

# 100 years of Cryptanalysis: Compositions of Permutations

(Non-Commutative)

1918 => Modern Block Ciphers!

$$AD = CPNP^{-1}QP^{-1}P^3NP^{-4}QP^4N^{-1}P^{-4}C^{-1}$$

complicated?

Nicolas T. Courtois

University College London, UK



Rejewski



Zygalski

# or How Some Mathematicians Won the WW2...



Turing



Welchman

# Encryption

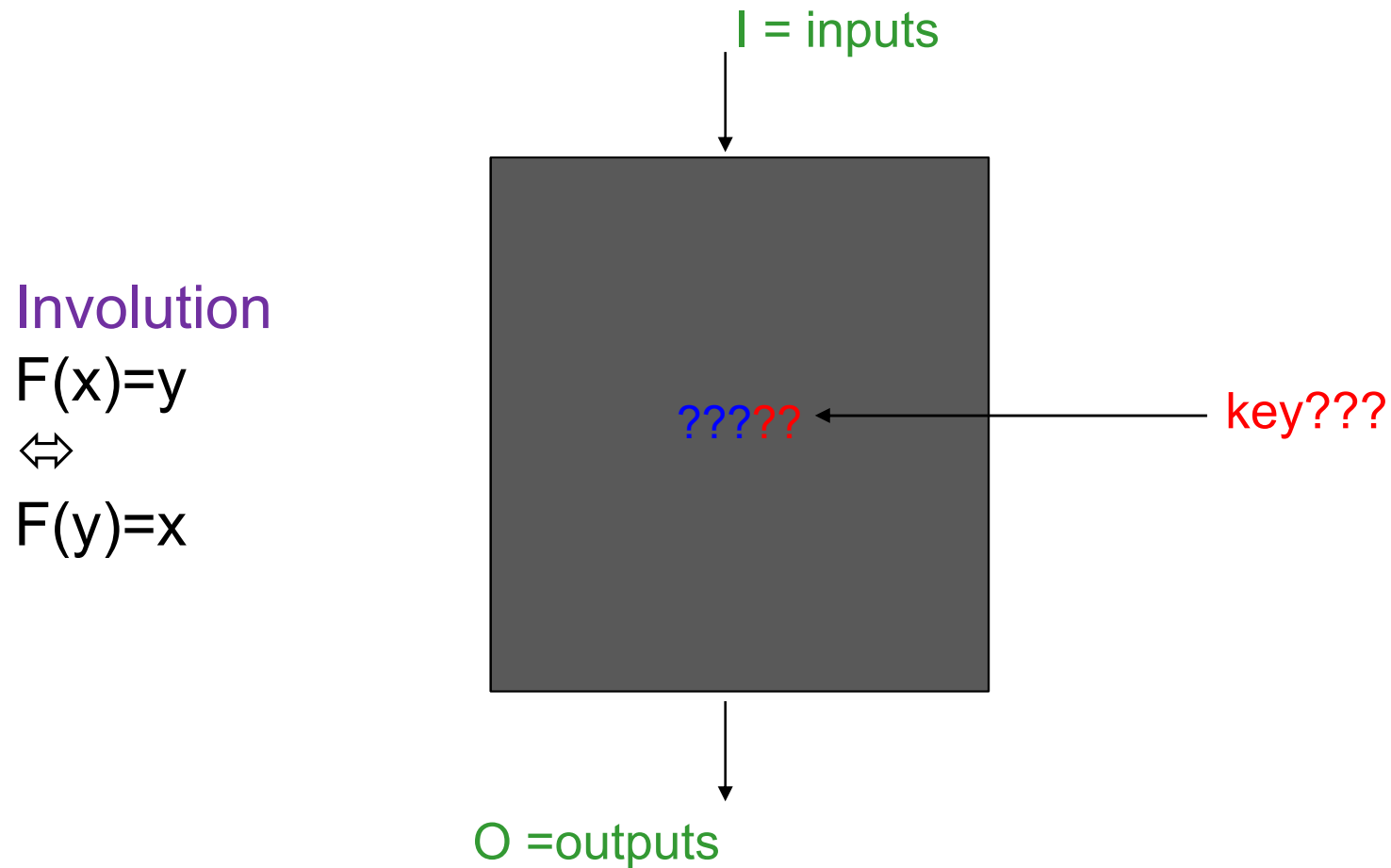
ciphertext

plaintext

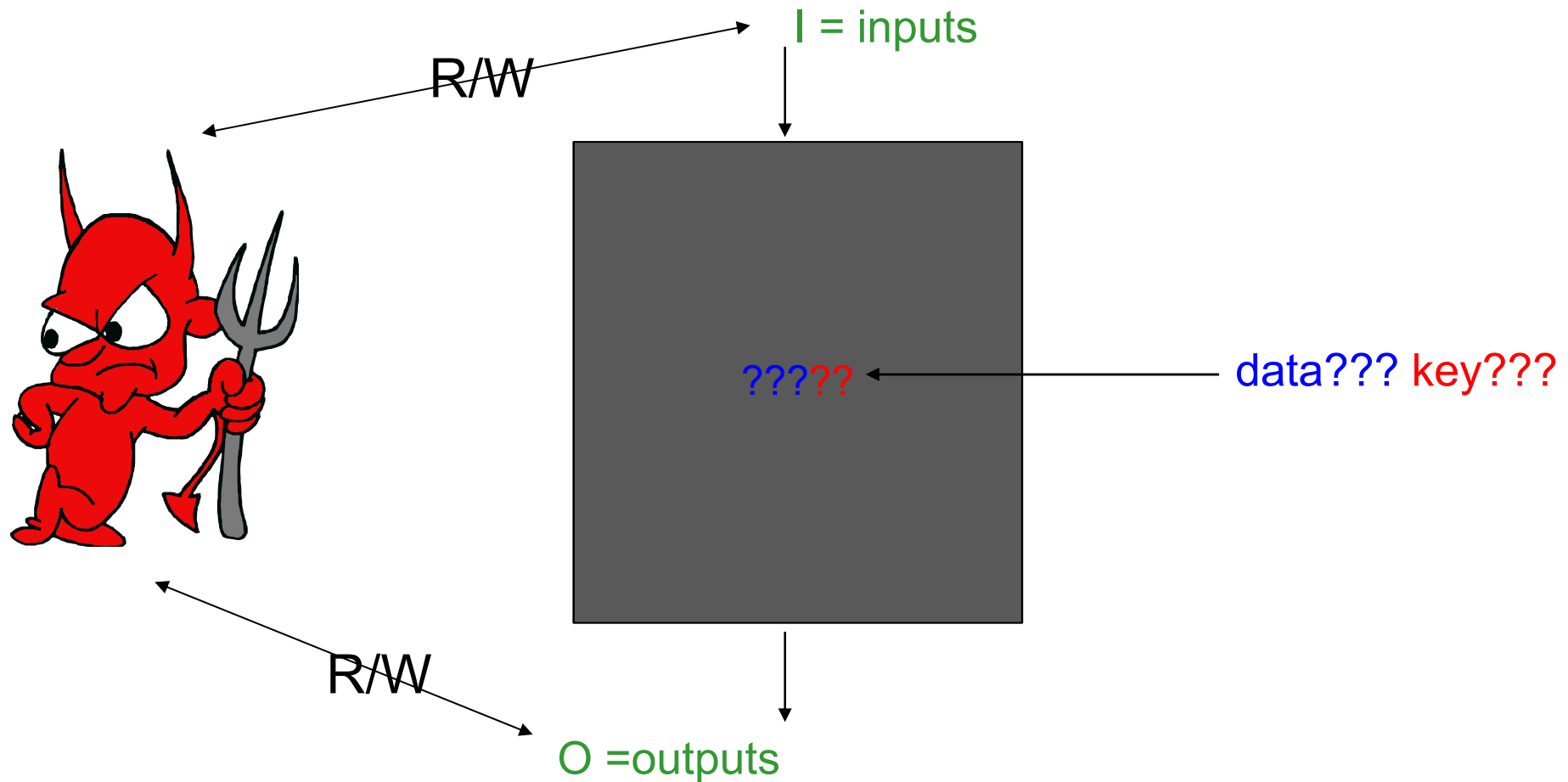


- self-reciprocity = **involution** pty
- no letter encrypted to itself

## Think Inside the Box



## The Box - General Setting



# Algebraic Cryptanalysis [Shannon]

Breaking a « good » cipher should require:

“as much work as solving a system of simultaneous equations in a large number of unknowns of a complex type”

**[Shannon, 1949]**



## Bottom Line

From FIRST cipher machines in 1920s to  
today's block ciphers, cryptanalysis has  
NOT changed so much (!!).

Same guiding principles.

## Motivation

Linear and differential cryptanalysis usually require **huge quantities** of known/chosen plaintexts.

Q: What kind of cryptanalysis is possible when the attacker has  
**only one known plaintext** (or very few) ?

**LOW DATA CRYPTANALYSIS**

=real world  
attacks

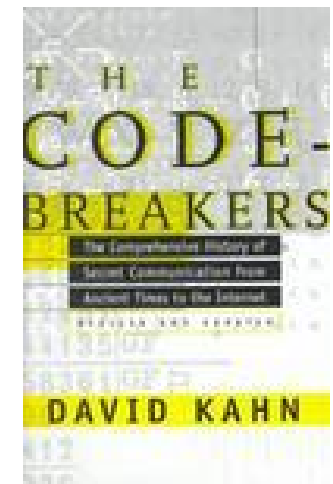


## Marian Rejewski

December 1932:  
reverse engineering of Enigma rotors



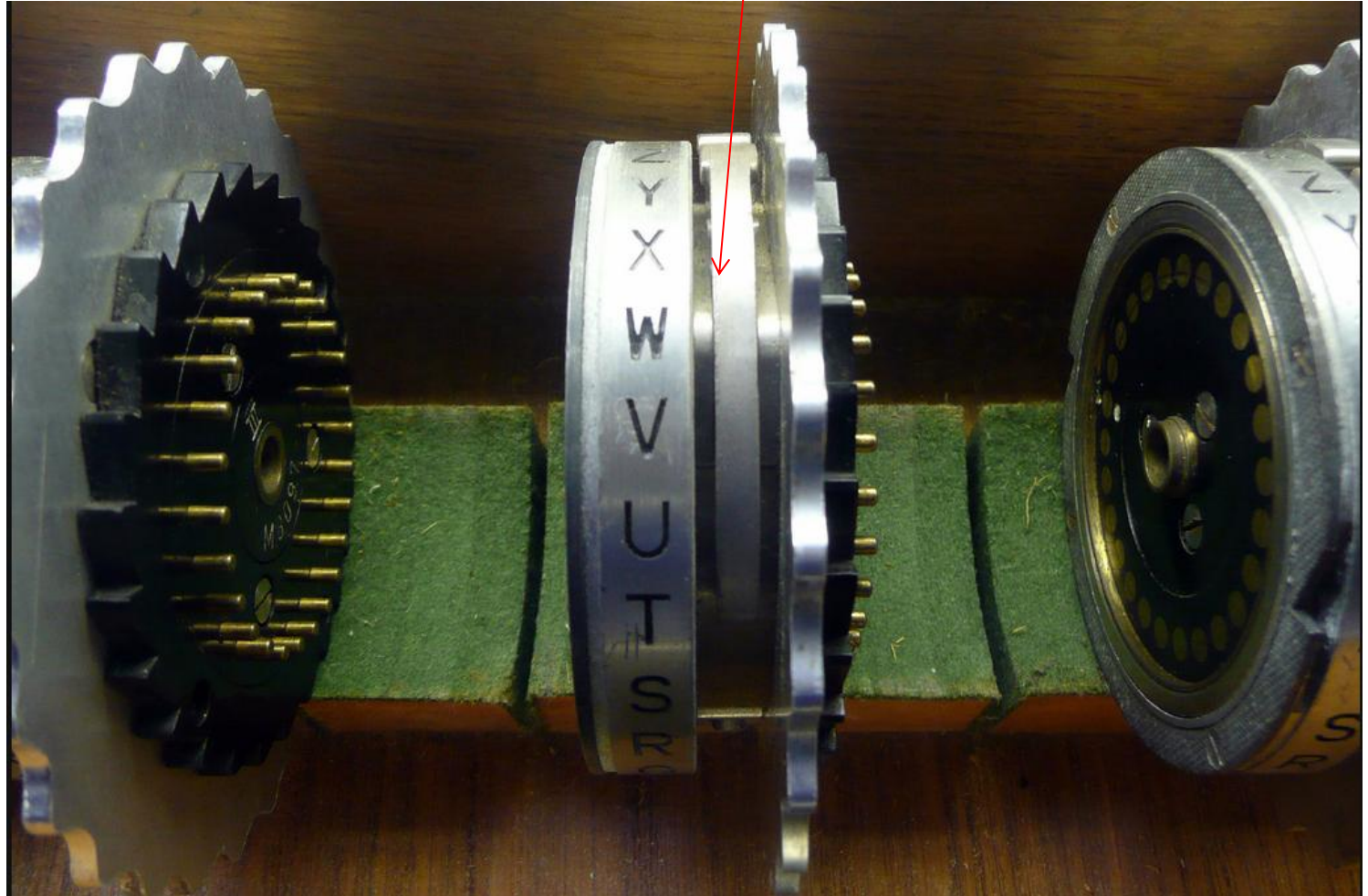
- “the greatest breakthrough in cryptanalysis in a thousand years” [David Kahn]
- cf. John Lawrence, "A Study of Rejewski's Equations", Cryptologia, 29 (3), July 2005, pp. 233–247. + other papers by the same author



## Rotors

26 relative settings

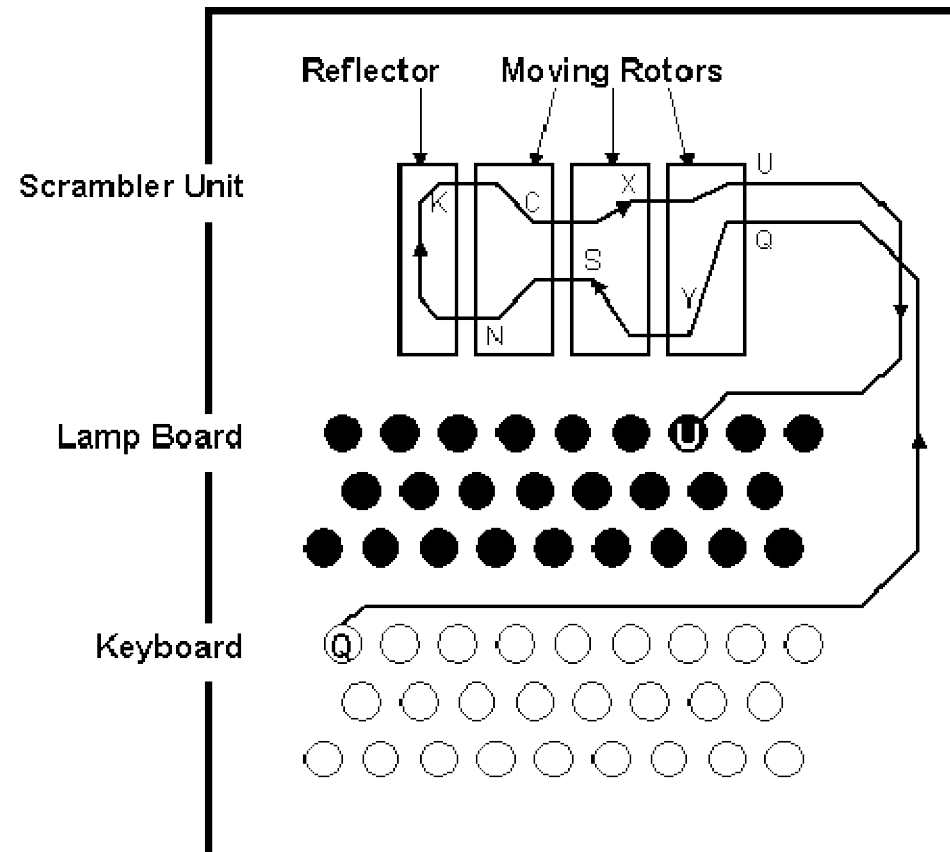
Difficult  
to obtain  
for the  
enemy...



# Commercial Enigma [1920s]

insecure

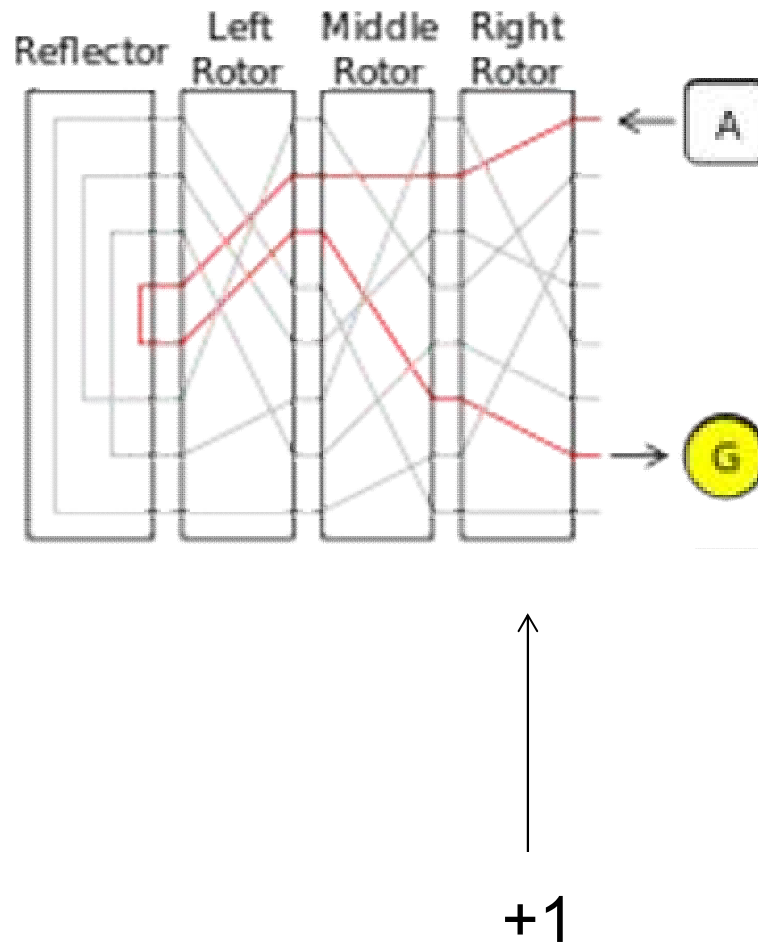
combines several  
permutations on 26  
characters...



## Rotor Stepping

Regular

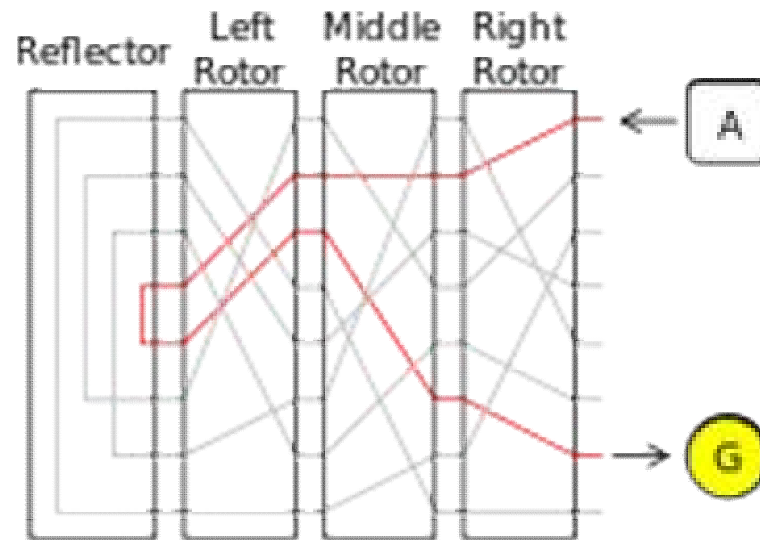
odometer-like



## \*Rotor Stepping

Regular

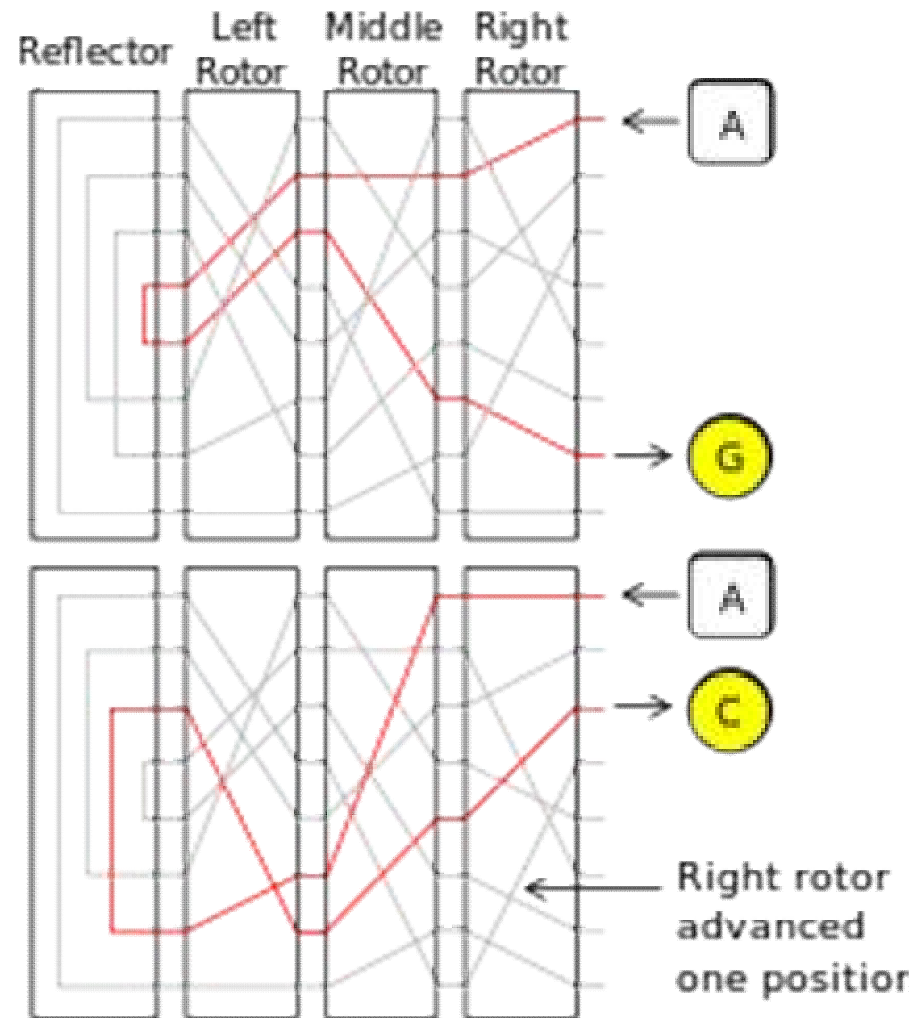
period=?



## \*Rotor Stepping

### Regular

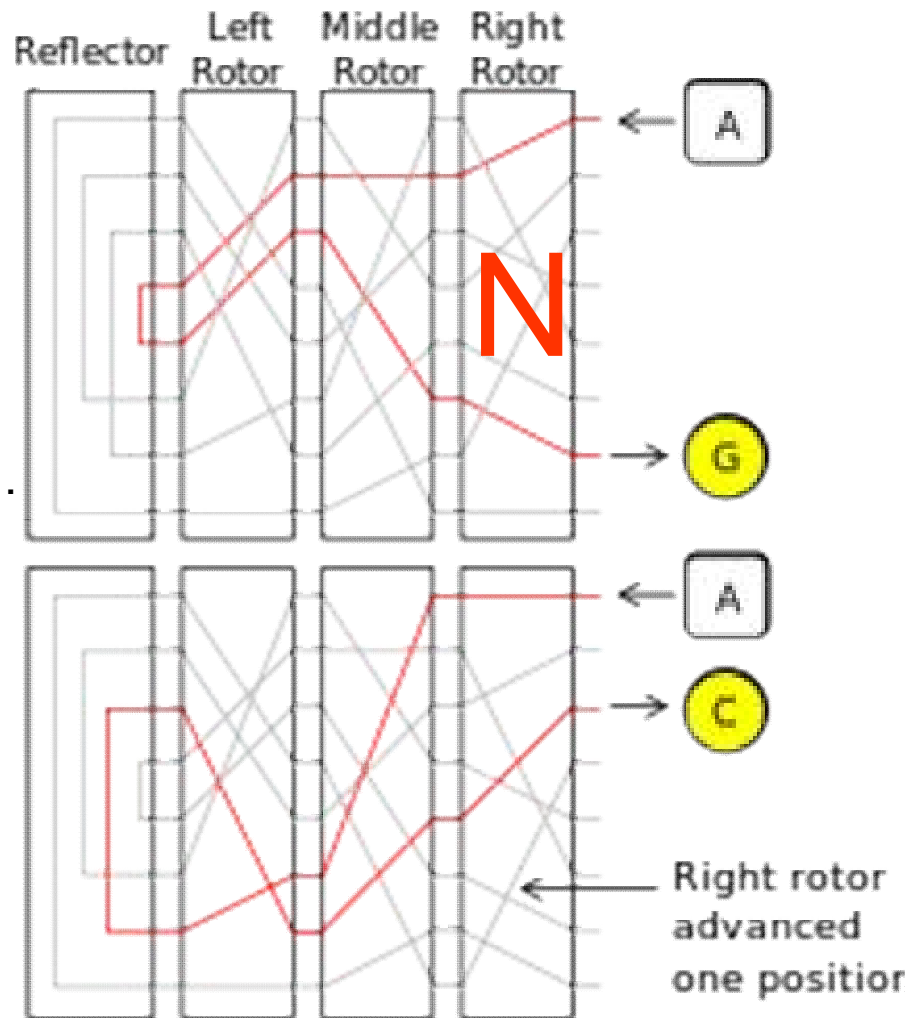
period= $26^3$   
or actually  
 $26^2 * 25$ ?



