Improved Attacks on Full GOST

Itai Dinur[1], Orr Dunkelman[1,2] and Adi Shamir[1]

[1]The Weizmann Institute, Israel

[2]University of Haifa, Israel

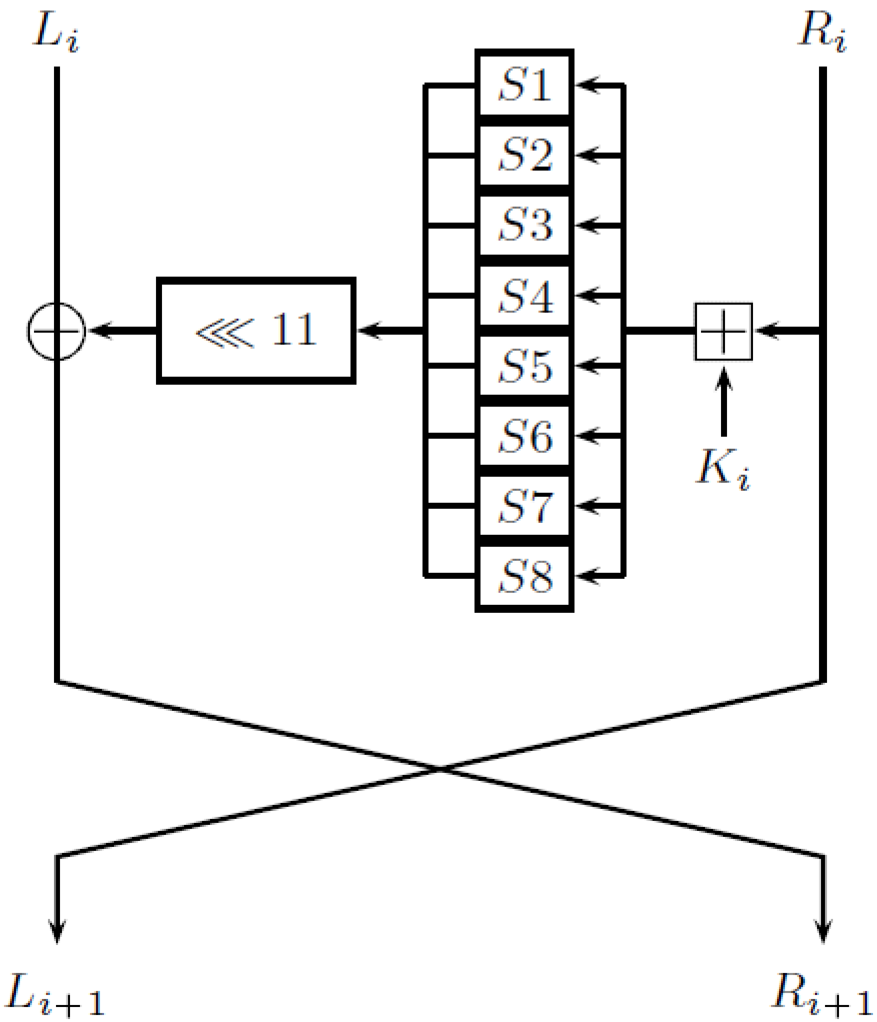
GOST

- Designed by Soviet cryptographers in the 1980’s
- Motivated by the desire to construct an **alternative** to DES
- Declassified in 1994

Design philosophy

- Like DES, a Feistel structure over 64-bit blocks
- Use **simpler components** compared to DES
- Try to get **higher** security
  - DES uses 56 bits of key and 16 rounds
  - GOST uses 256 bits of key and 32 rounds
- Does not specify the Sboxes

One Round of GOST



The Key Schedule

- Break the 256-bit key into 8 **subkeys** of 32 bits
- In the first 24 rounds the keys are used in their **cyclic order**
- In the final 8 rounds the round keys are used in **reverse order**
  - Perhaps to avoid slide attacks

