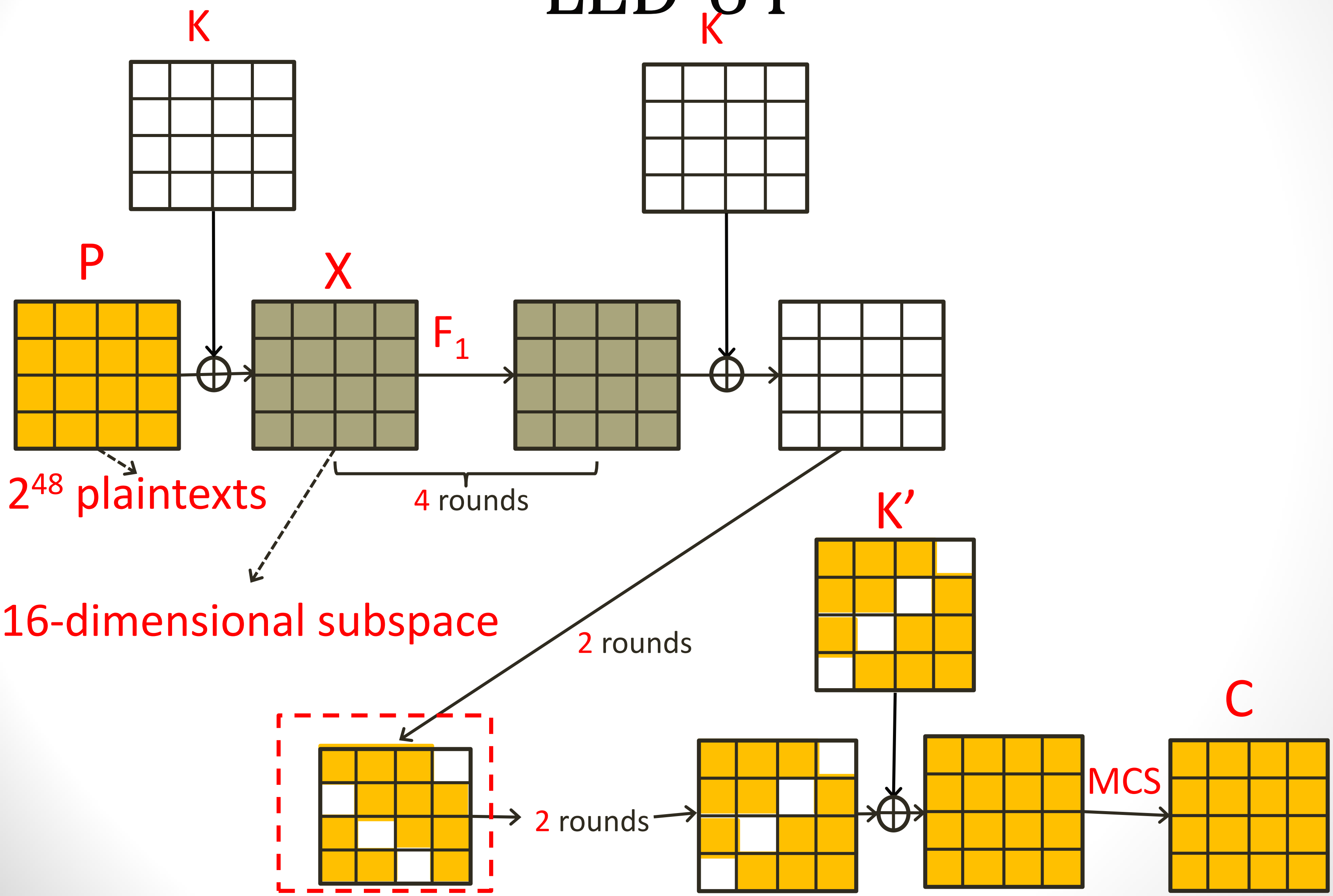
- We **obtain 2^{16} random plaintexts** and evaluate F_1 on 2^{48} values
- Each of the 2^{64} keys is covered **with high probability** by (i,j) such that $P_i+X_j=K$



An Attempt to Obtain a Known Plaintext Attack on 2-Step LED-64



The Known Plaintext Attack on 2-Step LED-64



The Known Plaintext Attack on 2-Step LED-64

- We need to **carefully reconstruct** the attack in order to obtain to obtain an efficient algorithm
- We obtain a **known plaintext splice-and-cut** attack on LED-64!
- The **time** complexity is 2^{48}, which is the **same** as for the chosen plaintext attack
  - The **data** and **memory** complexity are increased to 2^{48}

Conclusions

- We introduced the **linear key sieve** which exploits linear dependencies between **key bits** in **MITM** attacks on block ciphers
- We used this technique to efficiently apply for the first time a **splice-and-cut** attack in the **known plaintext** model
- We applied these techniques to obtain the best known attacks on 2-step LED-64
- We also obtained the best known attack on 3-step LED-64 in the **related-key** model

Thank you for your attention!