# Rotor Stepping

- Rotating a rotor:
- **N** becomes  $C^{-1} \circ N \circ C$  (p)
- **C** is a circular shift  $a \mapsto b \dots$

- Proof:





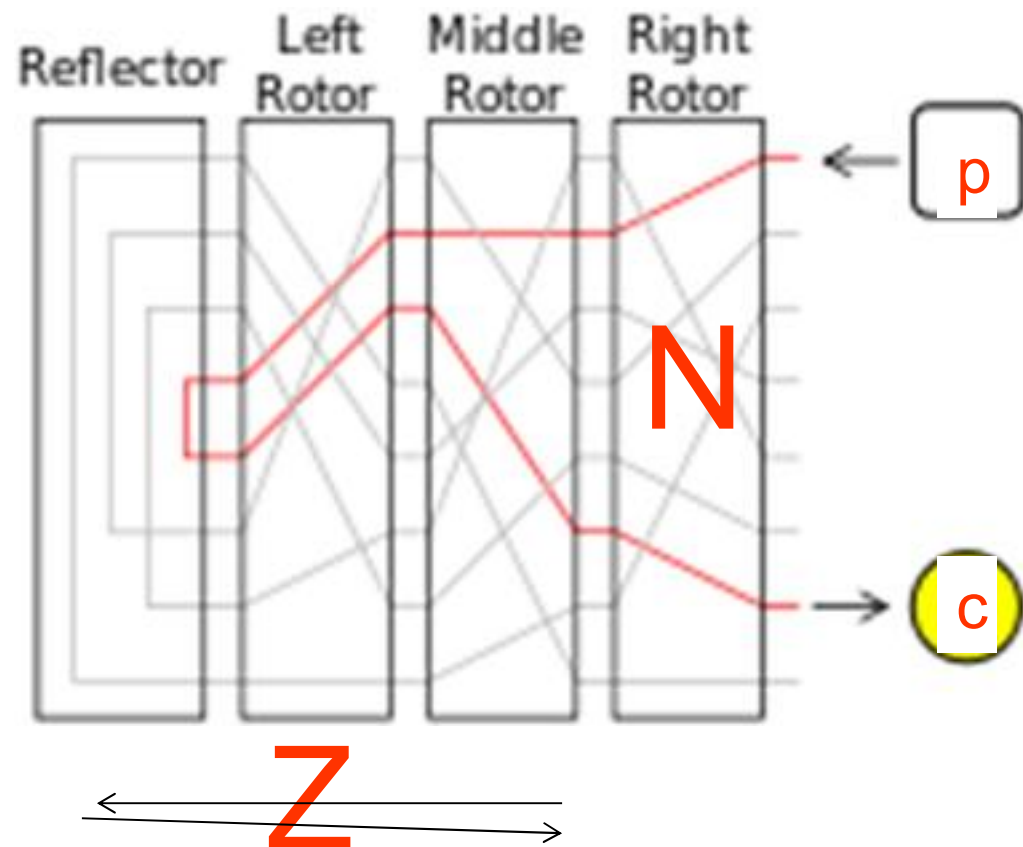
## Bâtons/Rods Attack

Used by French/British/Germans to break  
Swiss/Spanish/Italian/British ciphers in the 1930s...

- assumes only first rotor moving
- rotor wiring known
- guess which rotor is at right
- guess starting position (26)
- guess SHORT crib [plaintext]
- $t=0 \quad c = N^{-1} \circ Z \circ N(p)$
- $t=0 \quad Z \circ N(p) = N(c)$
- $Z$  is an involution

- Rotating a rotors:
- $P$  becomes  $C^{-1} \circ P \circ C(p)$
- $C$  is a circular shift  $a \rightarrow b \dots$

- $t=i \quad Z \circ C^{-i} N C^i(p) = C^{-i} N C^i(c)$



- attack worked until 1939 [cf. Spanish civil war]
- Germans: avoid the attack since 1929/30  
with a **steckerboard**

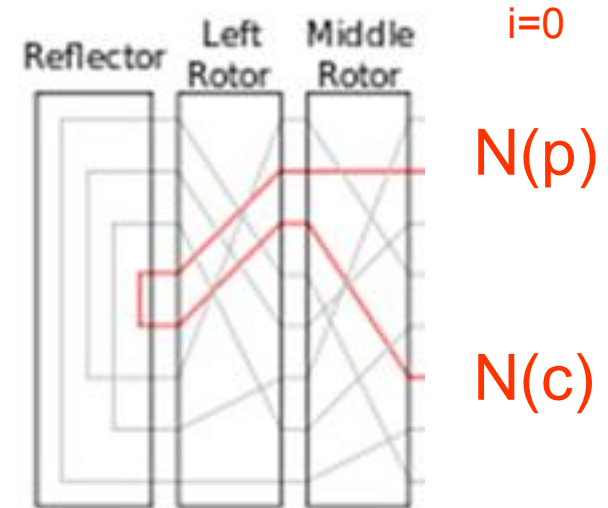


## Bâtons/Rods Attack

Example:

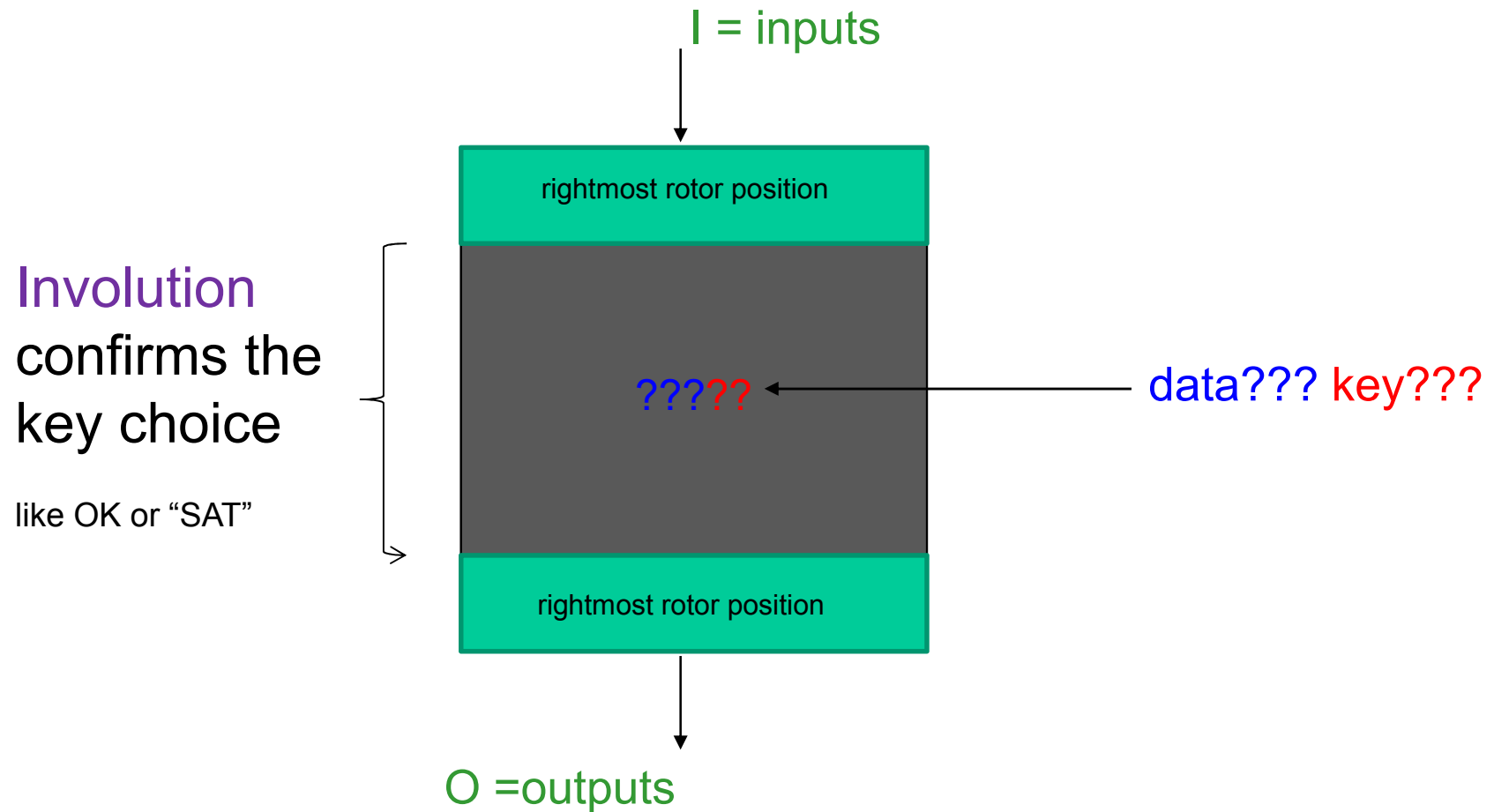
- guess crib = 14 letters plain
- guess which rotor is rightmost
- check ALL 26 starting position pairs for Z obtained:

starting rotor position (indicator setting)	ZN(P) above N(C)
<b>A</b>	U F J R Q N X A W B D R M H A W C Y R G U Q I N N D S Q
<b>B</b>	R B D O N J J P B C I X U E H S H N U A B M M O J K X C
<b>C</b>	N N Z K J T N L C U C D J A Z P I X H W T J P W G L Y G
...	...
<b>Y</b>	J G M G F U H R W C N S E W U Z C Z B J O T A M Q E S A

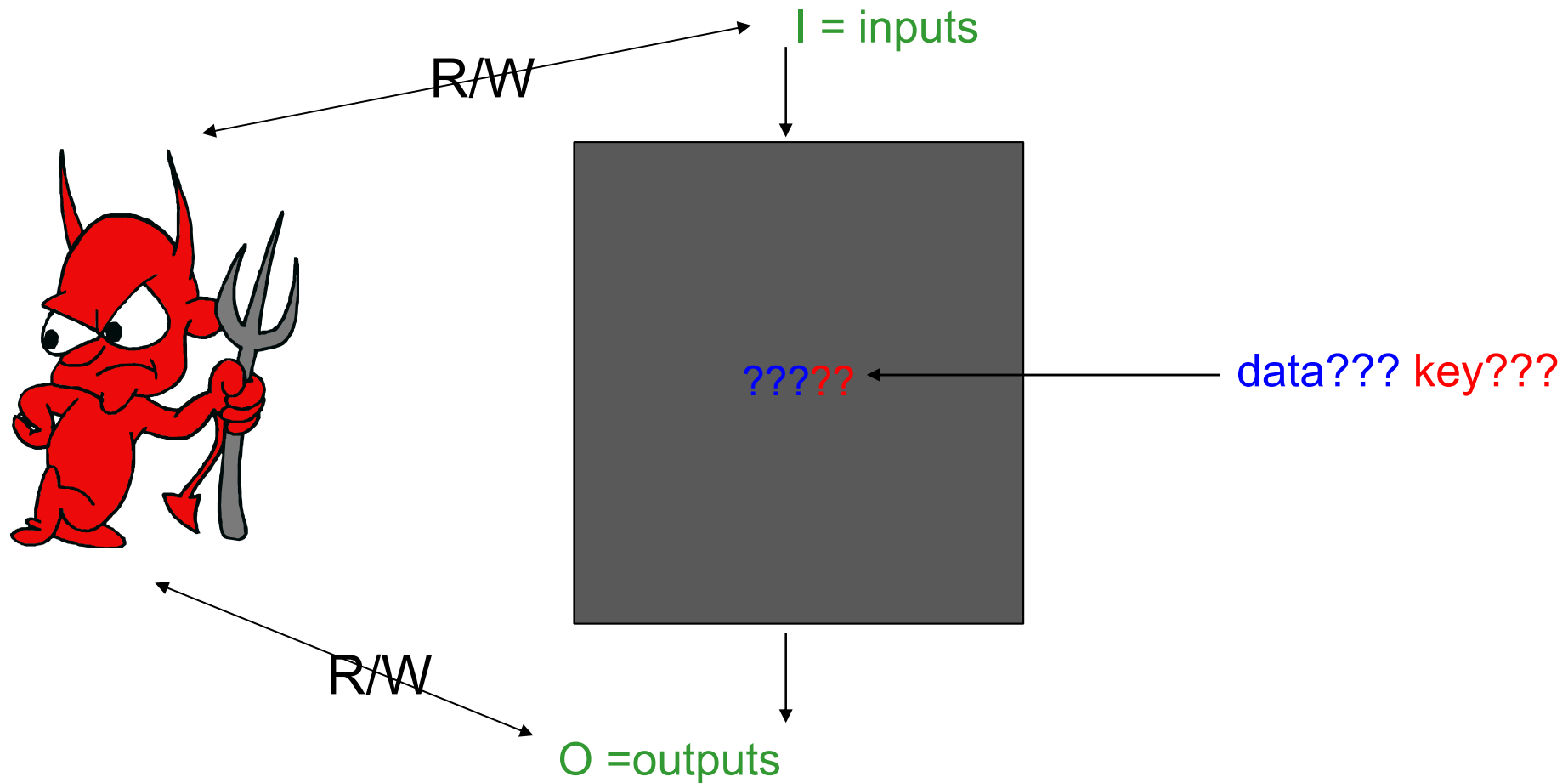


$2^{10}$

## Think Inside the Box



## the Box – General Setting



## \*Modern Method - Contradictions - SAT Solvers

-

There are two main approaches in SAT cryptanalysis or two main algorithms to break a cipher with a SAT solver:

1. **The SAT Method:** Guess  $X$  bits and run a SAT solver which, if the assumption on  $X$  bits is correct takes time  $T$ . Abort all the other computations at time  $T$ . The total time complexity is about  $2^X \cdot T$ .
2. **The UNSAT Method:** Guess  $X$  bits and run a SAT solver which, if the assumption on  $X$  bits is incorrect finds a contradiction in time  $T$  with large probability  $1 - P$  say 99 %.  
With a small probability of  $P > 0$ , we can guess more key bits and either find additional contradictions or find the solution.  
The idea is that if  $P$  is small enough the complexity of these additional steps can be less then the  $2^X \cdot T$  spent in the initial UNSAT step.

## SAT / UNSAT Immunity

See:

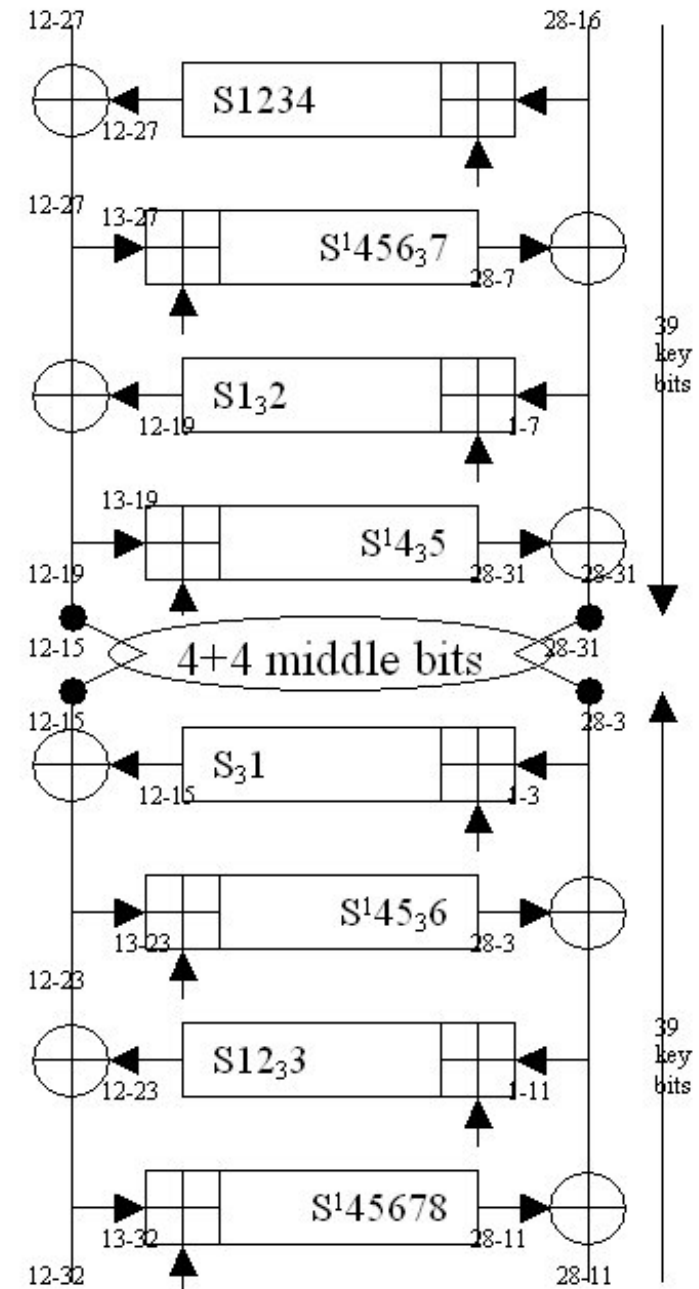
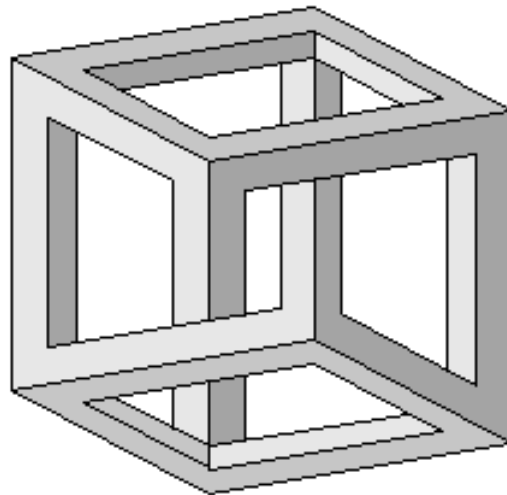
Nicolas Courtois, Jerzy A. Gawinecki, Guangyan Song: [Contradiction Immunity and Guess-Then-Determine Attacks On GOST](#), in Tatrascript 2012.

# UNSAT Immunity – Block Ciphers

Guess **78** bits

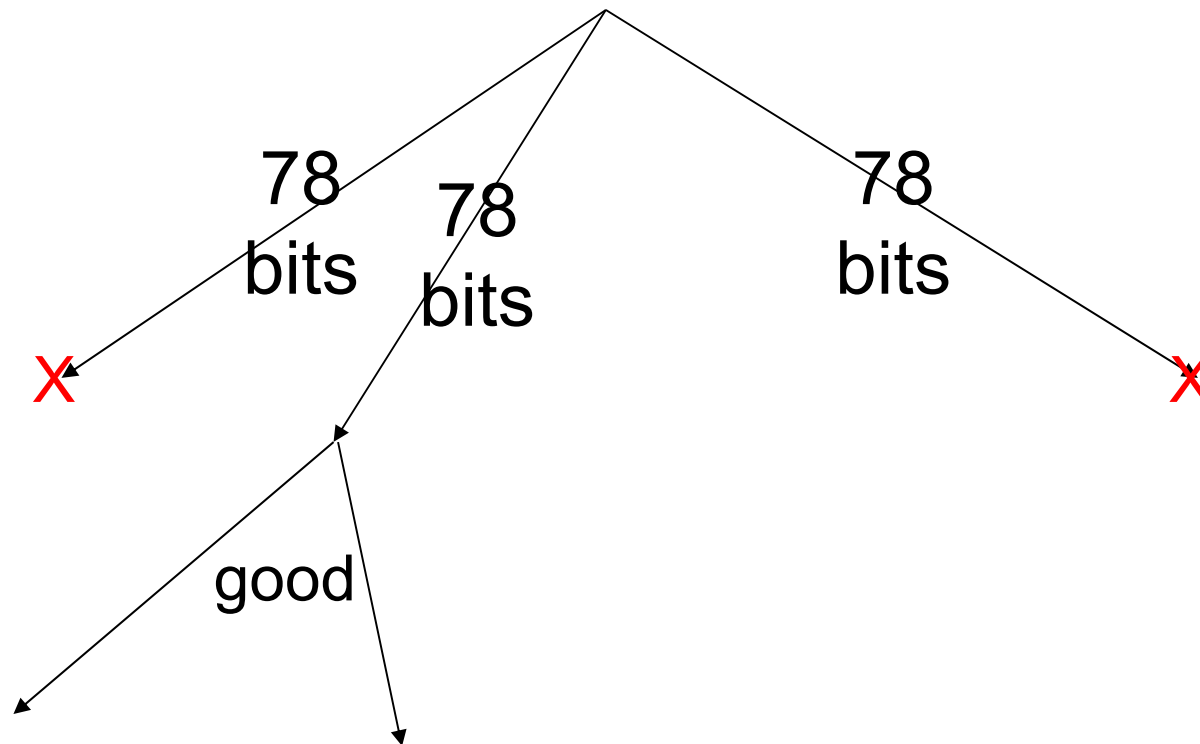
=> Contradiction  
with SAT solver software  
50 % of the time

We say that for 8 rounds of GOST  
the UNSAT Immunity  
is at most **78**  
[Tatracypt 2012]



↑ Guess Then Eliminate ↓

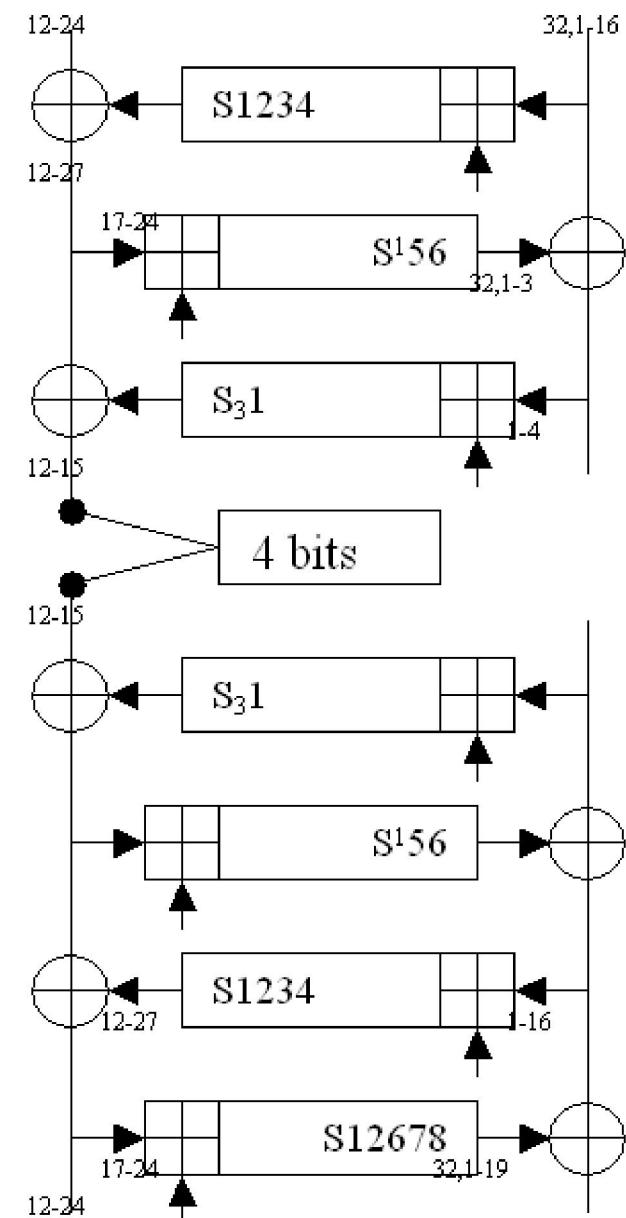
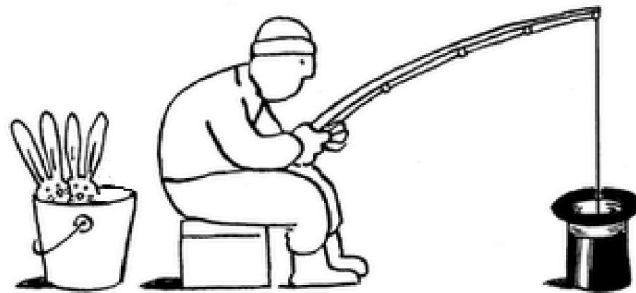
## Depth-First Tree Search.



## SAT Immunity – 4 pairs

Guess these **68** bits.

=> all the other bits?



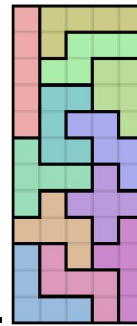


## SAT Immunity – 4 pairs

Guess these **68** bits.

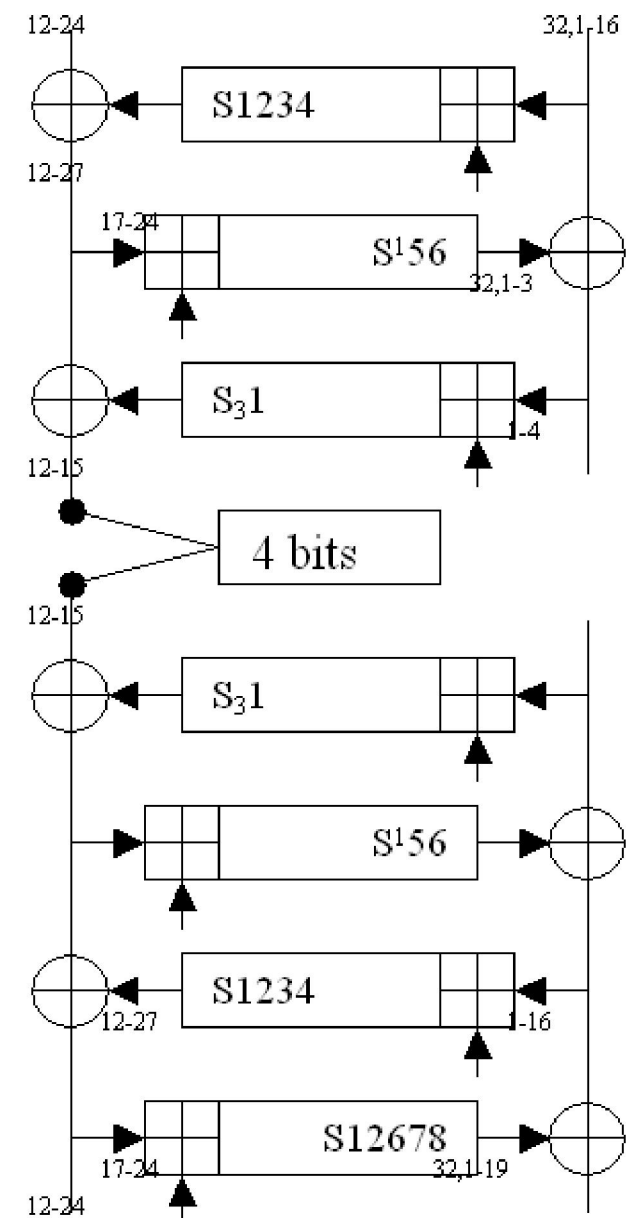
=> all the other bits  
are found in **400 s** on  
one laptop i7 CPU

=> using CryptoMiniSat x64 2.92.



Corollary: Given **4KP** for **8R**  
we determine all the key bits  
in time  **$2^{94}$** .