K1,K2,…,K8,K1,K2,…,K8,K1,K2,…K8 ,K8,K7,…,K1

Previous Single Key Attacks

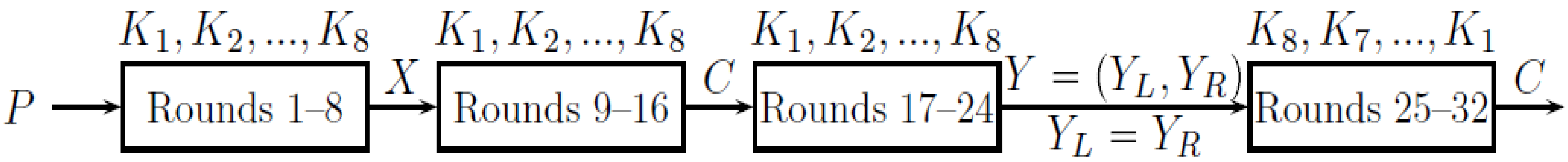
- In 2011 Isobe published the **first single key attack** on full GOST
  - Data 2^{32}, Time 2^{224} , Memory 2^{64}
  - Based on the **reflection** self-similarity property of GOST (Kara 2008)
  - Uses a **meet-in-the-middle** attack
  - Requires invertible Sboxes
- Several attacks were later published by Courtois
  - Their complexity was evaluated for the Sboxes used by Russian banks
  - It is expected that the attacks have similar complexities for other choices of Sboxes (C'12)

Self-Similarity Properties Used in Our Attacks

- The **reflection** property (Kara 2008)
- A new **fixed point** property (independently discovered by Courtois'11)

- Reduce attacking 32-round GOST to attacking 8-round GOST **given 2 input-output pairs**

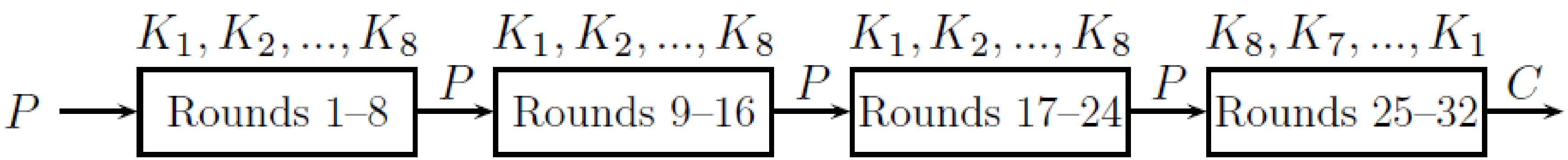
The Reflection Property (Kara 2008)



- Requires about 2^{32} known plaintext-ciphertext pairs
- Guess the 64-bit value X
- Altogether, apply the 8-round attack 2^{96} times

- We have another **"half pair"** since we know that the two sides of Y are equal
- We do not know how to efficiently exploit this information

The Fixed-Point Property
(independently discovered by Courtois'11)



- Requires about 2^{64} known plaintext-ciphertext pairs (the full codebook)
- Apply the 8-round attack 2^{64} times

- Given $c \cdot 2^{64}$ known plaintexts for $c<1$, this fixed point occurs with probability c
- The success probability is **reduced by c**

Given Two 8-Round Input-Output Pairs

- 128-bit constraint
- The 8-round attacks **leave $2^{256-128}=2^{128}$ keys**
- Need to test the remaining 2^{128} keys
- The time complexity of the 8-round attacks is **at least 2^{128}**

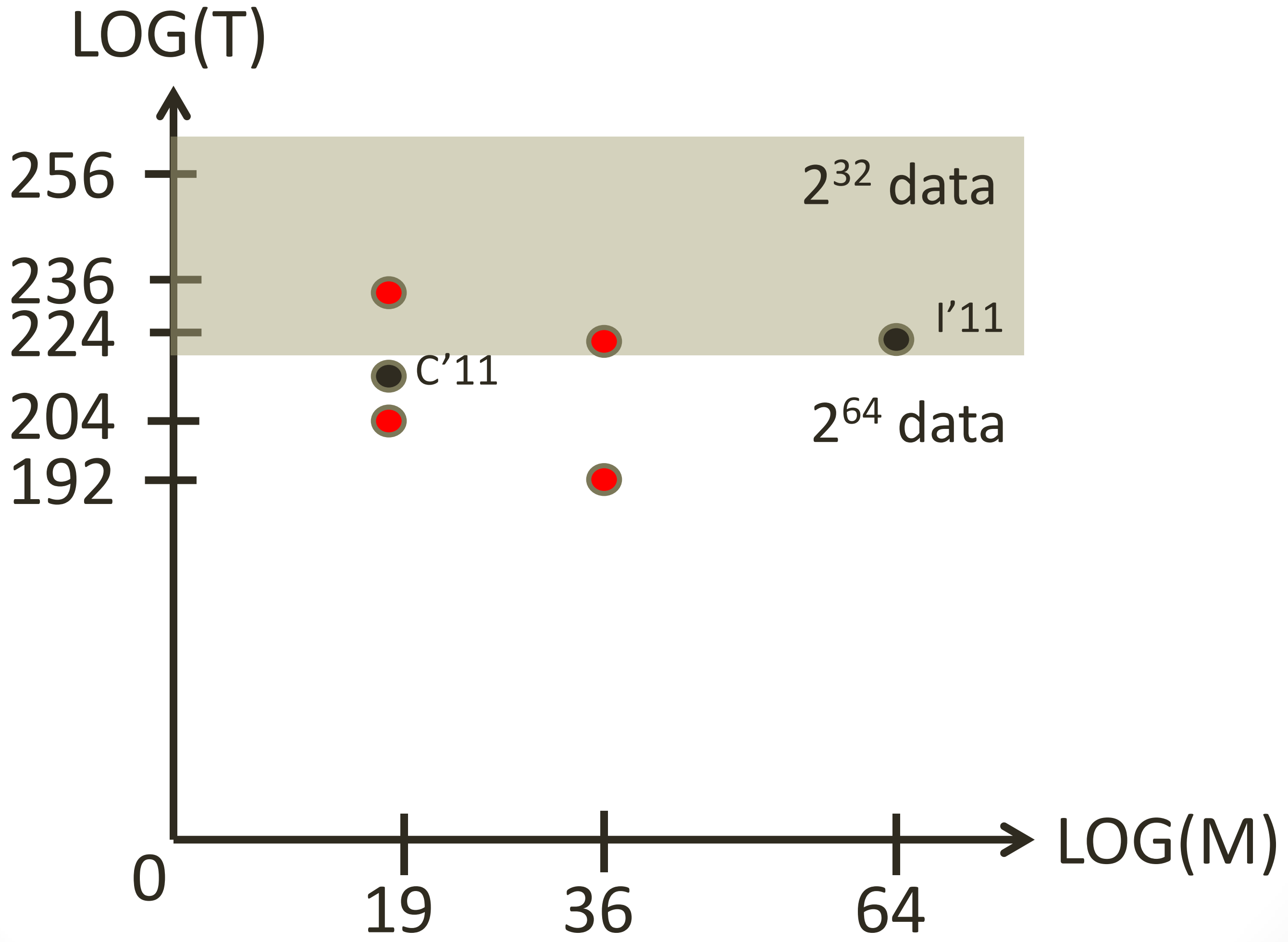
8-Round Attacks

- A **basic** meet-in-the-middle (MITM) attack
  - Time 2^{128}, memory 2^{128}
- A **more efficient** MITM attack
  - Time 2^{128}, memory 2^{64}
  - A variant of Isobe's attack
  - Combined with the reflection property, gives an attack on full GOST with the same parameters as Isobe's
- A **new** low-memory attack
  - Time 2^{140}, memory 2^{19}
- A **new** 2-dimensional meet-in-the-middle (**2DMITM**) attack
  - Time 2^{128}, memory 2^{36}

Attacks on Full GOST

- Select one of the two self-similarity properties for the outer loop:
  - If we have 2^{64} **data**, select the **fixed point** property
  - If we have 2^{32} **data**, select the **reflection** property
- Select one of last two 8-round attacks:
  - If we have 2^{36} **memory**, select the **2DMITM** attack
  - If we have 2^{19} **memory** (fits cache), select the **low-memory** attack
- Altogether we obtain 4 attacks on full GOST