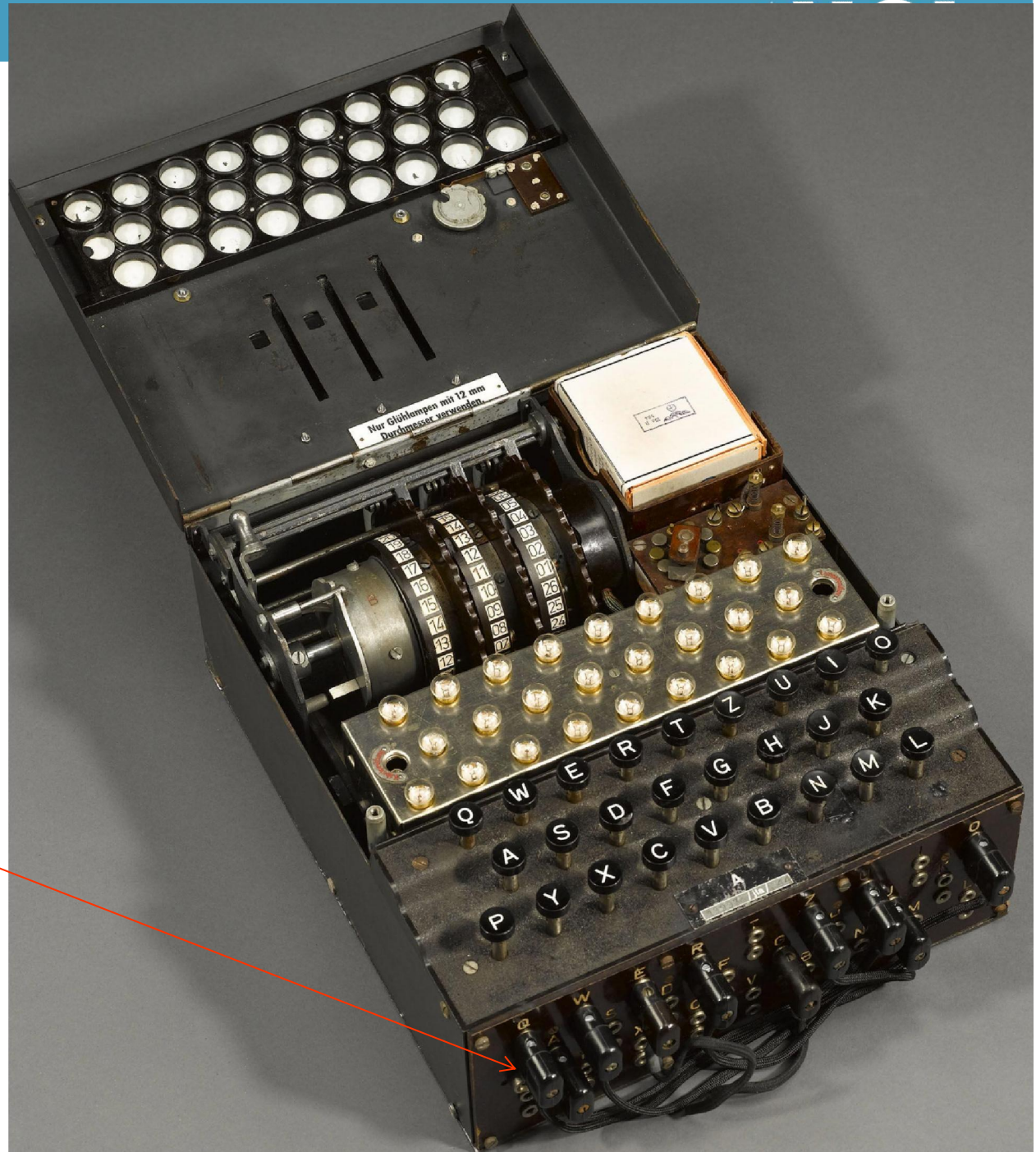
[Courtois Cryptologia vol 37, 2013]



# Military Enigma [1930s]

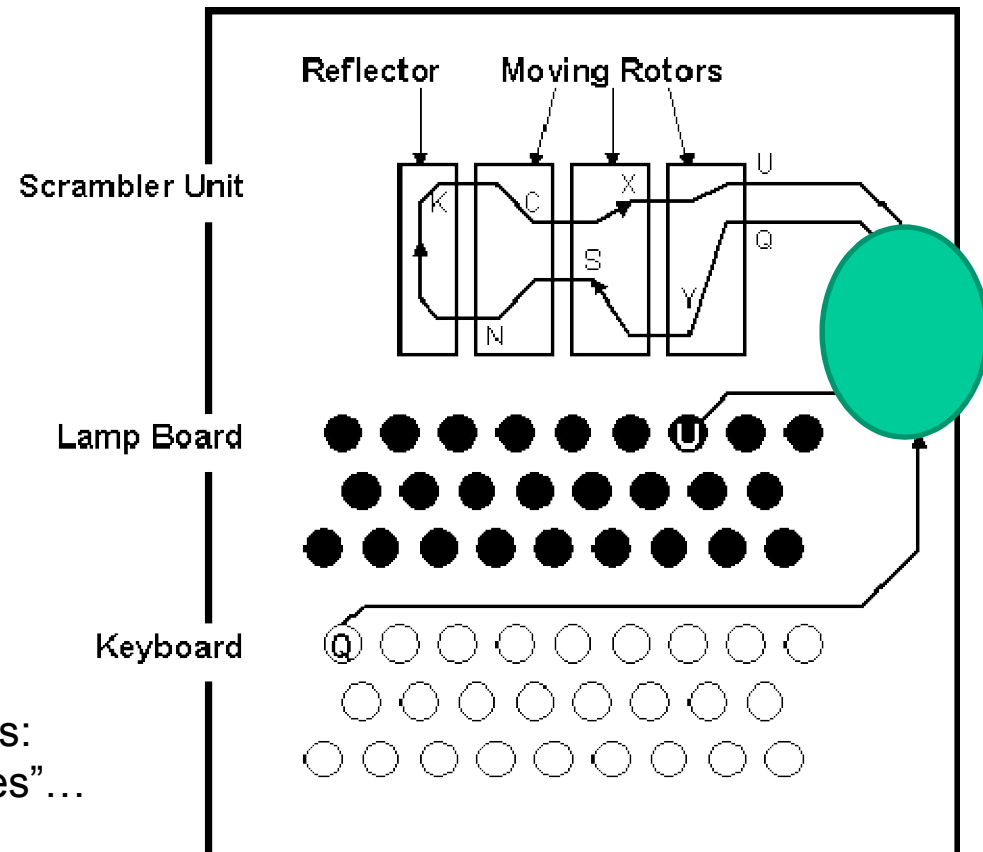
stecker=  
plugboard

Added in 6/1930:



## Stecker

# Huge challenge for code breakers



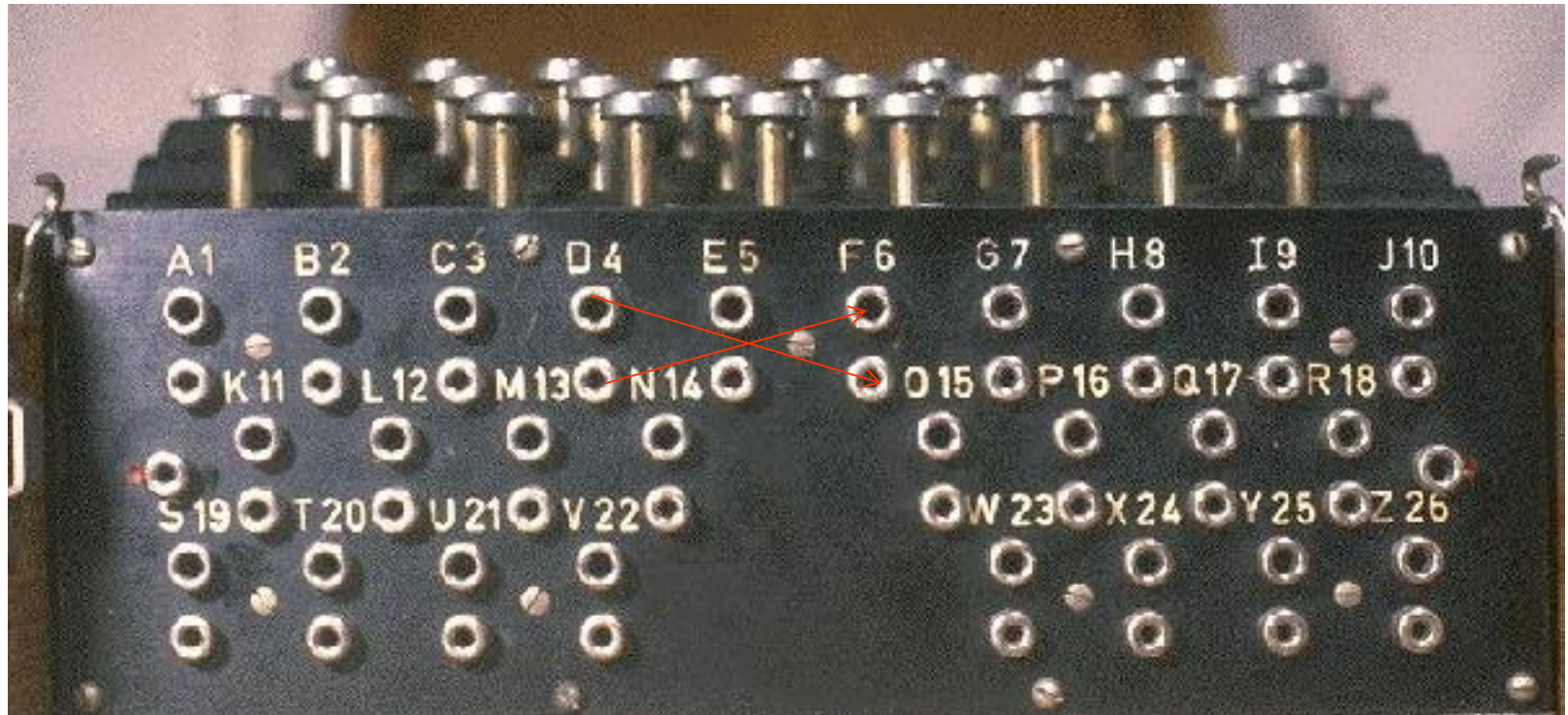
\*common point in all good Enigma attacks:  
eliminate the stecker, “chaining techniques”...

also for Abwehr



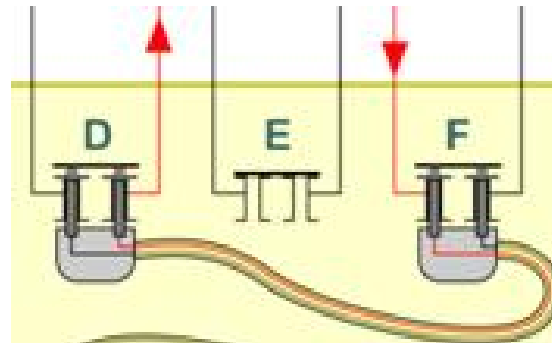
## Stecker

- 6 plugs until Nov 1937
- variable 5-8 plugs...  $\rightarrow$  **S=involution**,
- 10 plugs Nov 1939  $\Rightarrow$  most of the war 6 fixed points

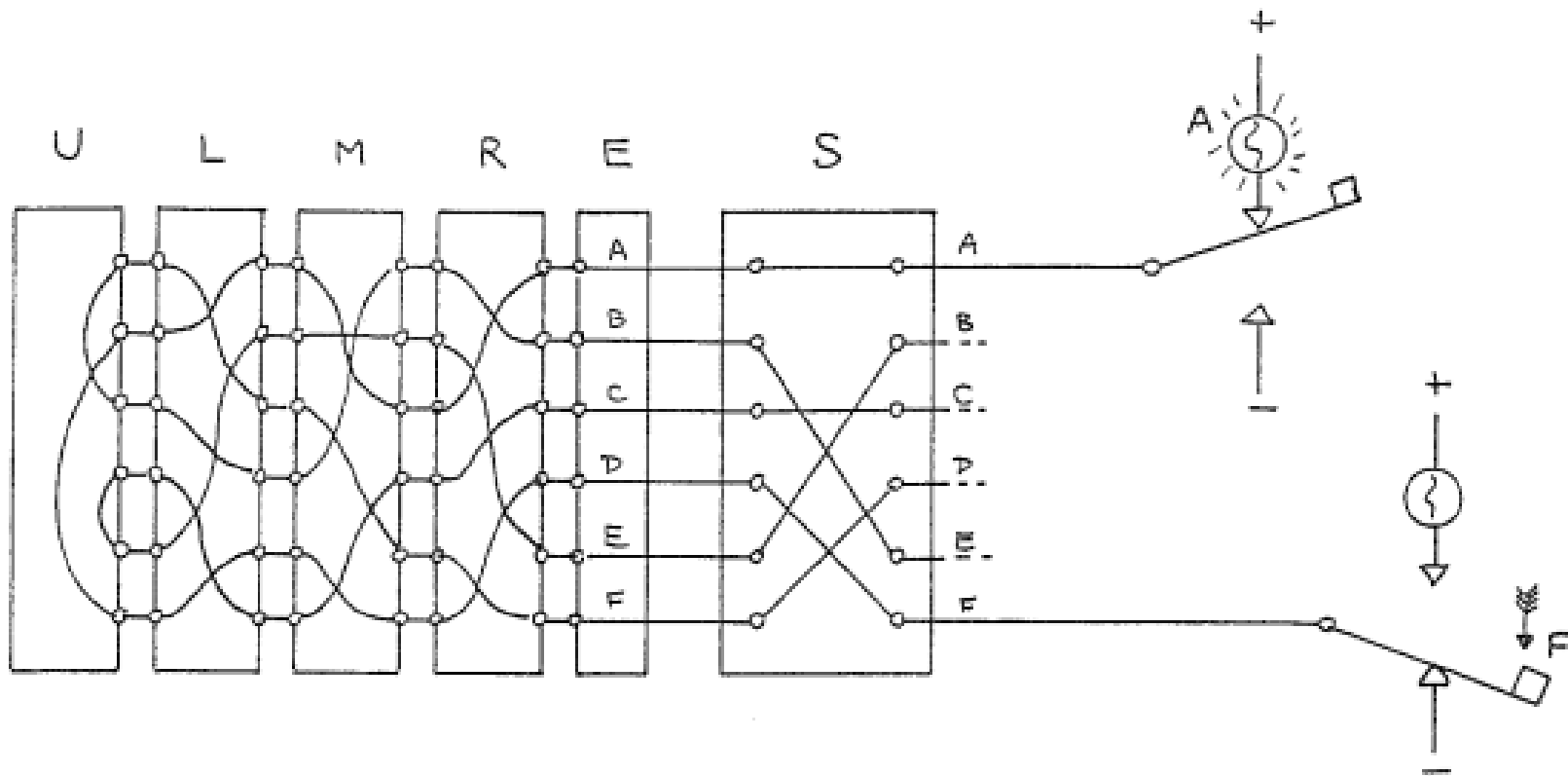


2 holes/letter

no plug  $\Rightarrow$  E  $\rightarrow$  E etc.



# Military Enigma



involution, 13 pairs

picture by D. Davies

## Key Size

About  $2^{380}$  with rotors

Only  $2^{76}$  when rotors are known.

5 main rotors were found by Polish mathematicians before WW2 started.

Same 3 rotors used since 1920s...  
until 1945!!! BIG MISTAKE.

## Part 1

# Permutations

non-commutative

$PoQ \neq QoP$



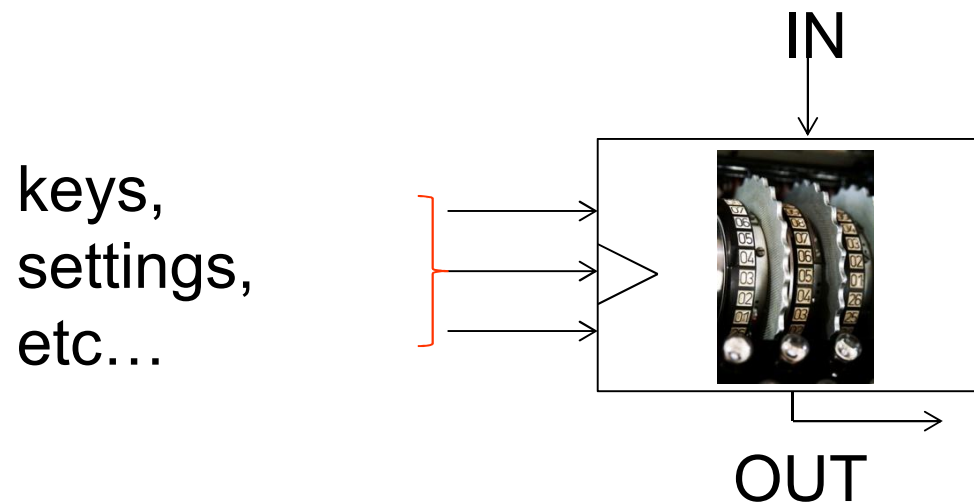
# INI Methods





## Two Main Families of Machines

- Self-reciprocal = involution,
  - e.g. Enigma
- E/D switch:
  - e.g. Fialka, KL7, Typex...

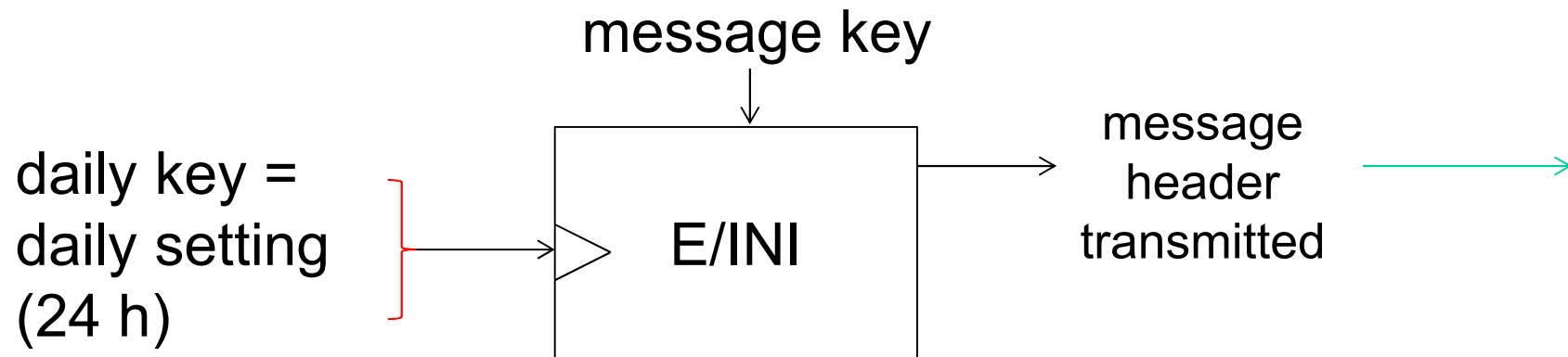


## Message key

Message key=session key=ephemeral key

Should never repeat for two different messages, makes encryption probabilistic

Transmitted to the receiver encrypted (E), must be decrypted (D) by the receiver.



1.1.

# Enigma INI



## History of Enigma Initialization – 3 Periods

Method 1 – 2 Mistakes

6 digits header =  $E(\text{session key})$   
encryption done **twice**,  
**lots** of data with one « daily key »

- 15 Sept 1938

Method 2 - 1 Mistake

9 digits header =  
**twice**  $E(\text{session key})$  with a random  
only 6 chars with the same key!

- 1 May 1940

Method 3 - 0 Mistakes

6 digits header  
no more repeated encryption

## \*Method 1 – 2 Mistakes

before 15 Sept 1938

Except...

- the intelligence agency of the SS
  - and the Nazi Party
- did not make the change until 1 July 1939.

## \*Additional Help: Message Keys NOT Random

(should be 3  
random letters)

~~AAA~~

~~XYZ~~

~~ASD~~

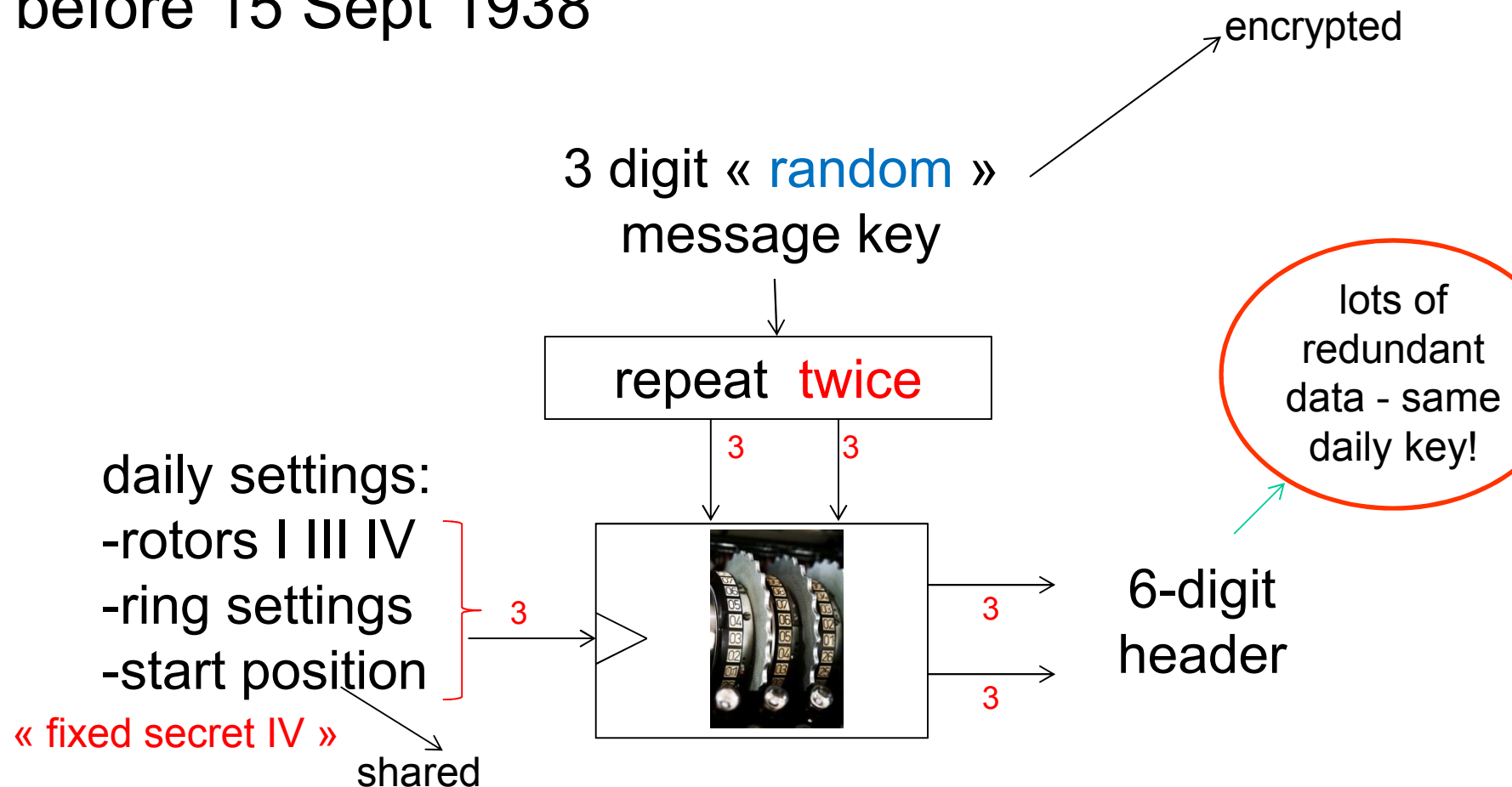
QAY



Operators always found a way to «degrade » their security

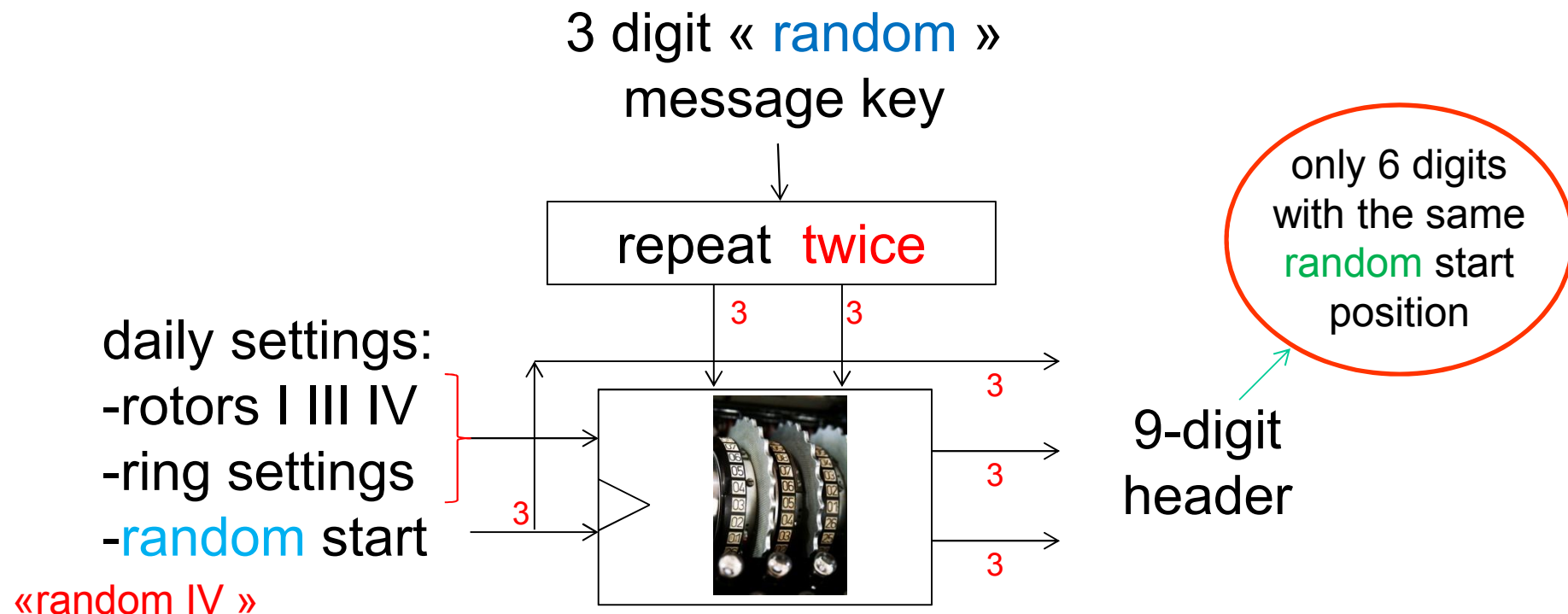
## Method 1 – 2 Mistakes

before 15 Sept 1938



## Method 2 – 1 Mistake

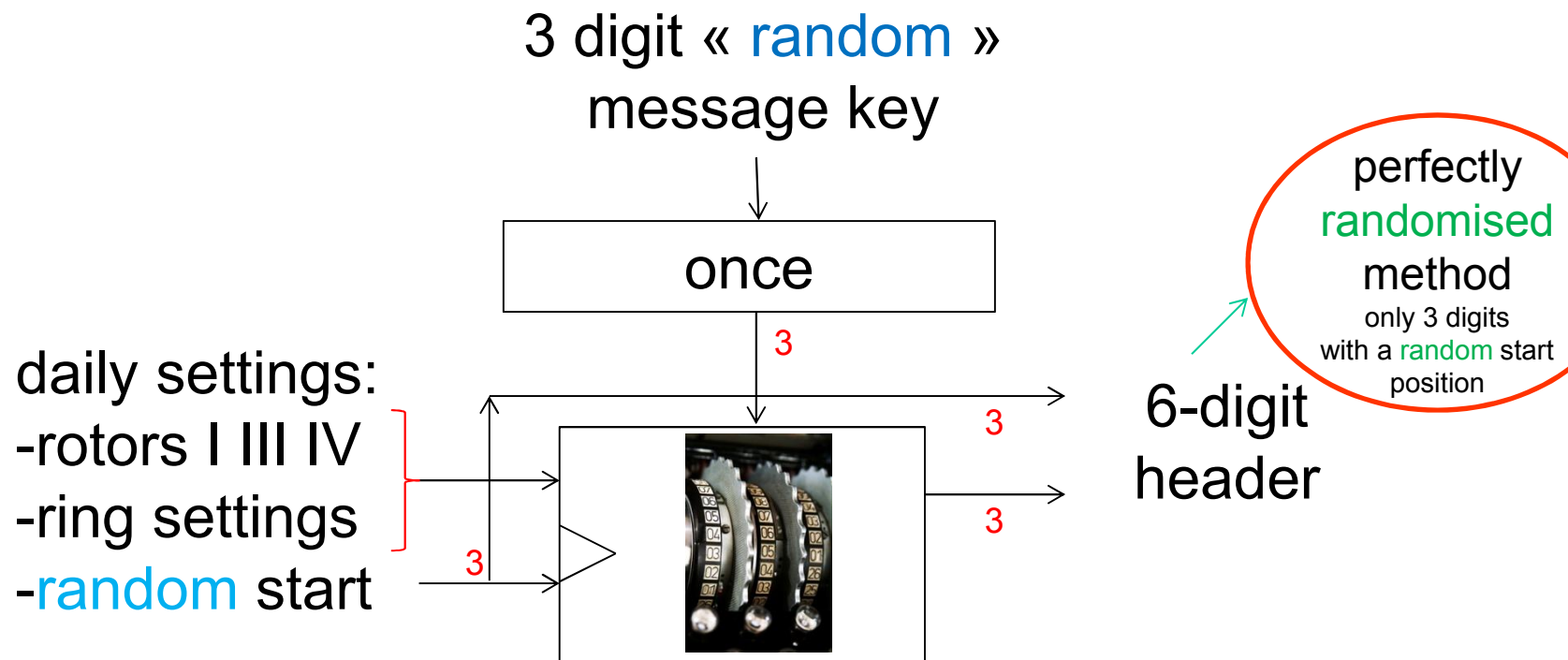
15 Sept 1938 - 1 May 1940 [sometimes also used later, e.g. Norway, Malta 1942]





## Method 3 – 0 Mistakes

after 1 May 1940



## Part 3

# Polish Attacks



Rejewski



Zygalwski

## Three Periods in the History of Enigma

2 Mistakes

Early Polish Methods  
encryption done **twice**,  
**lots** of data with one « daily key »

- 15 Sept 1938

1 Mistake

Zygalski Method  
implemented/used at BP

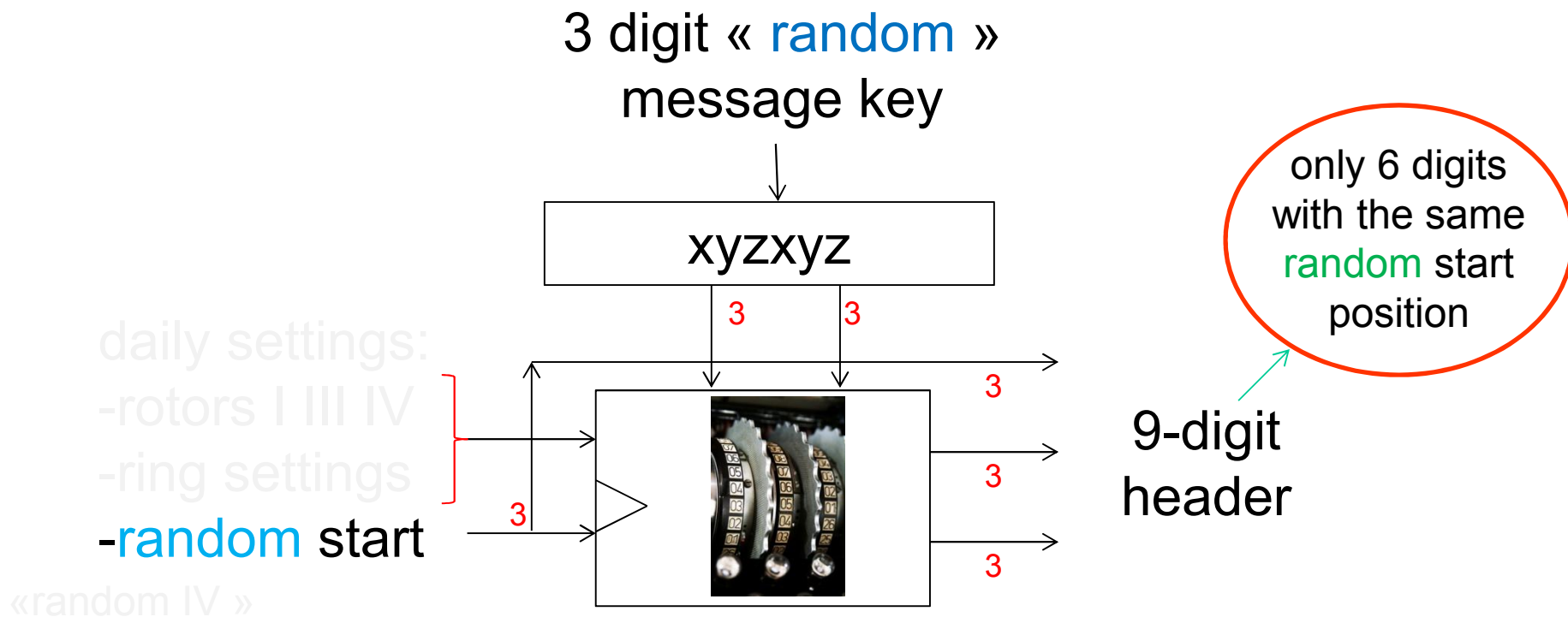
- 1 May 1940

0 Mistakes

[Herivel Attack]  
Turing-Welchman Bombes

## Method 2 – 1 Mistake

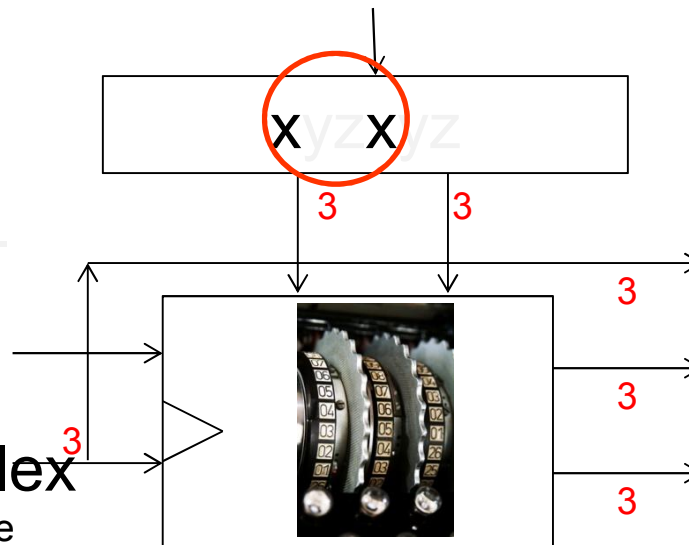
15 Sept 1938 - 1 May 1940 [sometimes also used later, e.g. Norway, Malta 1942]



## focus on repeated indicator:

15 Sept 1938 - 1 May 1940 [sometimes also used later, e.g. Norway, Malta 1942]

3 digit « random »  
message key



our 6 digits  
ciphertext  
header

c d e  
c' d' e'

daily settings: -  
rotors I III IV  
-ring settings

-**random** complex

key for the whole machine  
(arguably not very useful)

# Key Principle in Lots of Enigma Attacks

At steps 1 and 4

same x

$T=1$        $R'_1(x) = c$

$T=4$        $R'_4(x) = c'$

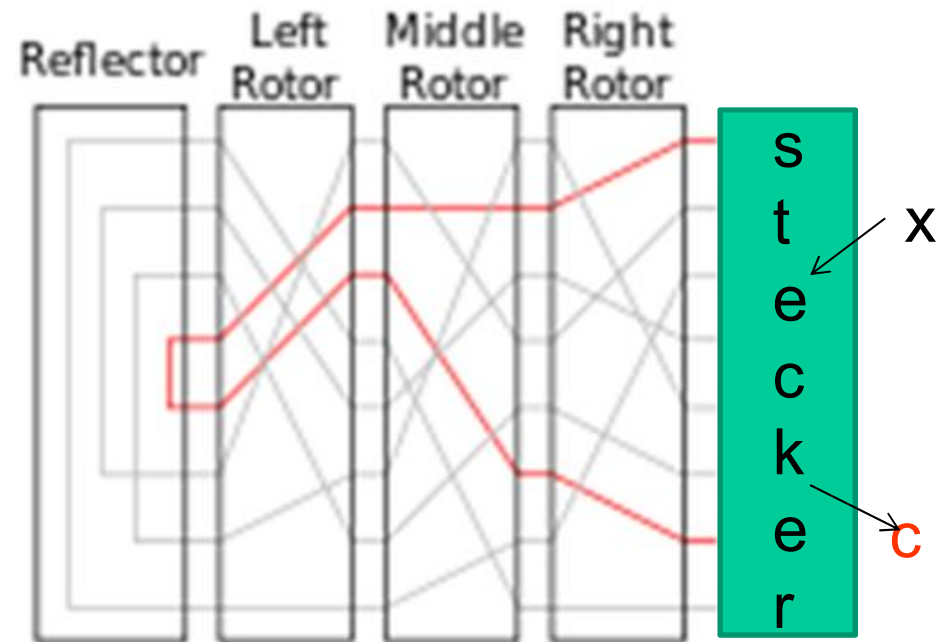
⇒ the attacker can OBTAIN pairs for:

$$R'_4{}^{-1} \circ R'_1$$

$$c \mapsto c'$$

IMPORTANT:  $R'_4$  is an **involution** ⇒

We get to know pairs for a special permutation  $R'_1 \circ R'_4$



$$\leftarrow R'_i \rightarrow$$

# Magic = Permutation Factoring!

At steps 1 and 4

$$\begin{aligned} T=1 & \quad R'_1(x) = c \\ T=4 & \quad R'_4(x) = c' \end{aligned}$$

⇒ the attacker can OBTAIN pairs for:

$$R'_4{}^{-1} \circ R'_1$$

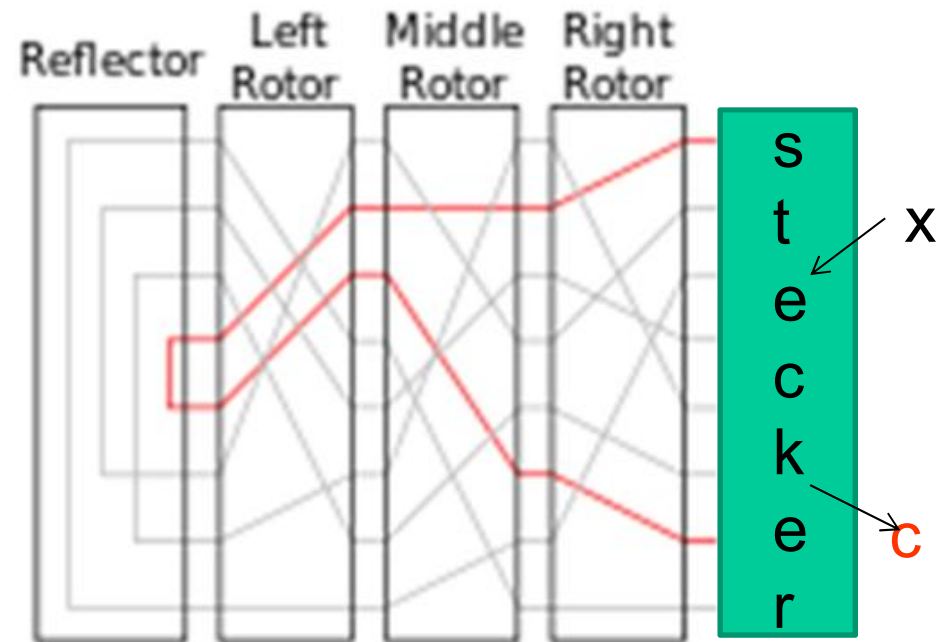
BOTH are involutions

⇒

we CAN recover BOTH by factoring  $R'_1 \circ R'_4$

[due to Rejewski Theorem, they map cycles to identical cycles, cf. slide 138]

Lemma: requires 74 events on average



$R'_i$   
can be recovered!

\*\*\*inside we have

At steps 1 and 4

$$T=1 \quad S^{-1} \circ R_1 \circ S(x) = c$$

$$T=4 \quad S^{-1} \circ R_4 \circ S(x) = c'$$

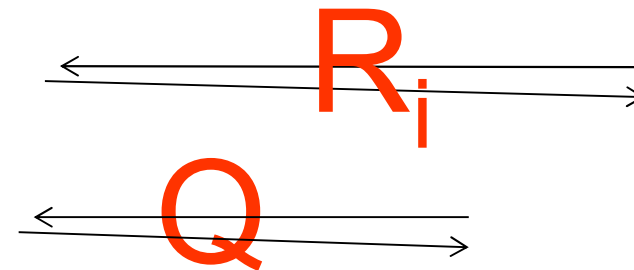
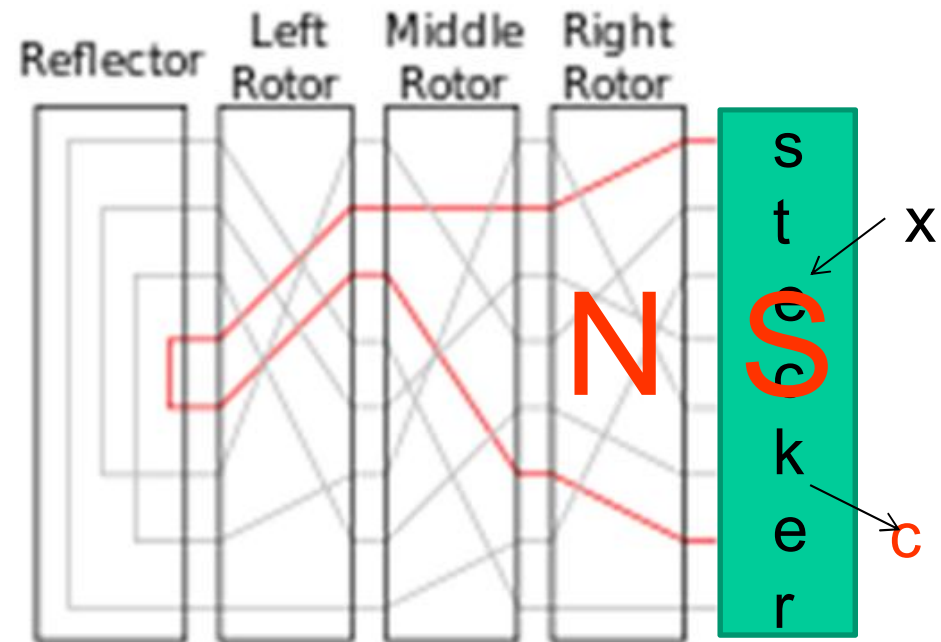
⇒ the attacker can RECOVER  
the combination of these 2 involutions

$$S^{-1} \circ R_4^{-1} \circ \underbrace{S \circ S^{-1}} \circ R_1 \circ S$$

$$c \mapsto c'$$

$$R_4^{-1} \circ R_1$$

$$S(c) \mapsto S(c')$$



the same, high prob  $\approx 0.75$ , no movement



## Do We Have Enough Data $\geq 74$ ?

At steps 1 and 4

$$\begin{array}{ll} T=1 & R'_1(x) = c \\ T=4 & R'_4(x) = c' \end{array}$$

$\Rightarrow$  the attacker can RECOVER both  $R_4^{-1}$  and  $R_1$ .

$\Rightarrow$  before 1938, 2 mistakes,

$R_4^{-1} \circ R_1$  was fixed

in all messages in 1 week or so...

$\Rightarrow$  74 samples  $\Rightarrow$  recover  $R'_i$  by factoring...

$\Rightarrow$  recover all rotors and break keys...

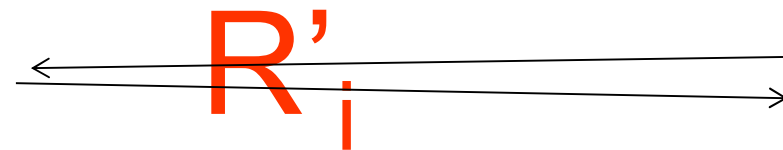
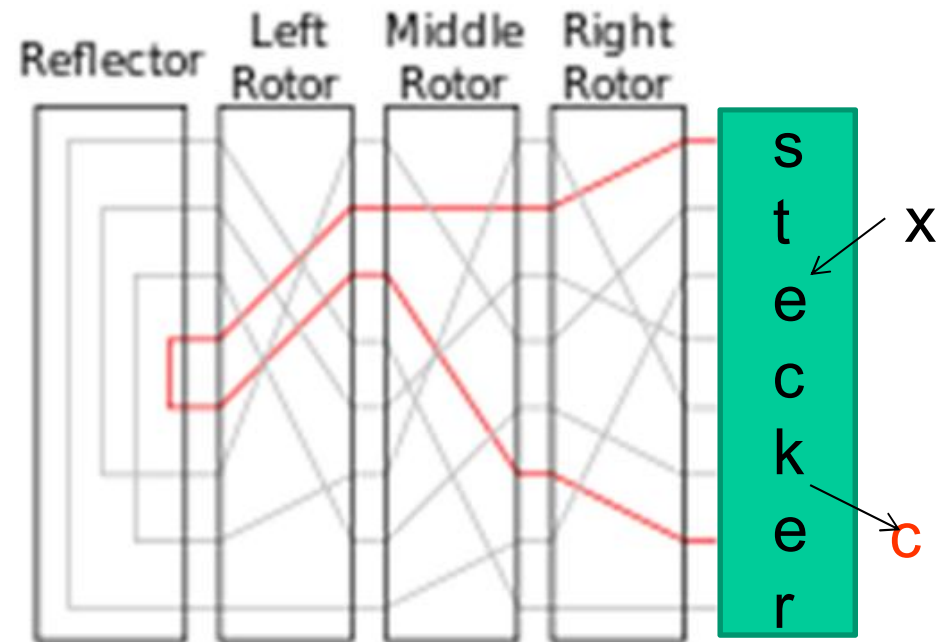
$\Rightarrow$  Sept 1938 - 1 May 1940, 1 mistake,

$R_4^{-1} \circ R_1$  was different

in each message, cf. [Zygalski attack](#)

$\Rightarrow$  could be observed only once

$\Rightarrow$  attacker can see when it has a fixed point  
= so called 'females' [from Polish/English pun same/samica].

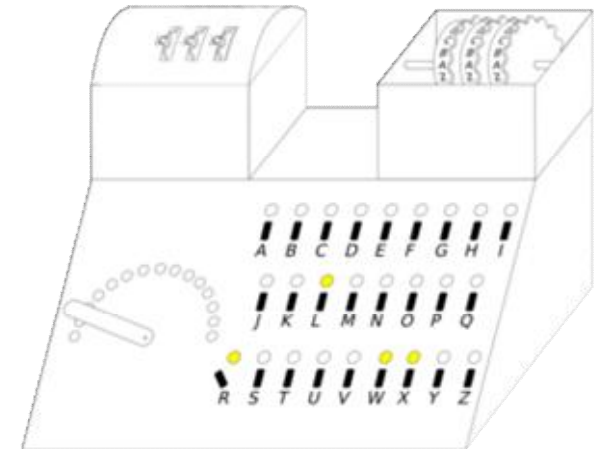


## Early Polish Methods: before Sept 1938

Cyclometer

Making a catalogue  
for breaking Enigma...

Key task for Rejewski...



15 Sept 1938

Panic in Polish Cipher Bureau!

2 new methods were invented:

⇒ Polish Bombs [pbs with stecker 6=>more]

⇒ Zygalski sheets

[BEST, eliminates the stecker totally => massively used during WW2]

## 3b

### Second Generation Enigma Attacks



Rejewski



Zygalski

## Conjugation

“Theorem Which Won World War 2”,

[I. J. Good and Cipher A. Deavours, afterword to: Marian Rejewski, "How Polish Mathematicians Deciphered the Enigma", Annals of the History of Computing, 3 (3), July 1981, 229-232]

$P$  and

$Q^{-1} \circ P \circ Q$

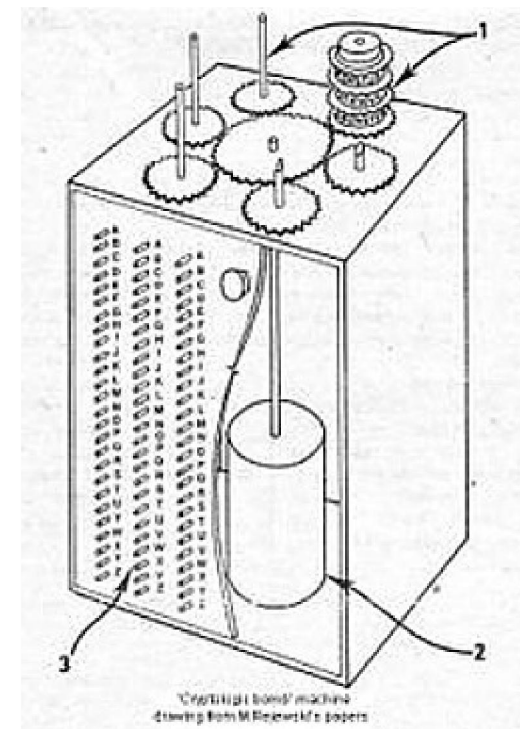
have the same cycle structure

## \*Polish Bombe: worked until 1 May 1940

Short cycles... assumed stecker not active.  
required MANY messages from the same setting...

- Each Message uses the same day-key
- Genius Revelation
  - 1 and 4
  - 2 and 5
  - 3 and 6
  - are encryptions of the same letter with the same **day-key**
  - The **day-key** is always the same

Messages/Characters						
	1	2	3	4	5	6
1	L	O	K	R	G	M
2	M	V	T	X	Z	E
3	L	K	T	M	P	E
4	D	V	Y	P	Z	X



$$R'_4{}^{-1} \circ R'_1$$

$$C \mapsto C'$$

## \*Zygalski Attack: until 1 May 1940

Based on “females”:

Cycles of length 1.

AFKASF  
female

$$T=1 \quad R_1 \circ S(x) = S(A)$$

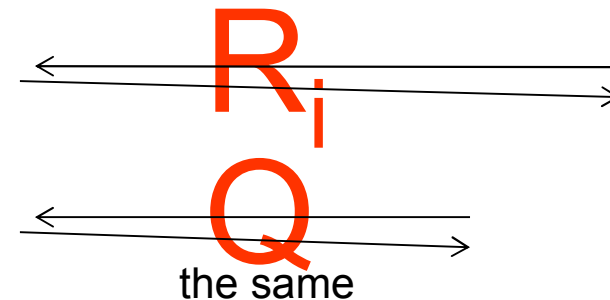
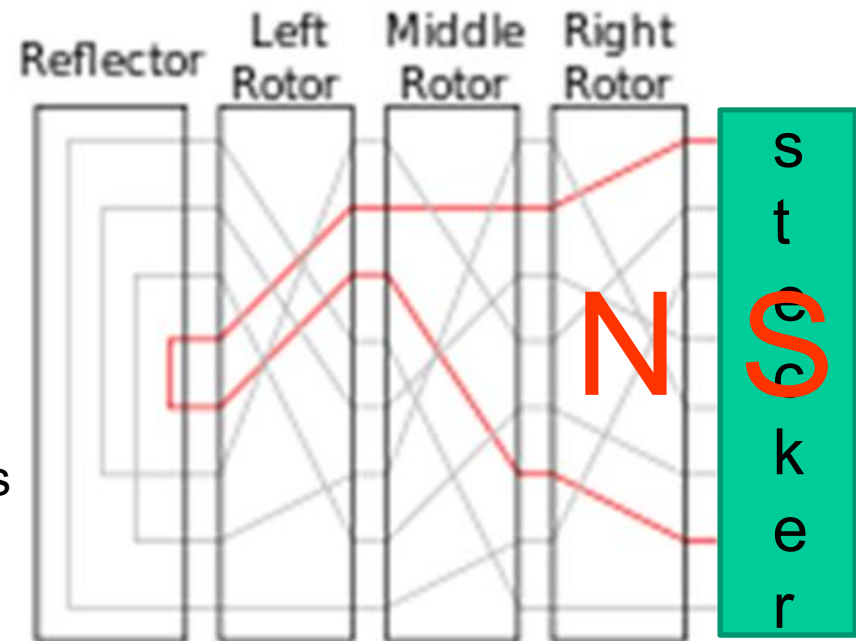
$$T=4 \quad R_4 \circ S(x) = S(A)$$

Same key, same input, same output... +3 steps

$$S^{-1} \circ R_1 \circ R_4 \circ S \quad A \mapsto A$$

$$R_1 = N^{-1} \circ Q \circ N$$

$$R_4 = C^{-4} N^{-1} C^4 \circ Q \circ C^{-4} N C^4$$



## Conjugation

“Theorem Which Won World War 2”,

[I. J. Good and Cipher A. Deavours, afterword to: Marian Rejewski, "How Polish Mathematicians Deciphered the Enigma", Annals of the History of Computing, 3 (3), July 1981, 229-232]

P and

$$Q^{-1} \circ P \circ Q$$

have the same cycle structure

$S^{-1} \circ R_1 \circ R_4 \circ S$  has a fixed point

$\Leftrightarrow$

$R_1 \circ R_4$  has a fixed point

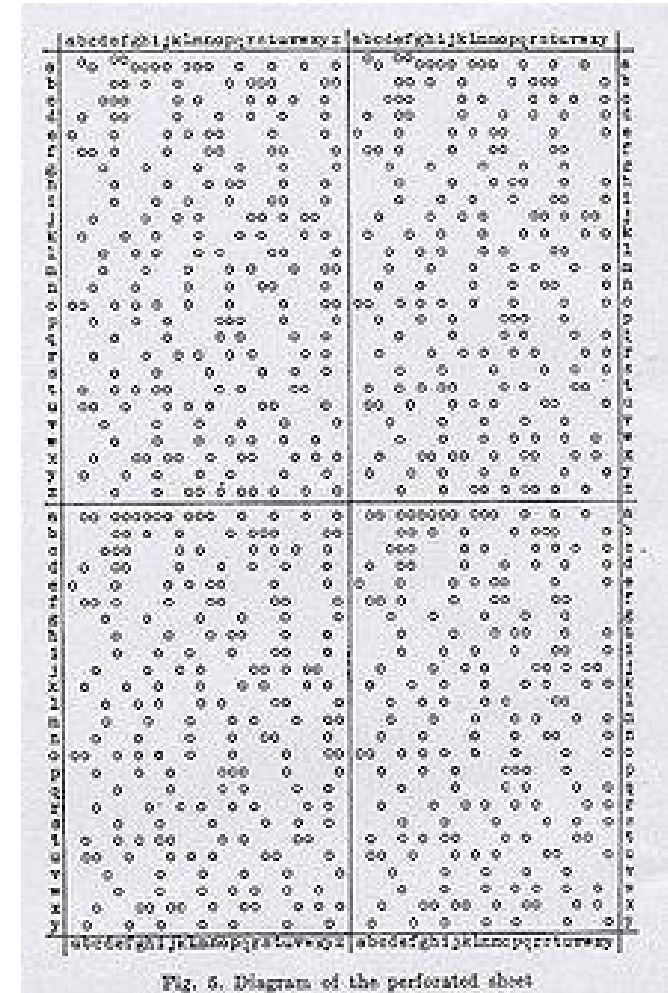
Pty independent on stecker!