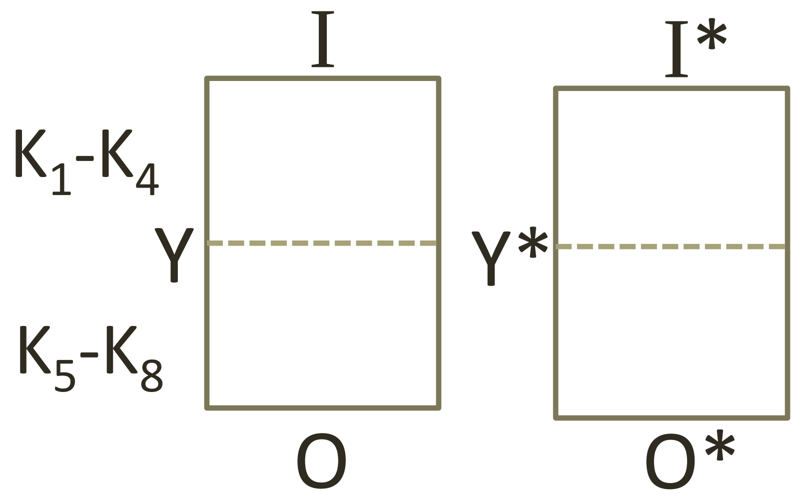
Attacks on Full GOST



8-Round Attacks

- **A basic meet-in-the-middle (MITM) attack**
  - **Time 2^{128}, memory 2^{128}**
- A more efficient MITM attack
  - Time 2^{128}, memory 2^{64}
  - A variant of Isobe's attack
  - Combined with the reflection property, gives an attack on full GOST with the same parameters as Isobe's
- A new low-memory attack
  - Time 2^{140}, memory 2^{19}
- A new 2-dimensional meet-in-the-middle attack
  - Time 2^{128}, memory 2^{36}

A Basic MITM Attack

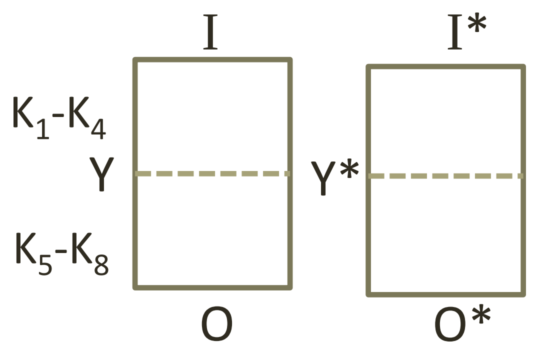


- For each 128-bit value of K_1-K_4
  - Partially encrypt I and I*, and store the 128-bit suggestions for Y and Y* in a **sorted list**
- For each 128-bit value of K_5-K_8
  - Partially decrypt O and O*, and look for **matches in the list**
  - For each match **test the full key**
- **Time 2^{128}, memory 2^{128}**

8-Round Attacks

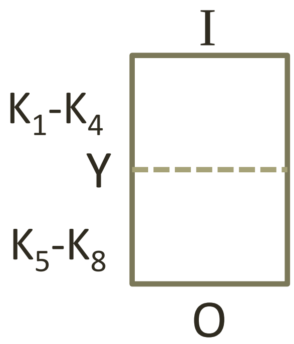
- A basic meet-in-the-middle (MITM) attack
  - Time 2^{128}, memory 2^{128}
- **A more efficient MITM attack**
  - **Time 2^{128}, memory 2^{64}**
  - **A variant of Isobe's attack**
  - **Combined with the reflection property, gives an attack on full GOST with the same parameters as Isobe's**
- A new low-memory attack
  - Time 2^{140}, memory 2^{19}
- A new 2-dimensional meet-in-the-middle attack
  - Time 2^{128}, memory 2^{36}

A More Efficient MITM attack



- For each 64-bit value of Y
  - Use a **4-round attack** to obtain suggestions for K_1-K_4 given (I,Y) in time 2^{64}
  - Independently obtain suggestions for K_5-K_8 given (Y,O)
  - Store the suggestions in **two lists of size $2^{128-64}=2^{64}$**
  - Perform a **basic MITM** attack on (I*,O*) **using the keys stored in the lists**
- **Time $2^{64}\cdot 2^{64}=2^{128}$, memory 2^{64}**