## \*Zygalski Attack: until 1 May 1940

fixed points for  $R_1 \circ R_4$

Stacking them allowed to  
determine the key uniquely...

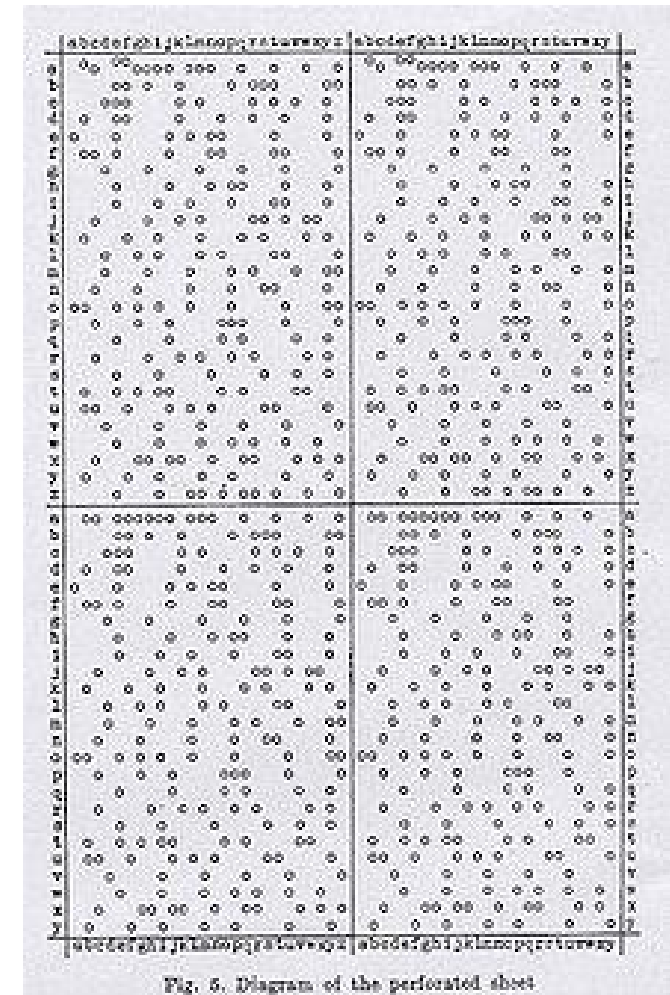


## \*Zygalski Attack: until 1 May 1940

Gave fixed points for ALL  
 $26^3$  settings of 'cleartext IV' [first 3 letters]

Stacking them allowed to  
 determine the key uniquely...

1 hole: for this position of 3 rotors **IFF**  
 $R_1 \circ R_4$  has a fixed point,



## \*Zygalski Attack: until 1 May 1940

Gave fixed points for ALL  
 $26^3$  settings of 'cleartext IV' [first 3 letters]

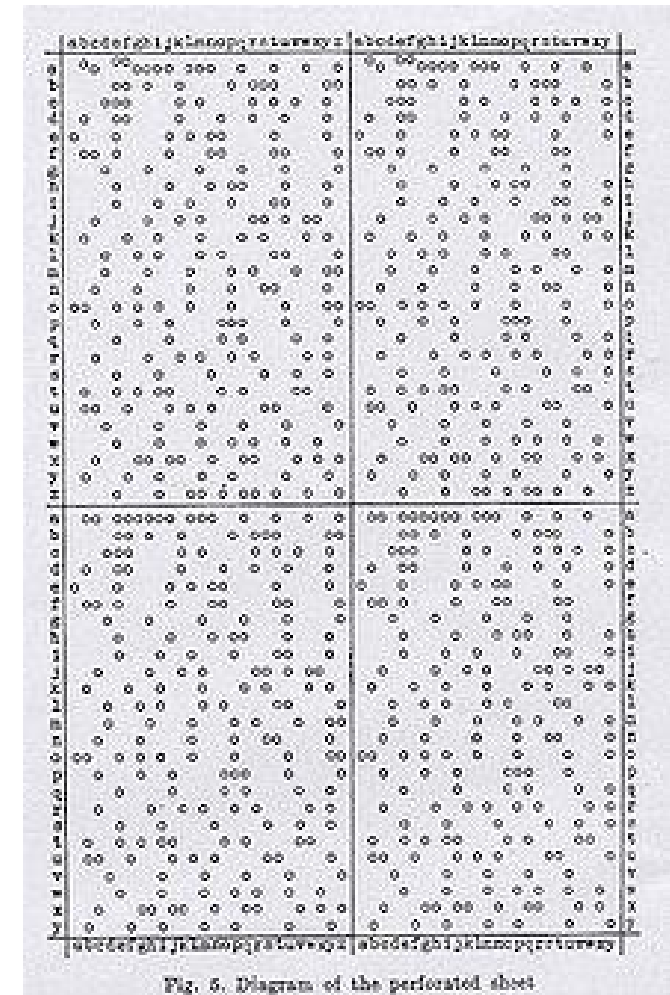
Stacking them allowed to  
 determine the key uniquely...

1 hole: for this position of 3 rotors **IFF**

$R_1 \circ R_4$  has a fixed point,

(a hole 40% of the time)

$P(\text{a 6-letter header has a female}) \approx 1/9$



## \*Zygalski Attack: until 1 May 1940

Gave fixed points for ALL  
 $26^3$  settings of 'cleartext IV' [first 3 letters]

Stacking them allowed to  
 determine the key uniquely...

Each message with a **female** =>

Use 1 sheet at XY offsets defined by  
 letter 2 and 3 in the 9 char header.  
 ZKE **A**FK **A**SF

1 hole: for this position of 3 rotors **IFF**

$R_1 \circ R_4$  has a fixed point,

(a hole 40% of the time)

$P(\text{a 6-letter header has a female}) \approx 1/9$

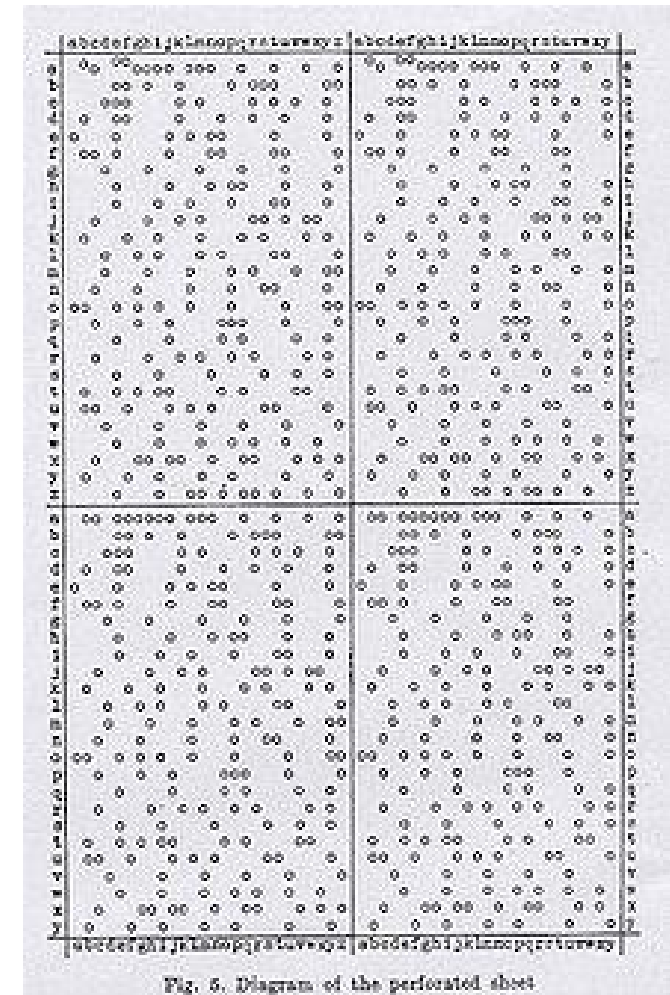


Fig. 5. Diagram of the perforated sheet

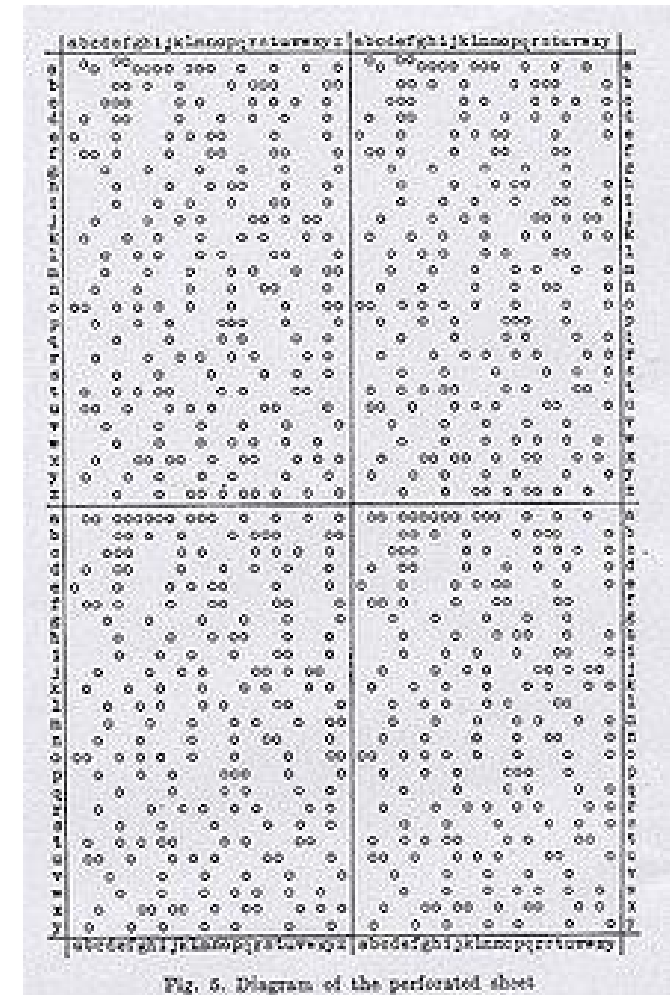
## \*Zygalski Attack: until 1 May 1940

Gave fixed points for ALL  
 $26^3$  settings of 'cleartext IV' [first 3 letters]

Stacking them allowed to  
 determine the key uniquely...

Stacking up to 11 sheets did the job  
 [only 1 common hole remains]

1 hole: for this position of 3 rotors IFF  
 $R_1 \circ R_4$  has a fixed point,  
 (a hole 40% of the time)  
 $P(\text{a 6-letter header has a female}) \approx 1/9$



## Part 4

# British BP Enigma Attacks

=3<sup>rd</sup> generation=



## Turing Attack – Preliminary Step

1. Rejecting possibilities

Encrypted text

p	e	g	m	u	o	x	y	q	p	w	t	j	a	b	x	l	p	v
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

2. Some are still possible.

w	e	t	t	e	r	v	o	r	h	e	r	s	a	g	e
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

The longer the crib, the easier to reject!!!!

WW2 messages had 100-500 characters only, rare exceptions

## Turing Attack = Crib Loops [Short Cycles]

3. We obtain pairs,  
KPA + rotors move

	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4
PLAIN TEXT:	O	B	E	R	K	O	M	M	A	N	D	O	D	E	R	W	E	H	R	M	A	C	H	T
CIPHERTEXT:	Z	M	G	E	R	F	E	W	M	L	K	M	T	A	W	X	T	S	W	V	U	I	N	Z

4. Find loops

A=>M=>E=>A

9    7    14

Main idea: cycles CAN eliminate  
most stecker connections

(1 guess may be needed)

150 million million =  $2^{47}$



# Eliminating the Stecker [Turing Method]

- $S$  is the stecker (involution)
- $T=i \quad R_i \circ S(p) = S(c)$

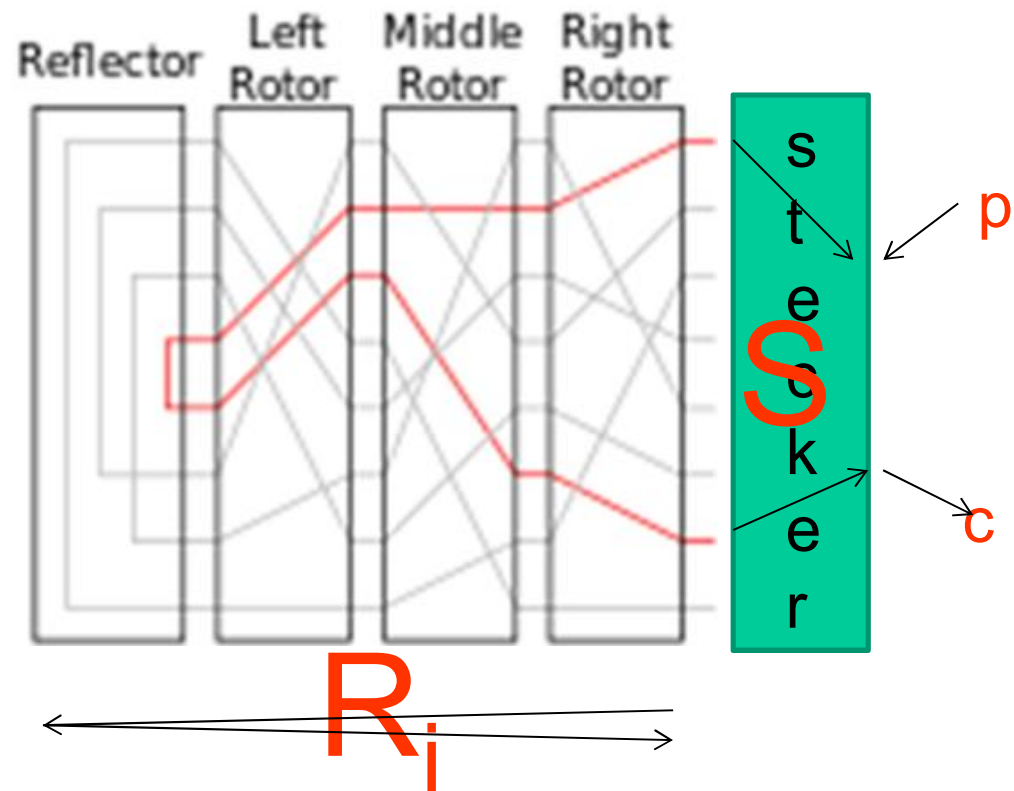
$A \Rightarrow M \Rightarrow E \Rightarrow A$   
 9    7    14

- $T=9 \quad R_9 \circ S(A) = S(M)$
- $T=7 \quad R_7 \circ S(M) = S(E)$
- $T=14 \quad R_{14} \circ S(E) = S(A)$

$$R_7 \circ R_9 \circ R_{14} \circ S(E) = S(E)$$

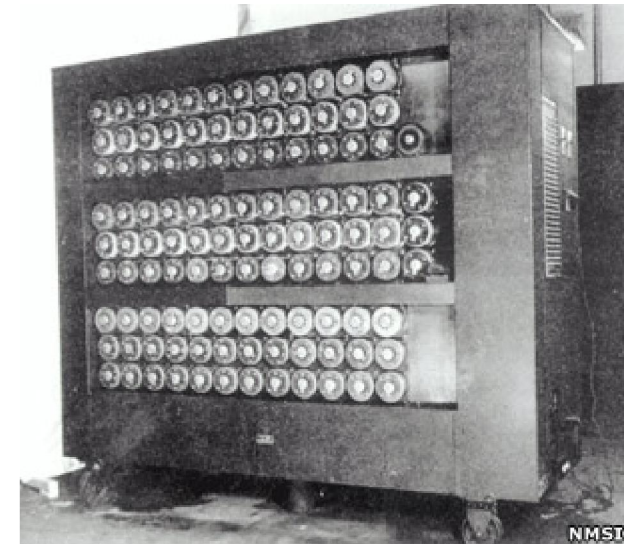
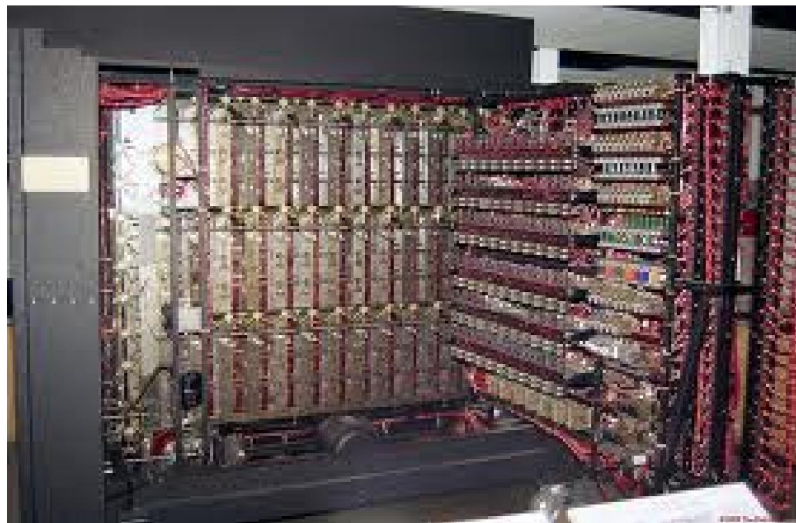
Needed 20 letter cribs,  
 4 loops, preferably  
 sharing letter E

Bombes implemented this:  
 serial connection of several simulated Enigmas



## Turing-Welchman Bombe

240 were built!



guess/test all 26  
possible S(E)

IMPORTANT:  
Bombes ASSUMED MIDDLE ROTOR not moving  
(large proba for shorter cribs, if fails, repeat...)

$$R_7 \circ R_9 \circ R_{14} \circ S(E) = S(E)$$

serial connection of several unsteckered Enigmas

Accept – Reject a guess for  $S(E)$

Closed loop  
connection

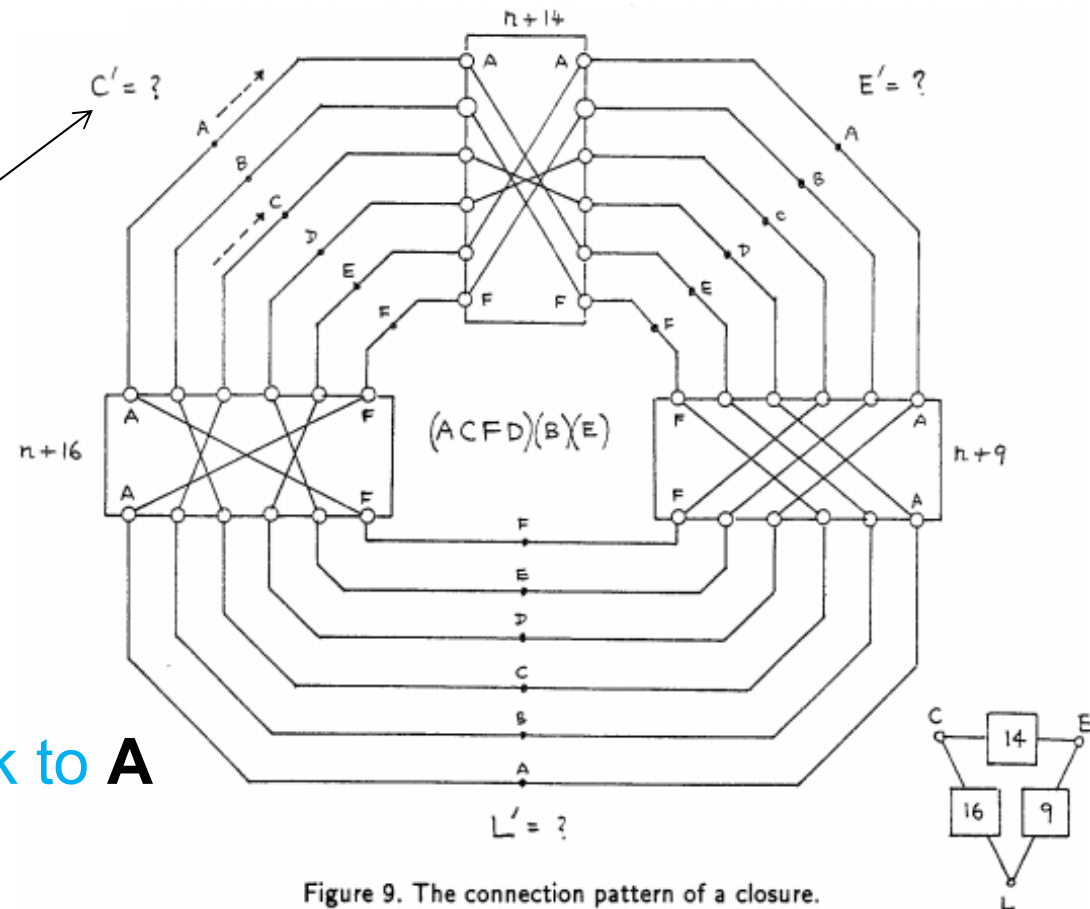
Correct  $S(E)$  is a fixed point!