A More Efficient MITM attack
The 4-Round Attack

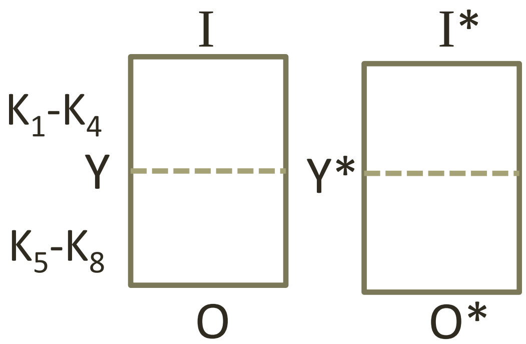


- Given (I,Y) perform a **basic MITM** attack to obtain 2^{64} suggestions for K_1-K_4
  - Repeat independently for K_5-K_8

8-Round Attacks

- A basic meet-in-the-middle (MITM) attack
  - Time 2^{128}, memory 2^{128}
- A more efficient MITM attack
  - Time 2^{128}, memory 2^{64}
  - A variant of Isobe's attack
  - Combined with the reflection property, gives an attack on full GOST with the same parameters as Isobe's
- **A new low-memory attack**
  - **Time 2^{140}, memory 2^{19}**
- A new 2-dimensional meet-in-the-middle attack
  - Time 2^{128}, memory 2^{36}

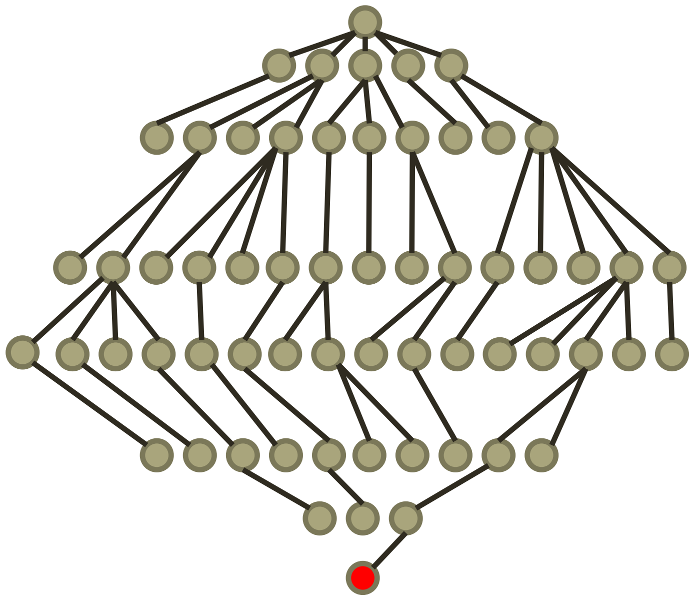
The Low Memory Attack



- For each 128-bit value of K_5-K_8
  - Partially decrypt O and O* and obtain **two 4-round input-output pairs (I,Y) and (I*,Y*)**
  - Execute a **4-round "Guess and Determine" routine** to obtain suggestions for the (expected number of) $\mathbf{2^{128-128}=1}$ **key**

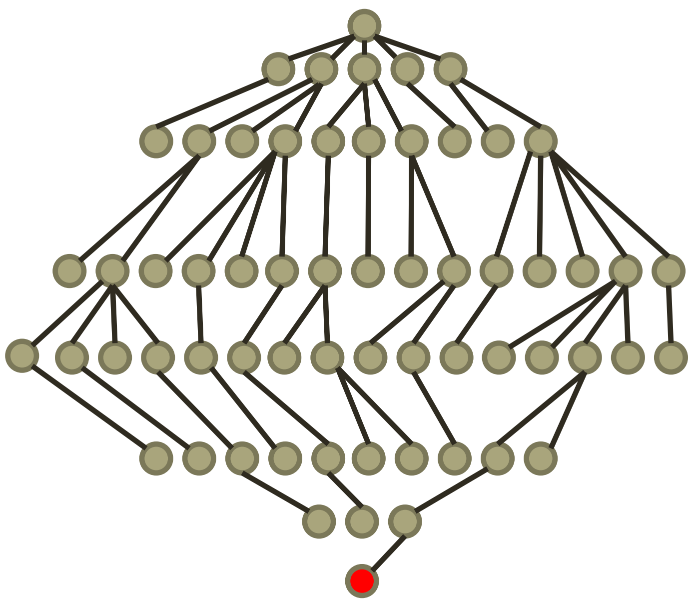
The 4-Round "Guess and Determine" Routine