**A** correct

=>The current comes back to **A**

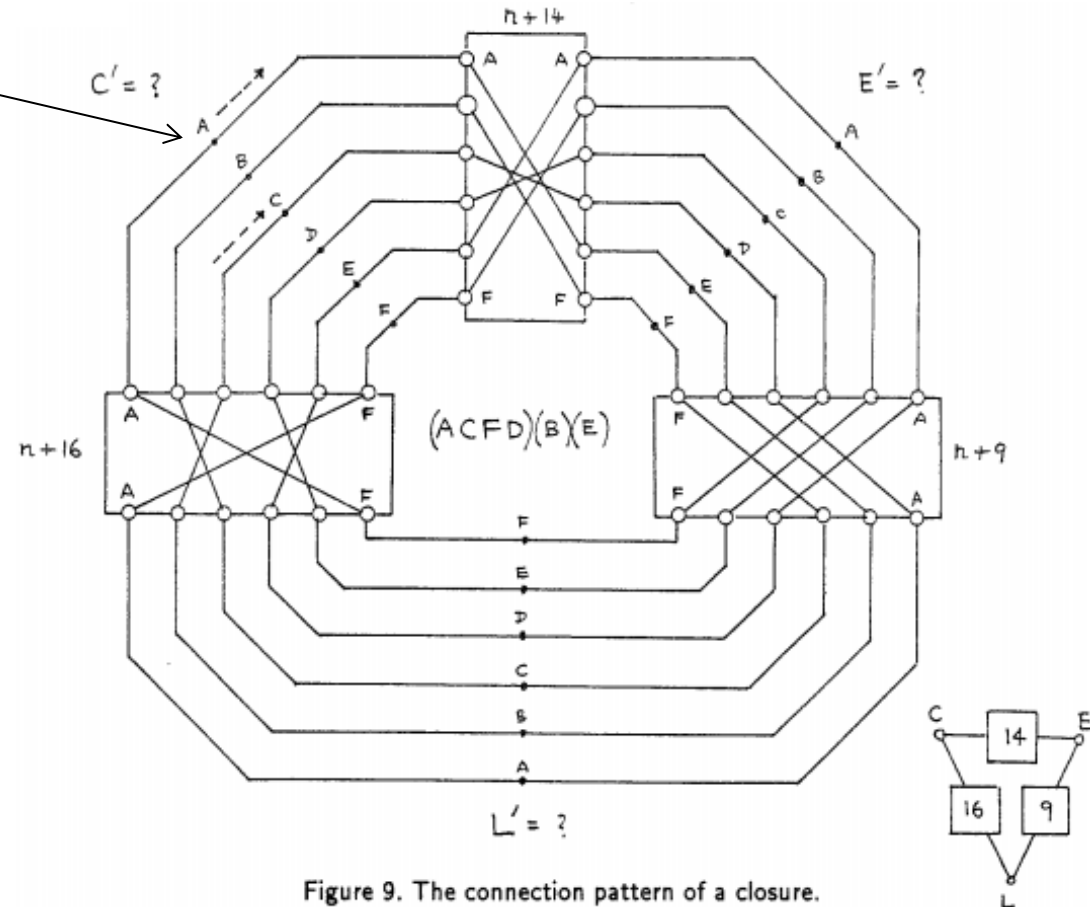
$$R_7 \circ R_9 \circ R_{14} \circ S(E) = S(E)$$

implemented by the bombe



Miracle: 95% of values rejected in 1 step – Simultaneous Scanning

- If A incorrect, the current makes several loops and all active values are incorrect!
- Most values rejected in 1 step
- What remains: 123 fixed points, correct guesses! (machine stops).



$$R_7 \circ R_9 \circ R_{14} \circ S(E) = S(E)$$

In a closed loop

## Most Frequent Case:

- A incorrect,  
AND all the settings  
were incorrect

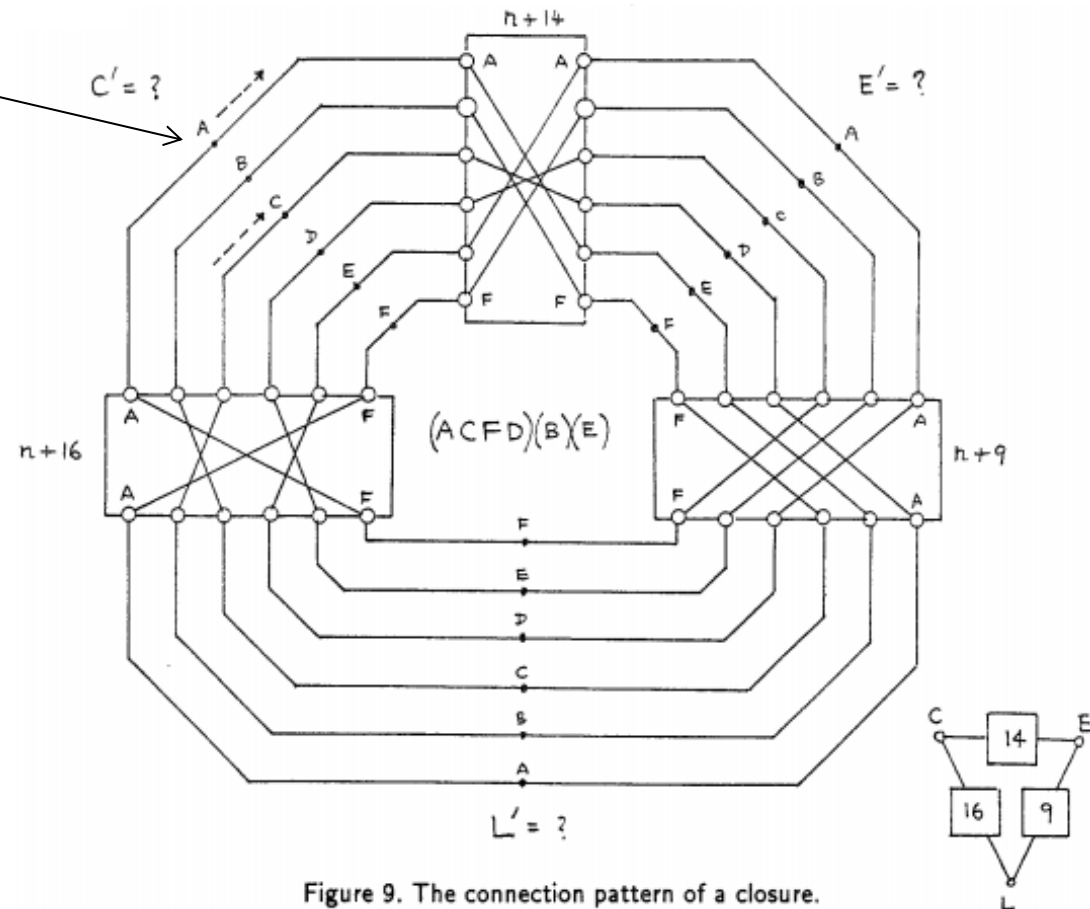
- 26 wires become 'live'

- what remains:  
0 fixed points,

(machine continues  
to next setting, rotate all drums)

$$R_7 \circ R_9 \circ R_{14} \circ S(E) = S(E)$$

In a closed loop



## Philosophy 1920s-Modern Attacks

For any cipher old/modern

1. Guess  $X$  bits (subset of the key)
2. Deduce  $Y$  bits
3. Find contradiction  
(large proba  $P=1$ -small)



## Turing Attack = Crib Loops [Short Cycles]

3. Allows to reject **Stop when no contradiction found A→A again**  
testing start pos+1 stecker connection...

$26^3$  settings at 1800rpm, 11 minutes to check  $26^3$  settings,  
>90% of possibilities for **S(E)** could be rejected in 1 clock  
⇒most wires active, 123 left = simultaneous scanning, electrical current was much faster than the mechanical movement of rotors.  
⇒remark: **most of the time spent rejecting settings**, false positives can be treated by additional checks with a bombe or another machines

## Turing Attack = Crib Loops [Short Cycles]

3. Allows to reject **Stop when no contradiction found A→A again**  
settings+stecker connections...

### Price to pay:

- guess 3/5 rotors+order – 10.6

- guess settings of rotors –  $26^3$

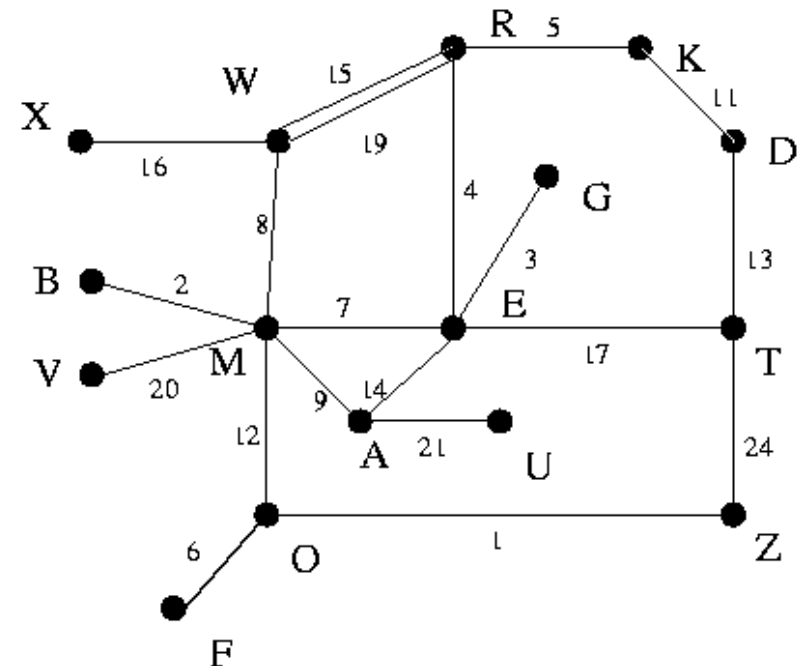
- guess some equation like  $S(E)=B - 26^{-1}$


Two loops with letter E => machine stops every  $26^{3-1-1}$  steps in a plausible configuration... NOT good enough!!! Too many false positives



## Turing Attack = Crib Loops [Short Cycles]

not directed, no  
arrows,  
bi-directional  
relations



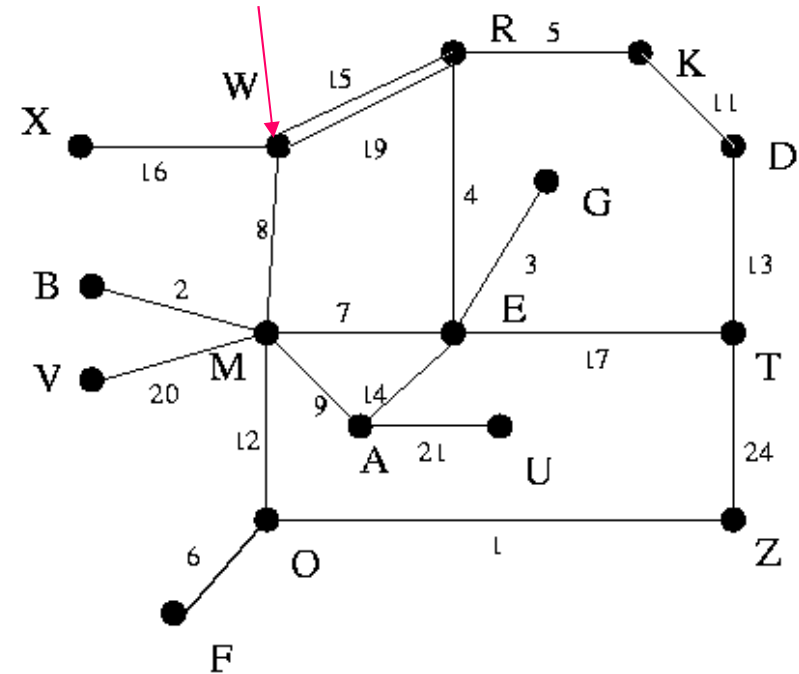
3. Each loop allows to reject 25/26 of cases guesses...
4. Did NOT work well. (too few loops=>long cribs, 20 characters + several loops preferred)
- 

## Turing Attack = Crib Loops [Short Cycles]

4. Did NOT work well. (too few loops=>long cribs, 20 characters)
5. Improved by Welchman:  
many extra deductions, less false positive stops,  
=> 10+ letter cribs only required!!!

Turing Attack = 1 cycle, 1 'central' letter [at place with several connections]

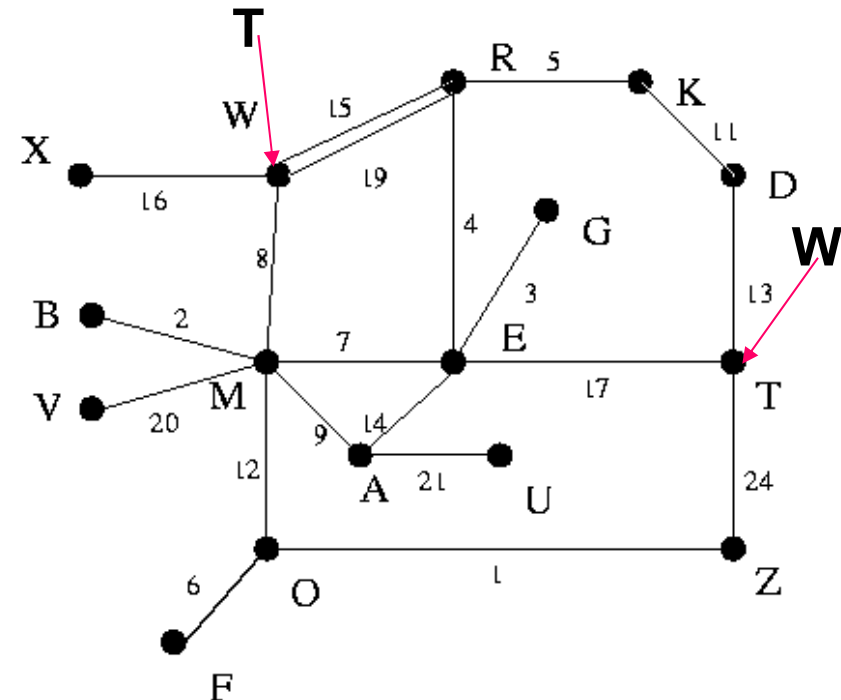
guess  $S(W)$   
=>test it



## Welchman - Observation

guess  $S(W)=T$   
 $\Rightarrow$  test it

Remark that T appears also  
 in our menu:  
 we get  
 2 guesses for the price of 1!  
 (amplification)



...and this goes a lot further.

## Contd.

The Turing attack has used 1 loop to find contradictions most of the time, and with 2-3 loops/chains it would stop more rarely, but still many false alarms.

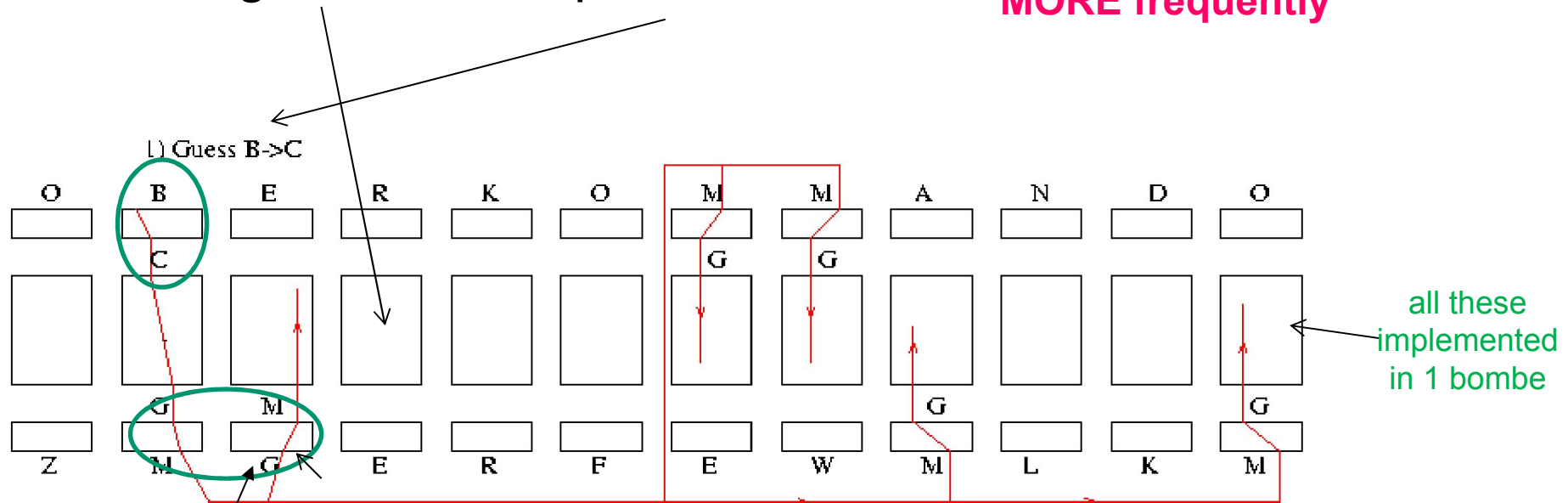
Welchman has found how to CONNECT circuits for several loops/chains together, resulting in dramatically improved capability to find contradictions for 1 assumption => less stops => shorter cribs.

Any pair of “nodes” can be connected with the diagonal board.

## Welchman Bombe Deductions [Diagonal Board]

## Testing 3 rotors + 1 pair for S

## Contradiction found MORE frequently



another place  
same menu  
letter G

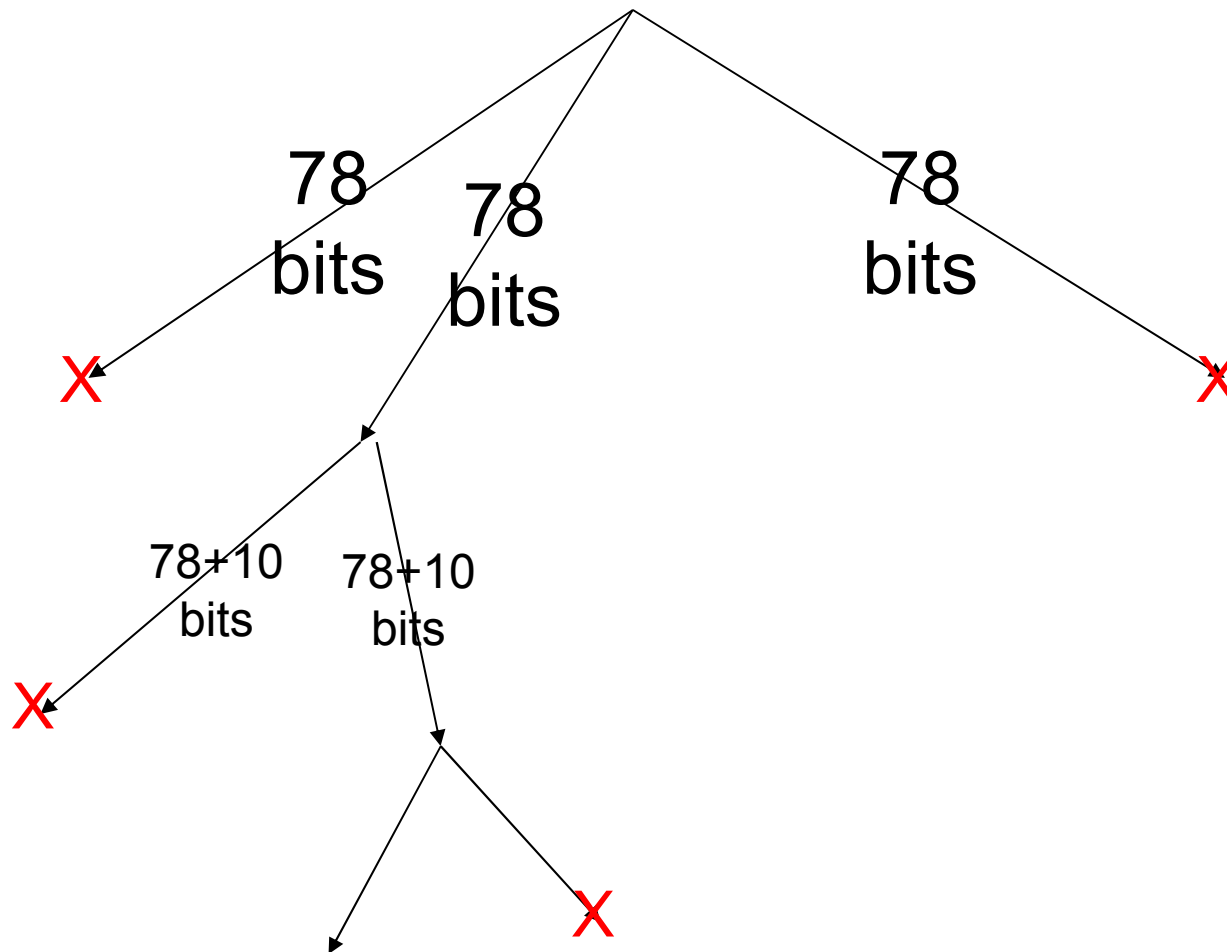
Welchman Exploited involution  
pty of the stecker=>more deductions

allowed shorter cribs (10+ letters)=>  
huge savings!!!!

# Guess-Then-Determine or UNSAT Attack

↑ Guess Then Eliminate ↓

## Depth-First Tree Search.





## Guess-Then-Determine: Amplification



## Amplification

**Definition 3.2.1 (Amplification, Informal).** The goal of the attacker is to find a reduction where he makes some assumption at a certain initial cost, for example they are true with probability  $2^{-X}$  or work for certain proportion  $2^{-Z}$  of keys. Then the attacker can in constant time determine many other internal bits inside the cipher to the total of  $Y$  bits.

We call amplification the ratio  $A = Y/X$ .

We are only interested in cases in which the values  $X$  and  $Z$  are judged realistic for a given attack, for example  $Z < 32$  and  $X < 128$ .

Killer example:

- Slide attacks – unlimited.



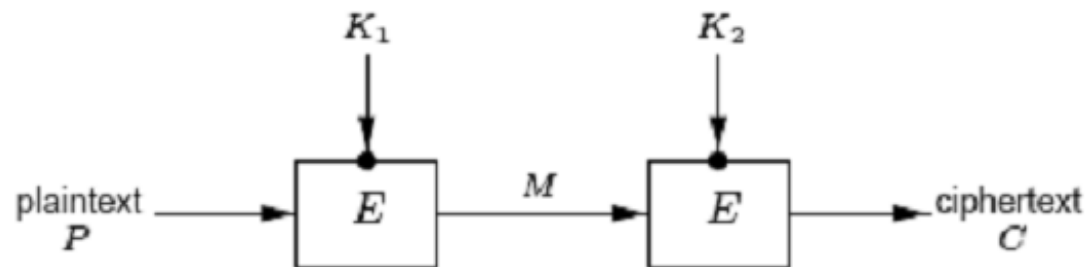
# High-Level Attacks



# Multiple Encryption with a Block Cipher

key sizes?

(a) double encryption



## Multiple Encryption

Practical question:

DES = US government

increase key size ?

## Double Encryption

Let's try  $2 * \text{DES}$  with 2 keys.

Security = 112 bits ?  
**NOT AT ALL !**

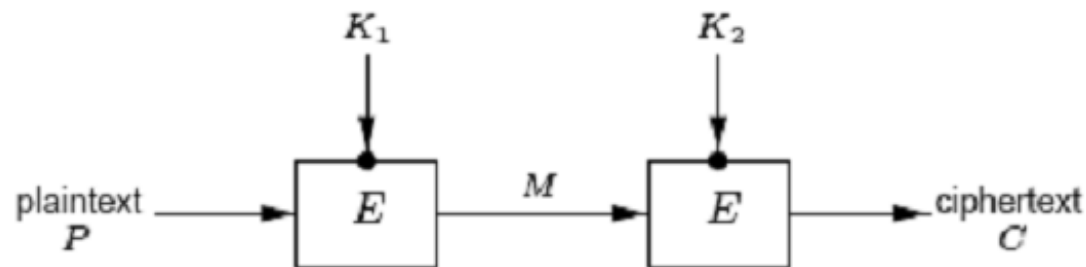
## Double Encryption – Not Good

Meet-in-the middle attack.

$k$ -bit cipher.  $n$ -bit block,  $n \geq k$ .

- Given: 1 (!) known plaintext.

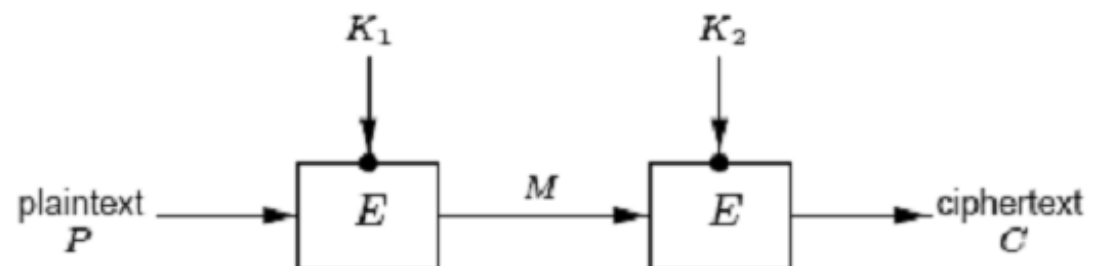
(a) double encryption



## Meet-in-the middle attack.

- Encrypt  $P$  with all  $2^k$  keys. Store the answers (in a “hash table” or sorted table.)
- Decrypt  $C$  with all  $2^k$  keys. Find if one is in the table.
- Both keys have been found.

(a) double encryption

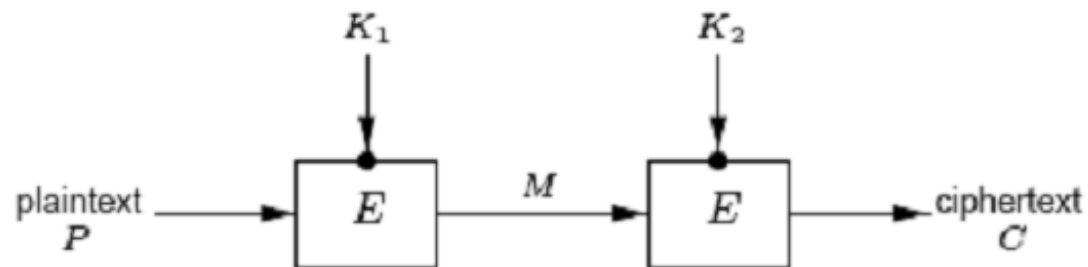




## Lesson Learned

Both keys must be long (e.g. full size)

(a) double encryption





## Sliding Attacks



# Sliding Attacks [1977]

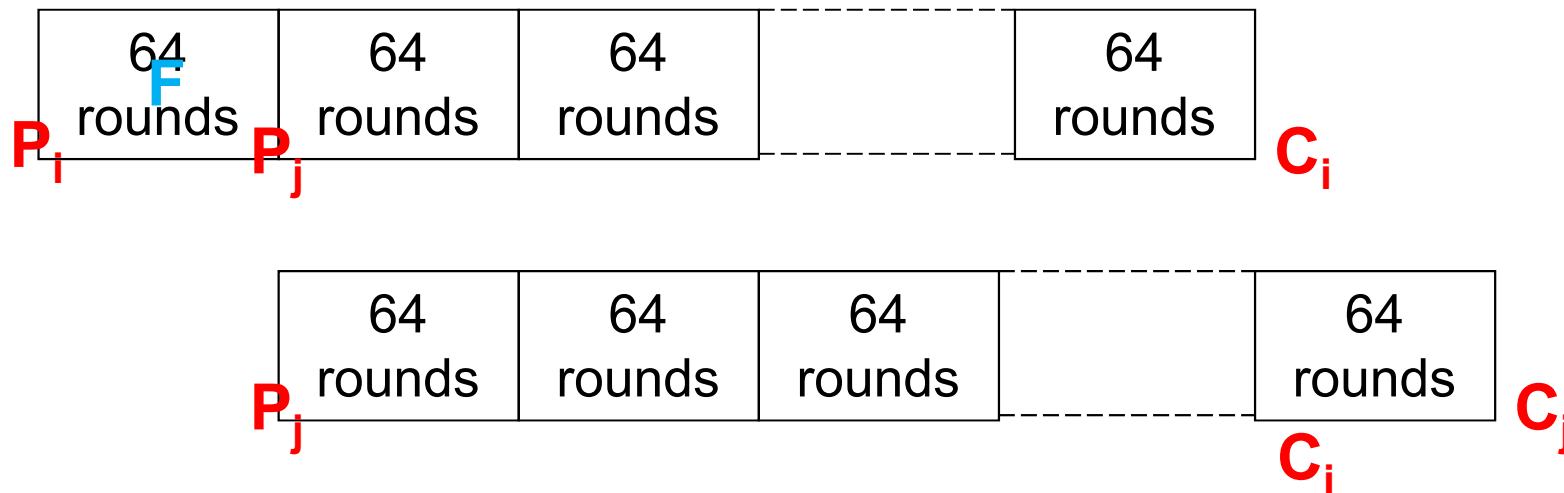
- **Periodic Cipher**

F	F	F
---	---	---

# Sliding Attack

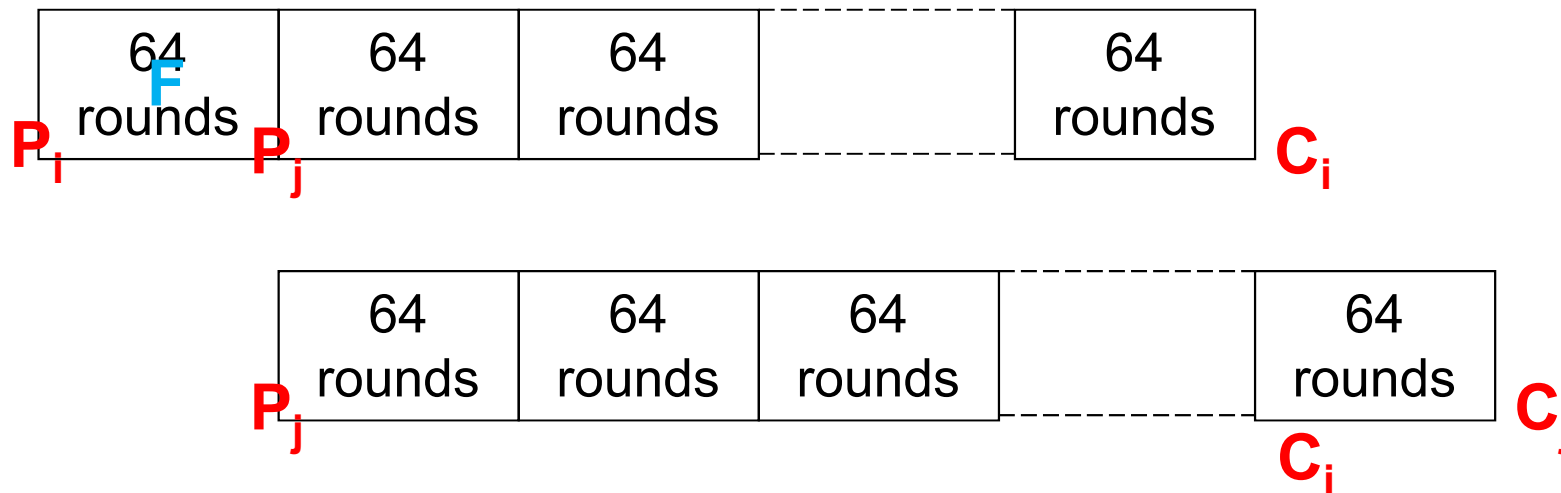
Classical Sliding Attack [Grossman-Tuckerman 1977]:

- Take  $2^{n/2}$  known plaintexts
- Imagine that we have some “slid pair”  $(P_i, P_j)$  s.t.