- Exploits the **slow diffusion** of the **key** into the state
- Traverse a **layered tree** of partial guesses for K_1-K_4
- The nodes in each layer specify guesses for a certain **subset of the key bits**
  - The nodes of the last layer contain guesses for K_1-K_4



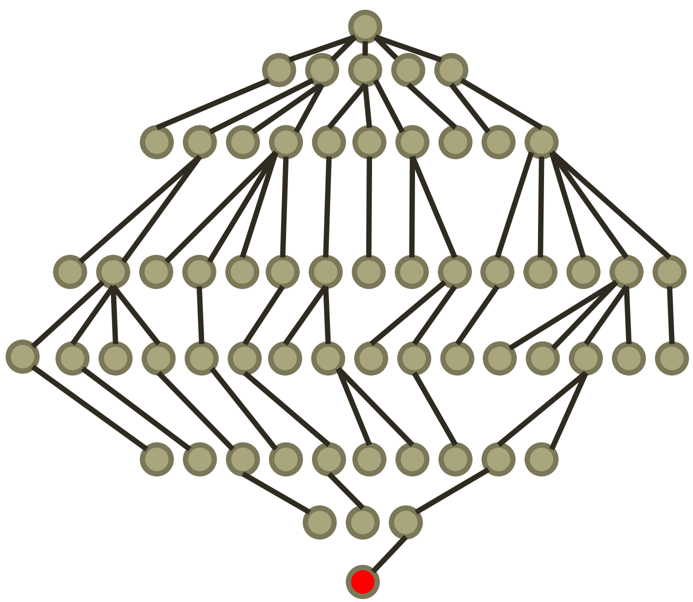
The 4-Round "Guess and Determine" Routine

- Expand a node by guessing the values of a small number of **additional key bits**
  - Calculate intermediate encryption bits from **both sides** of the block cipher
  - **Discard** nodes for which the values **do not match**



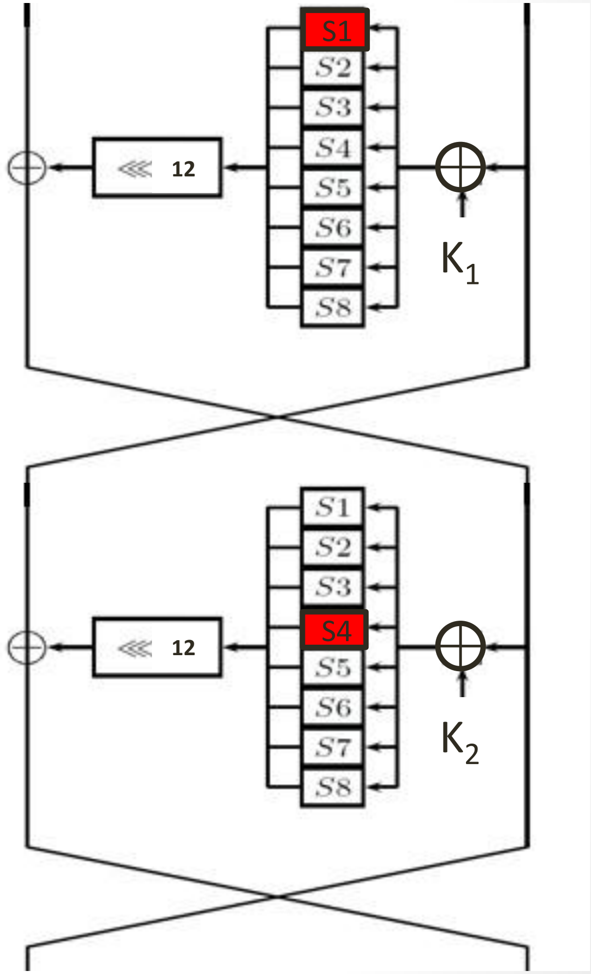
The Size of the Tree

- The time complexity is proportional to the **number of nodes**
- **Minimize** the number of nodes by guessing the **smallest number** of bits in each layer
- Use DFS to minimize memory complexity