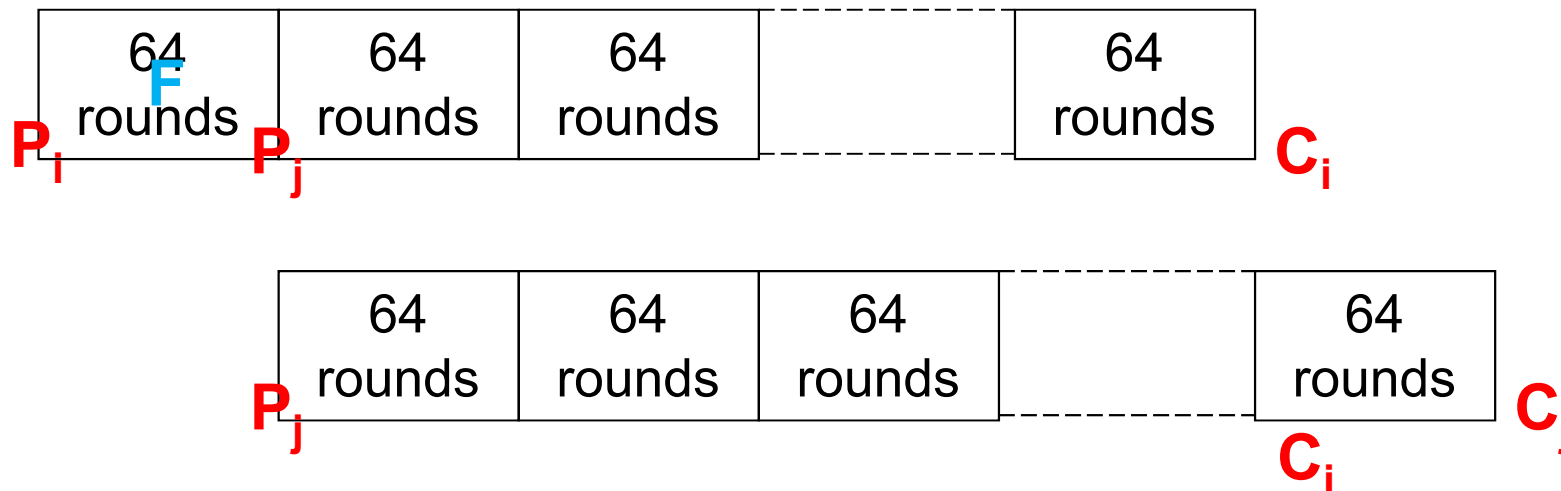
# One Step

- Assumption:  $F(P_i) = P_j$   $n$  bits
- Consequence:  $F(E_k(P_i)) = E_k(P_j)$   $2n$  bits, Amp.=2



# One Step

- Assumption:  $F(P_i) = P_j$   $n$  bits
- Consequence:  $F(E_k(P_i)) = E_k(P_j)$   $2n$  bits, Amp.=2



**THIS CAN be iterated!**

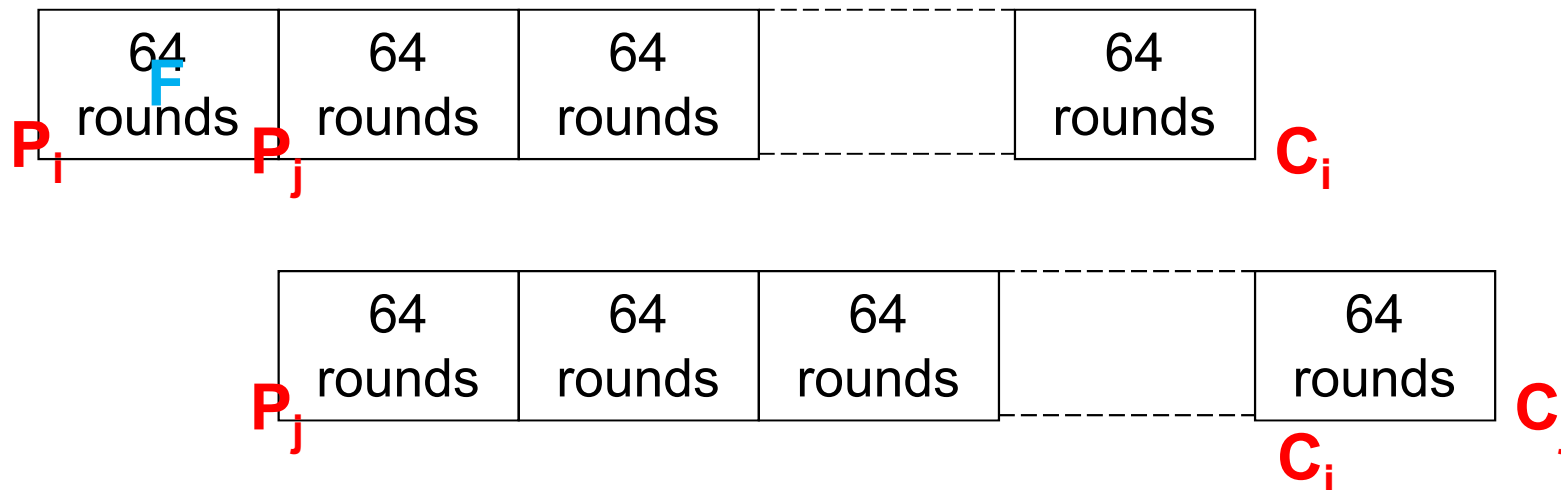
# One Step

- Assumption:  $F(P_i) = P_j$   $n$  bits
- Consequence:  $F(E_k(P_i)) = E_k(P_j)$   $2n$  bits, Amp.=2
- Also:  $F(E_k^2(P_i)) = E_k^2(P_j)$   $3n$  bits, Amp.=3
- $\dots$   $\dots$   $\dots$
- $\forall m$   $F(E_k^m(P_i)) = E_k^m(P_j)$  **Unlimited!**

# Sliding Attack

Classical Sliding Attack [Grossman-Tuckerman 1977]:

- Take  $2^{n/2}$  known plaintexts
- We have a “slid pair”  $(P_i, P_j)$  s.t.



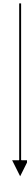
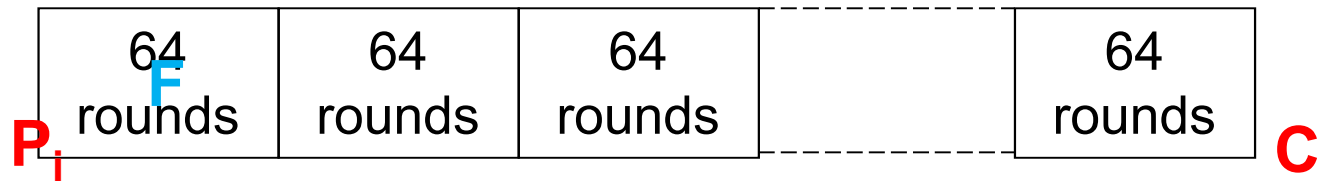
**Gives an unlimited number of other sliding pairs !!!**

=>unlimited amplification



## \*Black Box Reduction

We transform a CPA on  $E_k$



into a KPA on  $F$

many pairs!!!  
=>a lot easier to break!

# High-Level Attacks



## KeeLoq Cipher

- Designed in the 80's by Willem Smit.
- In 1995 sold to Microchip Inc for more than 10 Million of US\$.



## How Secure is KeeLoq

According to Microchip, KeeLoq should have "a level of security comparable to DES". Yet faster.

Miserably bad cipher, main reason:

its **periodic** structure: cannot be defended. The complexity of most attacks on KeeLoq does **NOT** depend on the number of rounds of KeeLoq.

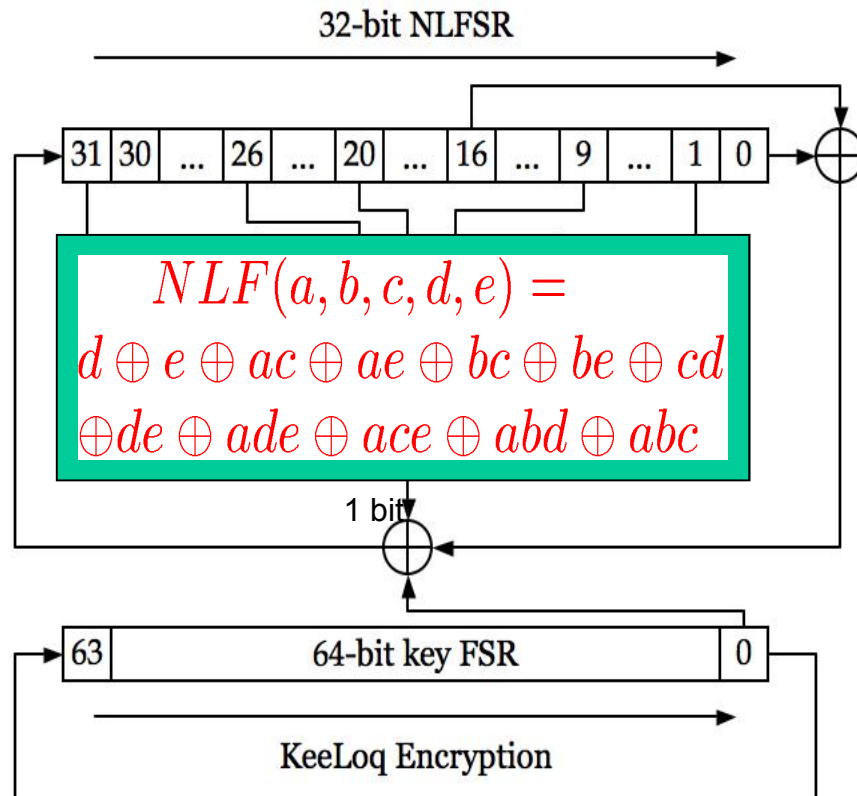


# Description of KeeLoq

# KeeLoq Encryption

Block Cipher

- Highly unbalanced Feistel
- 528 rounds
- 32-bit block / state
- 64-bit key
- 1 bit updated / round
- 1 key bit / round only !



# Notation

$f_k()$  – 64 rounds of KeeLoq

$g_k()$  – 16 rounds of KeeLoq, prefix of  $f_k()$ .

We have:  $E_k = g_k \circ f_k^8$ .  
 $528 = 16 + 8 \cdot 64$  rounds.

# Frontal Algebraic Attack on KeeLoq

KeeLoq can be implemented using about 700 GE.

=> “direct” algebraic attack: write equations+solve.

Two methods:

- ElimLin/Gröbner bases
- Conversion+SAT solvers.



## What Can Be Done ?

As of today, we can:

ElimLin (Method 1):

With ElimLin we can break up to **128** rounds of KeeLoq faster than brute force.

**128 KP** counter mode.

Conversion+MiniSAT (Method 2):

As much as **160** rounds of KeeLoq and only

**2 KP** (cannot be less).

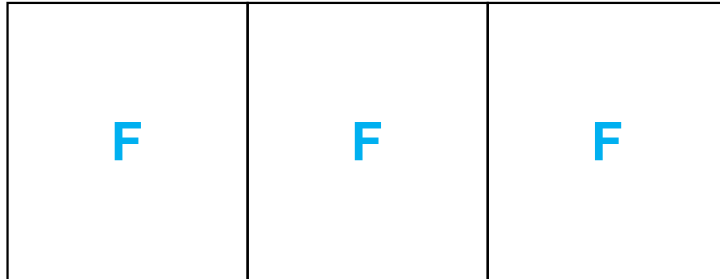
## Beyond?

KeeLoq has additional weaknesses.

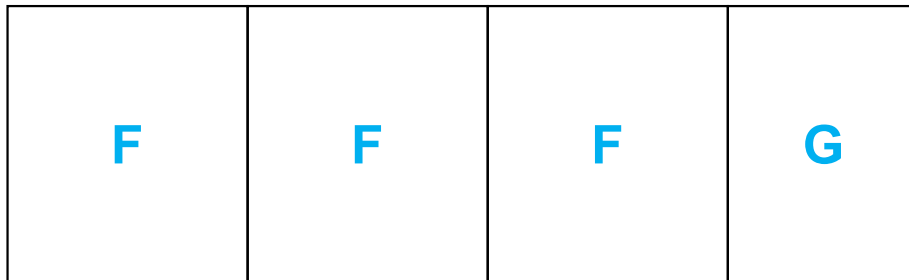
There are much better attacks.

# Sliding Attacks – 2 Cases

- **Complete periodicity [classical].**



- **Incomplete periodicity [new] – harder.**

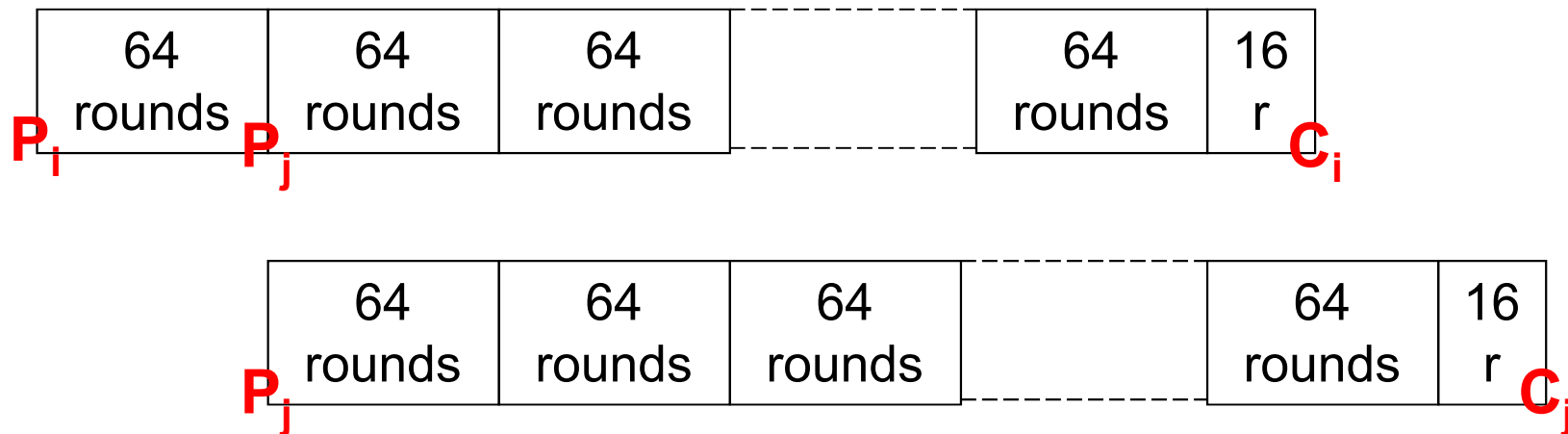


– **KeeLoq: G is a functional prefix of F. Helps a lot.**

# KeeLoq and Sliding

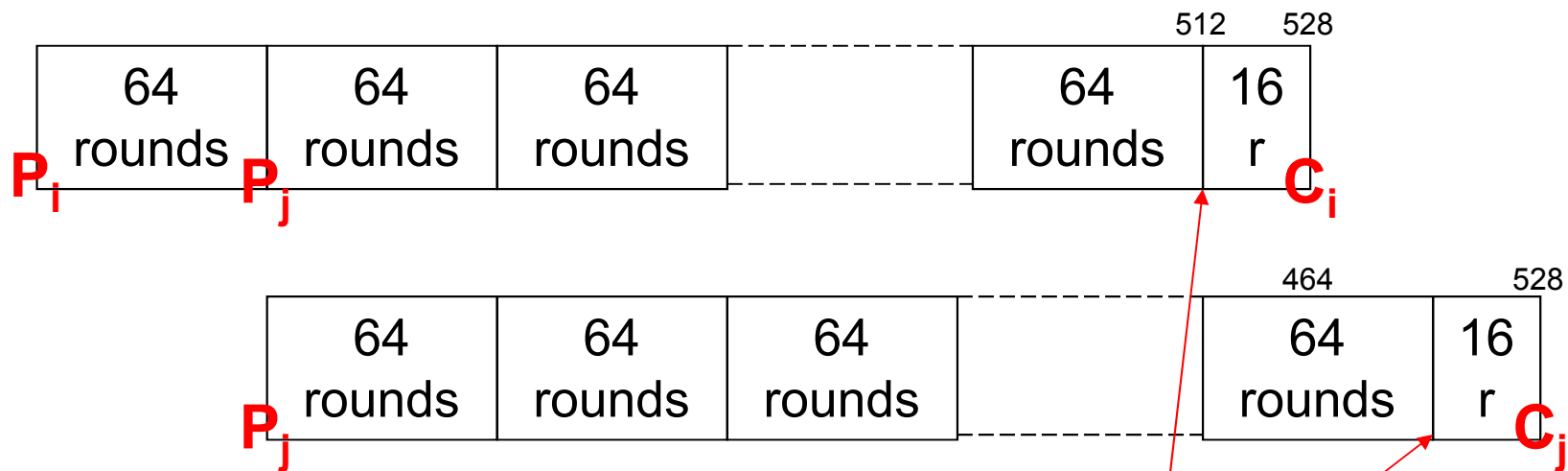
Apply Classical Sliding? Attack 1.

- Take  $2^{n/2}$  known plaintexts (here  $n=32$ , easy !)
- We have a “slid pair”  $(P_i, P_j)$  s.t.



Classical sliding fails – because of the “odd” 16 rounds:

# Classical Sliding –Not Easy



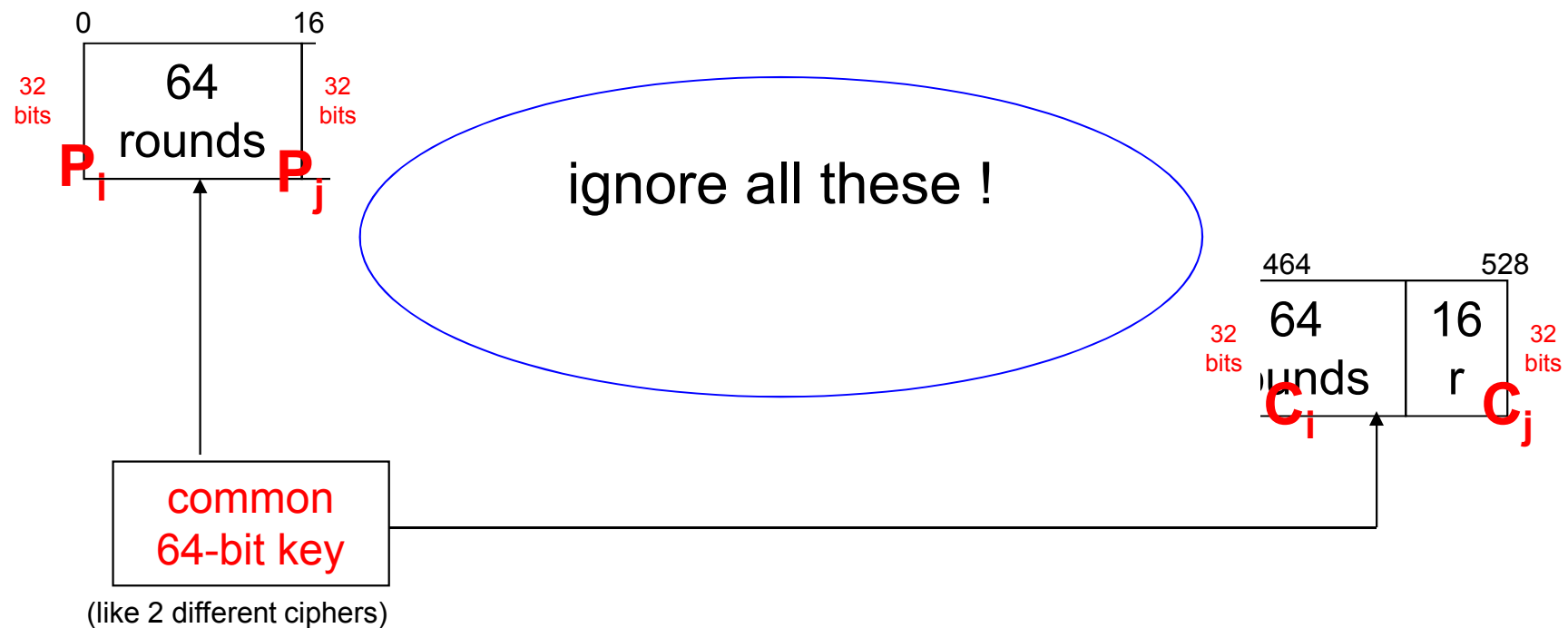
**HARD** - Problem:

What's the values here ?

# Algebraic Attack:

We are able to use  $C_i, C_j$  directly !

Merge 2 systems of equations:



# System of Equations

64-bit key. Two pairs on 32 bits.  
Just enough information.

Attack:

- Write a system of equations.
  - Gröbner Bases methods – miserably fail.
- Convert to a SAT problem
  - [Cf. Courtois, Bard, Jefferson, [eprint/2007/024/](http://eprint/2007/024/)].
- Solve it.
  - Takes 2.3 seconds on a PC with MiniSat 2.0.

## Attack Summary:

Given about  $2^{16.5}$  KP.

We try all  $2^{32}$  pairs  $(P_i, P_j)$ .

- If OK, it takes 2.3 seconds to find the 64-bit key.
- If no result - early abort.

Total attack complexity about  $2^{32+32}$  CPU clocks  
which is about  $2^{53}$  KeeLoq encryptions.



# What Happened?

**Power of Algebraic Attacks:** Any cipher that is not too complex is broken... (!)

- **Problem:** We hit the “wall” when the number of rounds is large.

**Power of Sliding Attacks:** their complexity does NOT depend on the number of rounds.

These two **combined** give a first in history successful algebraic attack on an industrial block cipher.

## KeeLoq is badly broken

Practical attack, tested and implemented:

Courtois, Bard, Wagner: Algebraic and Slide Attacks on  
KeeLoq in FSE 2008

# Another Attack on KeeLoq

## [Tatrachrypt 2007]

## \*\*All Known Attacks

**Table 1.** Various attacks on KeeLoq in order of increasing plaintext requirements

Type of attack	Data	Time	% Keys	Memory	Reference
Brute Force	2 KP	$2^{63}$	100%	small	
B. F. + Precomp. Speed-up	<b>2 KP</b>	<b><math>2^{62}</math></b>	100%	16 Gb	NEW: this paper
Brute F. + Self-Similarity	<b>2 CP</b>	<b><math>2^{61.4}</math></b>	100%	small	NEW: this paper
Brute F. + Self-Similarity	<b>2 CP</b>	<b><math>2^{60.4}</math></b>	100%	16 Gb	NEW: this paper
Brute F. + Self-Similarity	<b>2 CP</b>	<b><math>2^{57}</math></b>	11%	small	NEW: this paper
B.F. + Self-Sim. + Precomp.	<b>2 CP</b>	<b><math>2^{56}</math></b>	11%	16 Gb	NEW: this paper
Slide-Algebraic	$2^{16}$ KP	$2^{53}$	63% *	small	Slide-Alg. Attack 2 in [11]
Slide-Meet-in-the-Middle	$2^{16}$ KP	$2^{46}$	63% *	3 Mb	Dunkelman <i>et al</i> [2]
Slide-Meet-in-the-Middle	$2^{16}$ CP	$2^{45}$	63% *	3 Mb	Dunkelman <i>et al</i> [2]
Slide-Correlation	$2^{32}$ KP	$2^{51}$	100%	16 Gb	Bogdanov[4, 5]
Slide-Fixed Points	$2^{32}$ KP	$2^{43}$	26%	16 Gb	Attack 4 in eprint/2007/062/
Slide-Cycle-Algebraic	$2^{32}$ KP	$2^{40}$	63%	18 Gb	Attack A in [13]
Slide-Cycle-Correlation	$2^{32}$ KP	$2^{40}$	100%	18 Gb	Attack B in [13]
Slide-Determine	$2^{32}$ KP	$2^{31}$	63%	16 Gb	Version A in [11]
Slide-Determine	$2^{32}$ KP	$2^{28}$	30%	16 Gb	Version B in [11]
New improved versions in [12]:					
Slide-Determine	$2^{32}$ KP	<b><math>2^{30}</math></b>	63%	16 Gb	Overall average time [12]
Slide-Determine	$2^{32}$ KP	<b><math>2^{27}</math></b>	30%	16 Gb	'Realistic' version, [12]
Slide-Determine	$2^{32}$ KP	<b><math>2^{23}</math></b>	15%	16 Gb	'Optimistic' version, [12]

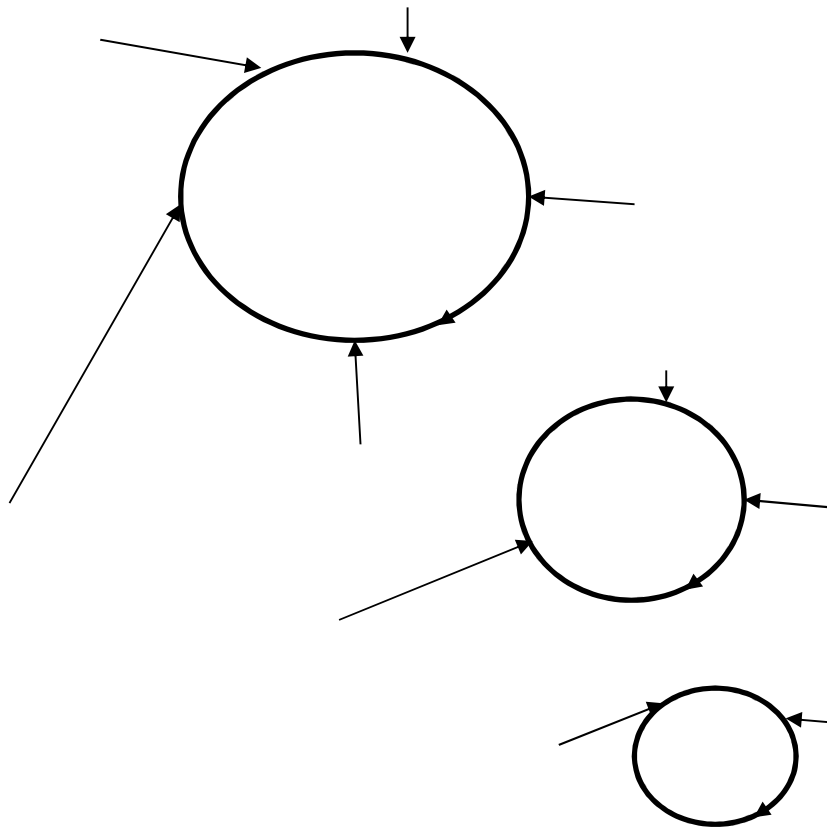
Legend: The unit of time complexity here is one KeeLoq encryption.