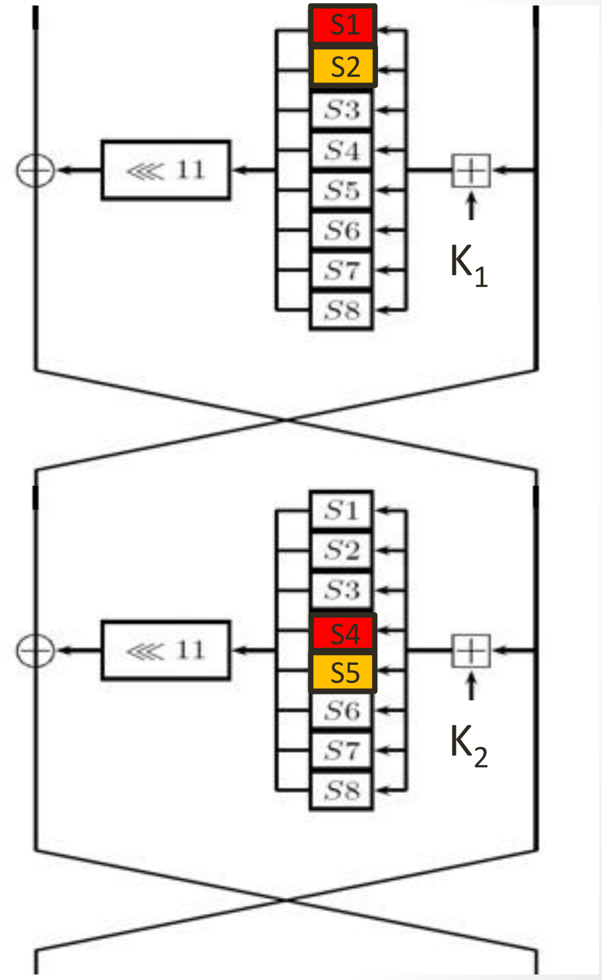
The “Guess and Determine” Routine
S-GOST

- A simplified version of GOST
- The layer procedure: work on **4-bit chunks**
- **Discard** wrong key guesses by evaluating 4 state bits from **both sides**
- The procedure of each layer is basically the same and is called an **iteration**
  - 8 iterations to recover the key



The "Guess and Determine" Procedure
Real GOST

- Guess **additional carry** and **state bits**
- The iterations are performed in their **natural order**
- Guess carries only in the **first iteration**
  - In the remaining iterations they are known
  - We pay for **state** bit guesses only in the **first iteration**



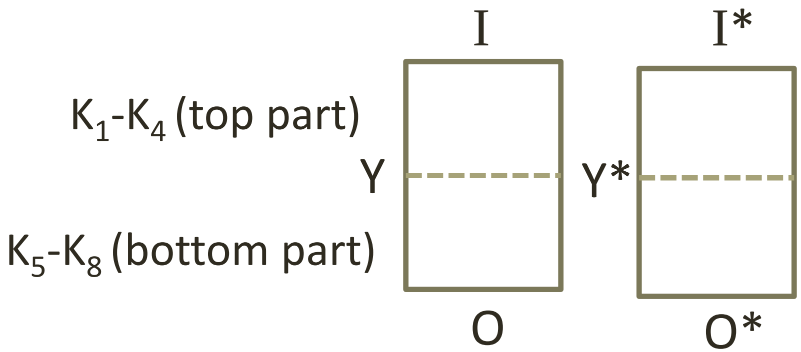
The Low Memory Attack
Complexity Analysis

- The **"Guess and Determine"** routine
  - **Time 2^{12}, memory 2^{19}** (using tables computed once and for all)
- The **low memory** attack
  - **Time $2^{128} \cdot 2^{12} = 2^{140}$, memory 2^{19}**

8-Round Attacks

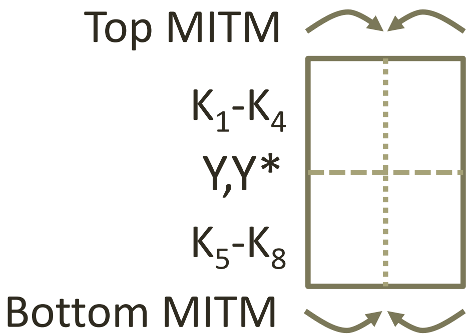
- A basic meet-in-the-middle (MITM) attack
  - Time 2^{128}, memory 2^{128}
- A more efficient MITM attack
  - Time 2^{128}, memory 2^{64}
  - A variant of Isobe's attack
  - Combined with the reflection property, gives an attack on full GOST with the same parameters as Isobe's
- A new low-memory attack
  - Time 2^{140}, memory 2^{19}
- **A new 2-dimensional meet-in-the-middle attack**
  - **Time 2^{128}, memory 2^{36}**

The 2-Dimensional Meet-in-the-Middle Attack (2DMITM)



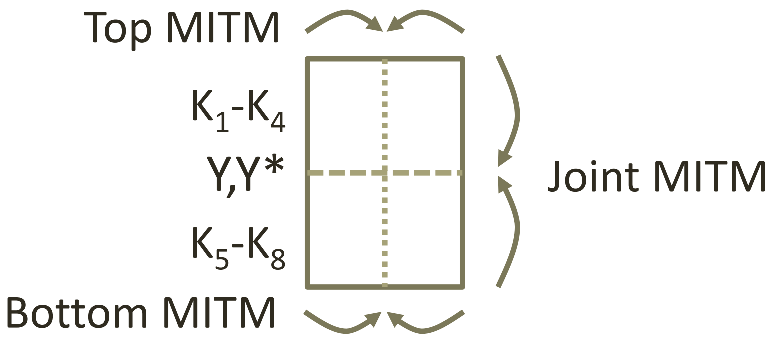
- Exploit **slow avalanche** of **state bits**
- Do not guess K_5-K_8 in advance
- Run 4 out of 8 iterations of the “Guess and Determine" attack with **knowledge** of 82 out of 128 bit of Y and Y*

The 2-Dimensional Meet-in-the-Middle Attack



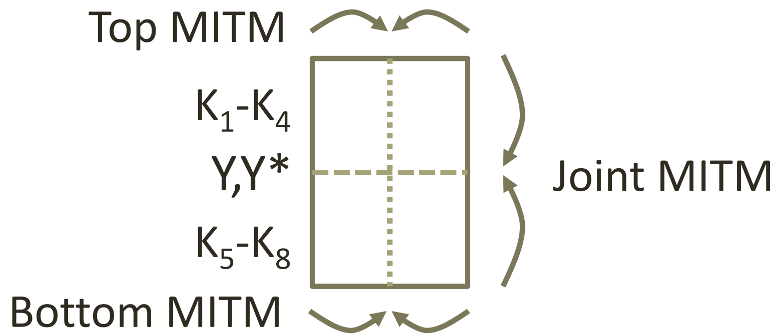
- Split each “Guess and Determine" attack into **two partial 4-round attacks**
- Run each attack for all possible values of Y and Y* it requires (2^{82} times)
- Run MITM attacks to **combine** the suggestions of the partial attacks to suggestions for the **4-round keys**

The 2-Dimensional Meet-in-the-Middle Attack



- **Join** the values suggested by the **top and bottom parts** to obtain suggestions for the **full key** using a **final MITM attack**
- We did not **filter** any keys in the top and bottom 4-round attacks
  - The attack requires 2^{128} memory
- The partial 4-round attacks take at most 2^{18} time and are executed 2^{82} times

The 2-Dimensional Meet-in-the-Middle Attack



- Since $2^{18} \cdot 2^{82} = 2^{100} << 2^{128}$ the 4-round attacks are **not the bottleneck**
- We guess bits of Y, Y* in advance **without increasing** the 2^{128} time complexity of the attack
- The 4-round attacks give **fewer suggestions** for the top and bottom keys which we need to store
- **Total time 2^{128}, memory 2^{36}**

Conclusions

- We presented **improved** attacks on full GOST
- Use **new** techniques
  - The **fixed point property** (Independently discovered by Courtois)
  - The **new 2DMITM** attack

Future Work

- Efficiently exploit the **“half pair”** in the reflection-based attacks
- New applications of **2DMITM**

Thank You For Your Attention!
Spasibo!