# Iterated Permutation Attacks [Tatracrypt07]

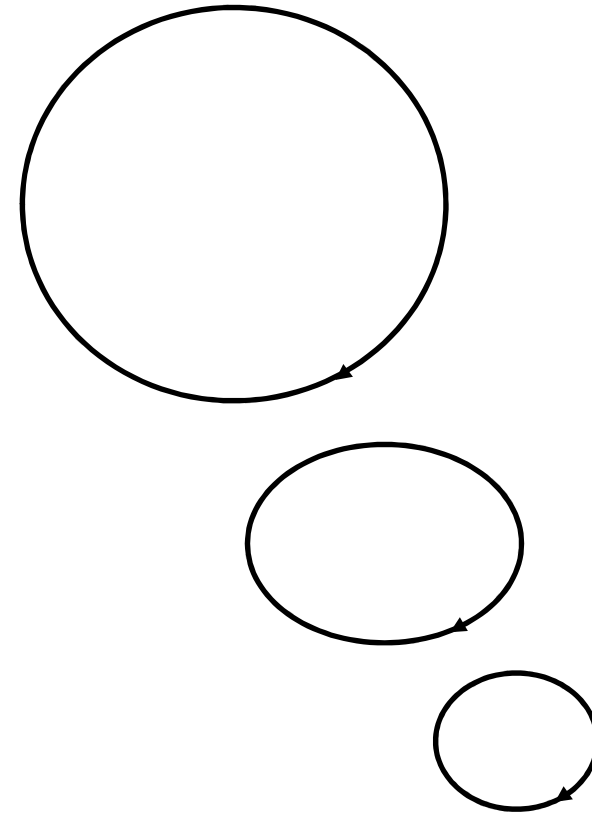
$$E_k = g_k \circ f^8_k.$$


- Guess **16** key bits.
- Confirm if correct. (!)
- Recover missing key bits by
  - an algebraic attack.
  - correlation attack
  - other..

## Cycles in RF/RP



Function



Permutation

# Random Functions

$n$  bits  $\rightarrow$   $n$  bits

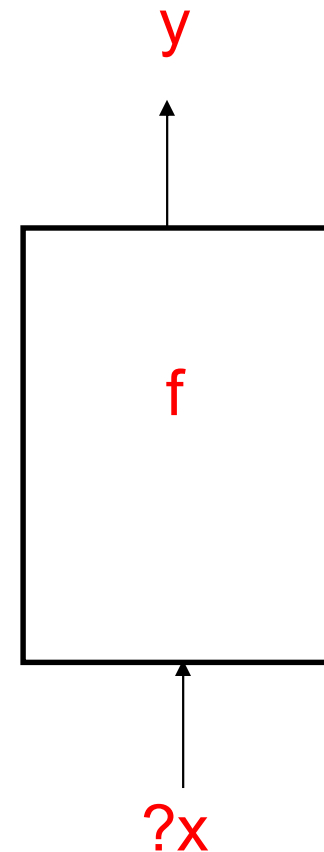
The probability that a given point has  $i$  pre-images is  $1/ei!$ .

Fixed points:

number of fixed points of  $f(x) \Leftrightarrow$

number of points such that  $g(x)=0$

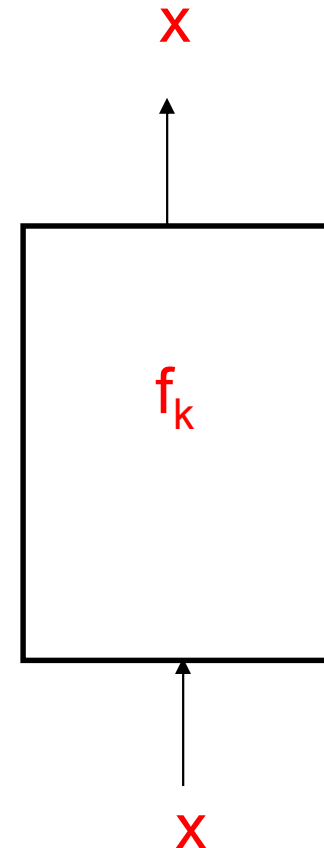
with  $g(x) = f(x)-x$ .



# Fixed Points for 64 rounds of KeeLoq

$f_k$  is expected to have at least 1 fixed points  
for  $1-1/e \approx 0.63$  of all keys.

$f_k$  is expected to have at least 2 fixed points  
for  $1-2/e \approx 0.26$  of all keys.





# Cycles for 64 Rounds of KeeLoq

$n$  bits  $\rightarrow$   $n$  bits

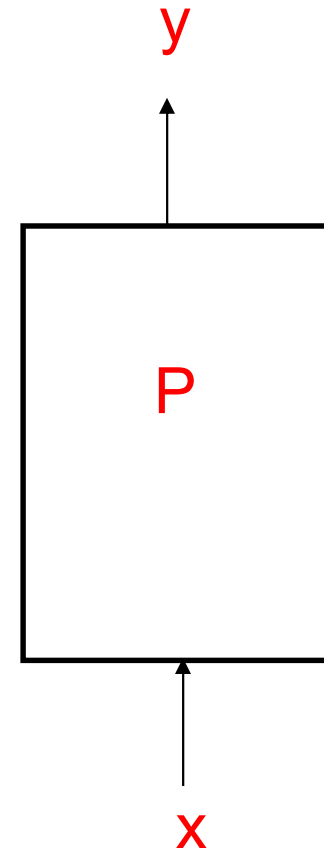
**Theorem.** The expected number of cycles in a permutation on  $n$  bits is equal to  $H_2^n$  where

$$H_k = \sum_{i=1}^k \frac{1}{i} \approx \ln k + \gamma$$

is the  $k$ -th Harmonic number

$$\gamma \approx 0.58$$

$\gamma$  is the Euler-Mascheroni constant



# Cycles and Random Permutations

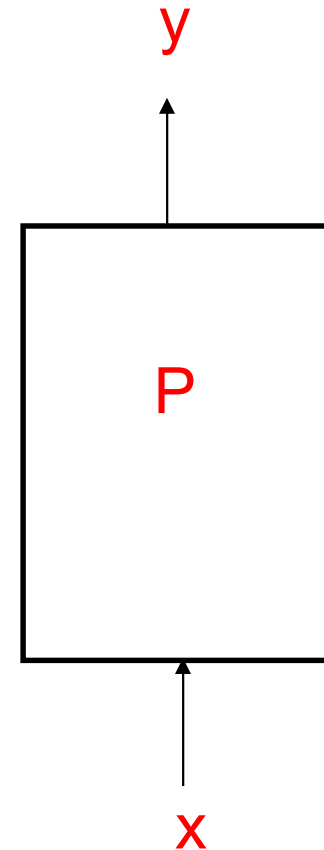
$n$  bits  $\rightarrow$   $n$  bits

**Corollary.**

$n=32 \Rightarrow 23$  cycles on average

(of decreasing size  $2/3 \cdot 2^{32} \dots 1$ ).

About 11 are of odd size.



## Fact:

↑RP

If we have (a nearly complete) table of  $F^8$  and  $E$ , for two permutations, it is easy to distinguish them.  
(this will be done for  $f_k$  without knowing the key).

Why ?

What happens when we iterate a permutation ( $F^2$ ):

- Cycles of even size split in two
- Cycles of odd size remain.

So between  $F$  and  $F^2$  we expect that the number of even-size cycles is divided by two.

if we do not count  
cycles with repeated  
sizes!

## Odd Cycles in $P^8$

So between  $P$  and  $P^2$  we expect that the number of even-size cycles is divided by two.

$$F \rightarrow F^2 \rightarrow F^4 \rightarrow F^8$$

$$11.5 \rightarrow 5.75 \rightarrow 2.8 \rightarrow 1.4$$

if we count  
without  
multiplicity

So we expect that  $P^8$  has 1-2 even-length cycles instead of 11-12. Our distinguisher has negligible probability of being wrong...

## Attack 3

Notation:

$f_k()$  – 64 rounds of KeeLoq

$g_k()$  – 16 rounds of KeeLoq, prefix of  $f_k()$ .

We have:  $E_k = g_k \circ f_k^8$ .

$528 = 16 + 8 \cdot 64$  rounds.

## Attack 3

$$E_k = g_k \circ f^8_k.$$

- Guess **16** key bits.
- Confirm if correct. (!)

## Attack 3

$$E_k = g_k \circ f_k^8.$$


The diagram shows three arrows originating from the equation  $E_k = g_k \circ f_k^8$ . One arrow points from  $g_k$  to the bullet point. Another arrow points from  $f_k^8$  to the bullet point. A third arrow points from the composition symbol  $\circ$  to the bullet point.

- Recover missing key bits by an algebraic attack.

## Attack 3

What we get so far?

16 key bits =>

48 bits remain to be recovered.

We still need to break KeeLoq with 512 rounds and 48 bit key.

Completely from the start.

EASY: classical sliding attacks are possible => reduction to 64 rounds of KeeLoq.

We propose ANOTHER method now.



## Completing Attack 3

We assume that  $f_k$  has 2 or more fixed points  
=> this attack works  
for  $1 - 2/e \approx 0.26$  of all keys.

We expect that  $f_k^8$  has about 6 fixed points.

We can compute all of them, guess which two are  
fixed points also for  $f_k$ .

Works with probability  $1 / \binom{6}{2} \approx 1/15$ .

## Completing Attack 3

Then given two fixed points, the missing 48 bit of the keys are computed in 0.2 s with MiniSat 2.0.

This is about  $2^{28}$  CPU clocks.

## Summary of Attack 3

1. Guess **16** key bits.

2. Confirm if correct.

**$2^{16} * 2^{32}$**  operations in RAM.

Assuming accessing RAM takes **16** CPU clocks,  
about  **$2^{52}$**  CPU clocks.

3. Recover missing key bits by an algebraic attack.

This is about  **$15 * 2^{28}$**  CPU clocks. Negligible.

## Part 5

# Inside The Box



**Fig. 46.** General principle of “Induction”: relations in the middle may occur with high probability due to a combination of constraints from both sides

P,Q = two keyed permutations

## Part 5.0

# Involutions

## Involutions

Theorem: Let  $Q$  be an involution.

The expected number of fixed points is as large as  $2^{n/2}$  instead of  $O(1)$  in a random permutation.

Proof:

see page 596 of Philippe Flajolet, Robert Sedgewick,  
Analytic Combinatorics, Cambridge University Press.

$\Rightarrow$  We already had this all over the place in our works,  
“semi-transparent cylinder” syndrome [Courtois],

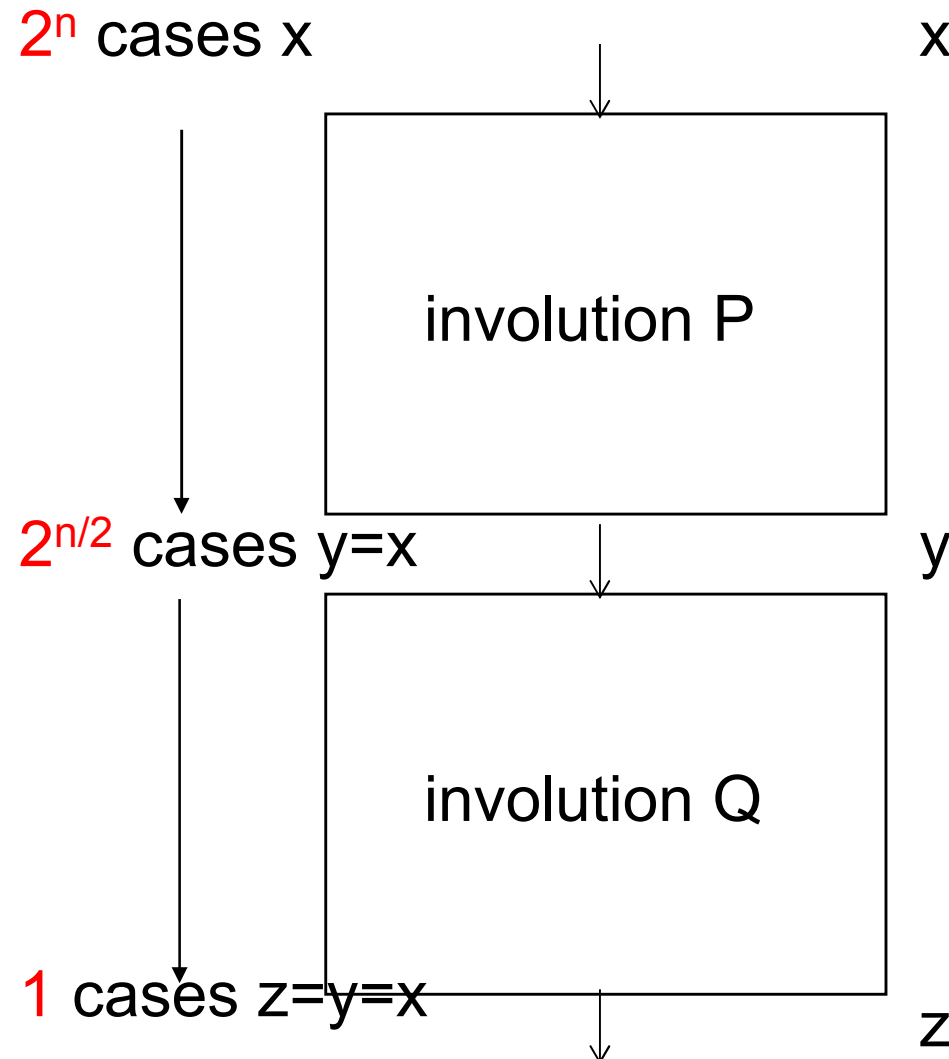
# Two Involutions

$$AD = CPNP^{-1}QPN^{-1}P^3NP^{-4}QP^4N^{-1}P^{-4}C^{-1}$$

Enigma, 1930s...

$$S^{-1} \circ R_1 \circ S \circ S^{-1} \circ R_4 \circ S$$

## Seeing Inside!



Large cost...  
 $\Rightarrow$  we can do  
 much better...



# Two Involutions

**Fact 19 (Rejewski Theorem).** Let  $Q \circ P$  be a composition of two involutions without fixed points. The number of cycles of each length  $k$  for  $Q \circ P$  is an even number.

Moreover these cycles are in a one-to-one correspondence induced by  $P$ , the inverse of which is a one-to-one correspondence induced by  $Q$ .

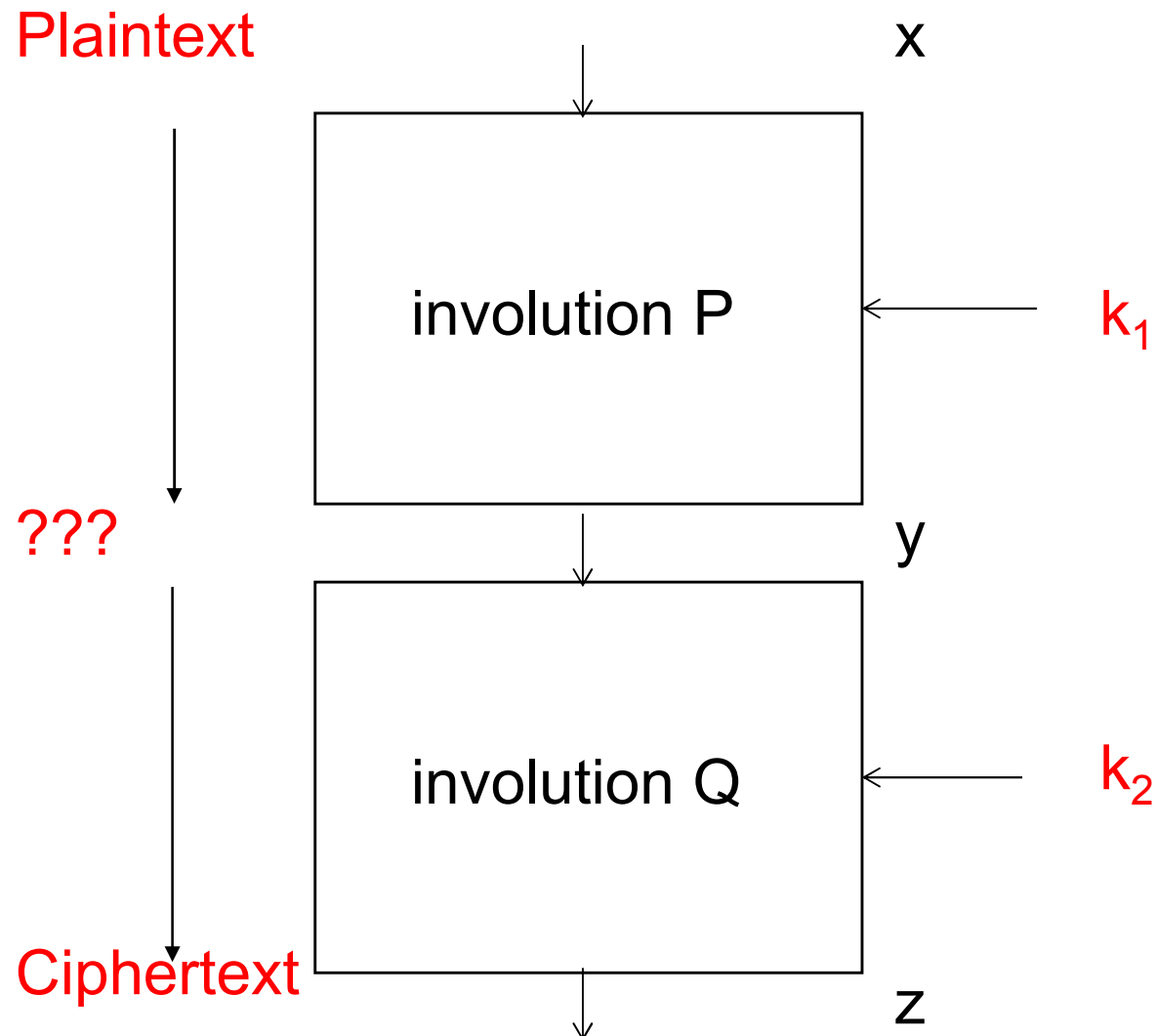
**Proof:** simple proof. Let  $X$  be a point which lies on a cycle of length  $k$ , and does not lie on a shorter cycle. Then  $X$  is a fixed point of  $(Q \circ P)^k$ . However because both are involutions, the same  $X$  is also a fixed point for its inverse permutation which is simply  $(P \circ Q)^k$ . Then  $Q(X)$  is also a fixed point for  $(Q \circ P)^k$ .

We see that each time  $X$  lies on a cycle of length exactly  $k$  and not on a shorter one, also  $Q(X)$  lies on a cycle of the same exact length, which cannot be shorter because this property holds for every point on this cycle and  $Q$  is bijective. Now can  $X$  and  $Q(X)$  ever lie on the same cycle (and the two cycles would merge)? This means that either we have  $X = Q(X)$  which is excluded because we assumed that  $Q$  had no fixed points, or that  $X = (Q \circ P)^k(X)$ , for some smaller  $k$ , however we assumed there was no shorter cycle for  $X$ . Therefore the bijection  $X \mapsto Q(X)$  maps whole cycles to whole cycles which are distinct from the original cycle. Now this bijection  $Q$ , since  $Q$  is an involution, is clearly one-to-one when acting on cycles and no cycle is transformed onto itself. Thus we get an even number of cycles of each length  $k$ . We also remark that the inverse mapping acting on whole cycles will be the one induced by  $P$ .

# Factoring Permutations

quite surprising  
(already used for Enigma cf. slide 47)

## Key Recovery?



# Factoring Permutations – Miracle!

**Fact 20 (Rejewski Permutation Factoring Method).** Let  $Q \circ \mathcal{P}$  be a composition of two involutions, and let  $\mathcal{P}$  have  $p$  rounds and let  $Q$  have  $q$  rounds with  $p \leq q$ . We assume that the attacker has oracle access to  $Q \circ \mathcal{P}$ . We assume that there is a key recovery attack on  $\mathcal{P}$  given the fact that it has only  $p$  rounds, and that this attack requires only a limited number of P/C pairs. Then attacker can factor  $Q \circ \mathcal{P}$  and recover the key of  $\mathcal{P}$ .

## Proof:

*Justification:* We apply Fact 19 above and consider the smallest value  $k$  such that  $Q \circ \mathcal{P}$  has exactly 2 cycles of length  $k$ , which following Fact 19 must be related and one cycle is  $X, Q(\mathcal{P}(X)), \dots$  the other is  $\mathcal{P}(X), \mathcal{P}(Q(\mathcal{P}(X))), \dots$  possibly starting at some location inside the other cycle. We just need to guess which cycle is which, pick a random point on once cycle, and guess which point on the second cycle is the corresponding points. Overall with probability  $\frac{1}{2k}$  we obtain as many as  $k$  correct P/C pairs for  $\mathcal{P}$  which should be sufficient for key recovery.

## Part 5.1

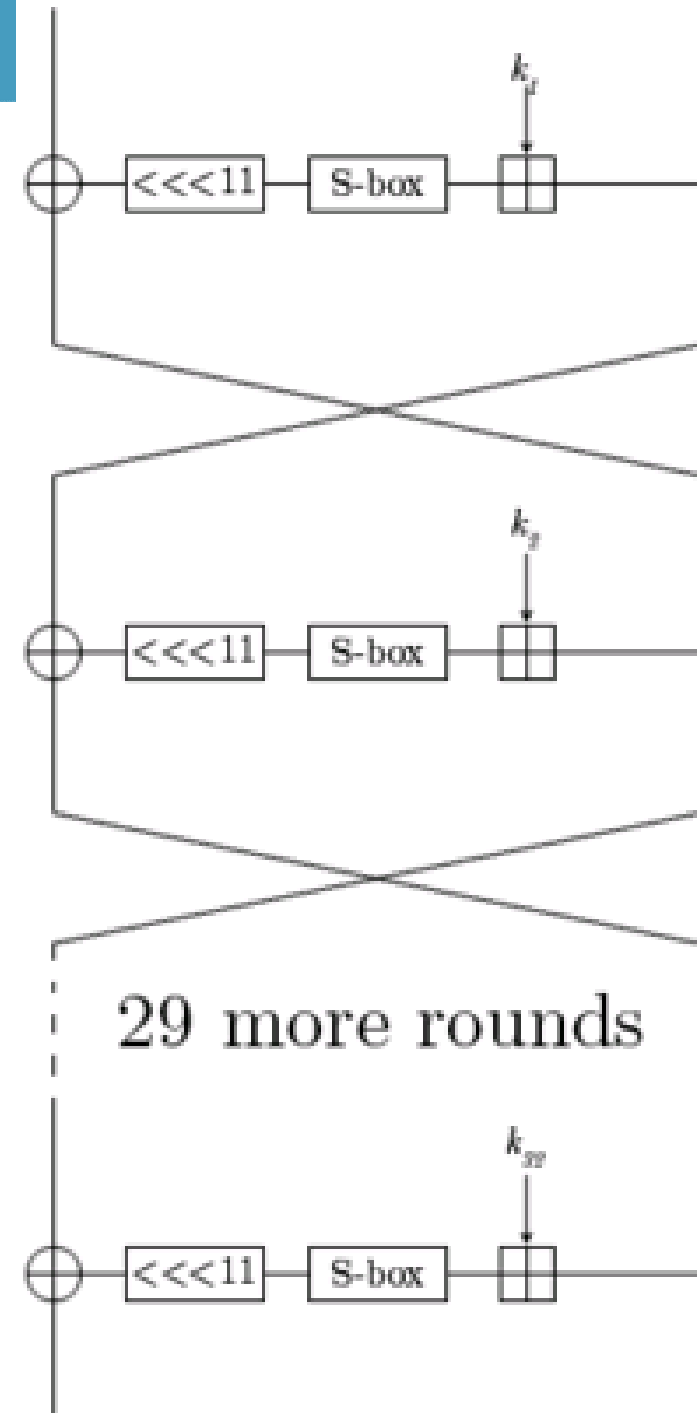
# Involutions In Modern Block Ciphers

# GOST Cipher

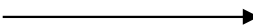


# GOST

- Key =  $2^{256}$  initial settings.
- S-boxes =  $2^{512}$  possibilities.
  - But if bijective  $2^{354}$  possibilities.
- Total  $2^{610}$  (or  $2^{768}$ ).



## GOST Boxes

- 8 secret S-boxes. (354 bits of info)
  - Central Bank of Russia uses these: 
- Secret S-boxes are the equivalent of secret rotors in FIALKA
- Our attacks work **for any S-boxes** but they must be known.
  - there are methods about how to recover the secret S-boxes...

#	S-Box
1	4 10 9 2 13 8 0 14 6 11 1 12 7 15 5 3
2	14 11 4 12 6 13 15 10 2 3 8 1 0 7 5 9
3	5 8 1 13 10 3 4 2 14 15 12 7 6 0 9 11
4	7 13 10 1 0 8 9 15 14 4 6 12 11 2 5 3
5	6 12 7 1 5 15 13 8 4 10 9 14 0 3 11 2
6	4 11 10 0 7 2 1 13 3 6 8 5 9 12 15 14
7	13 11 4 1 3 15 5 9 0 10 14 7 6 8 2 12
8	1 15 13 0 5 7 10 4 9 2 3 14 6 11 8 12




## Consensus on GOST Security [2010]

Axel Poschmann, San Ling, and Huaxiong Wang:  
256 Bit Standardized Crypto for 650 GE – GOST Revisited,  
In CHES 2010

“Despite considerable  
cryptanalytic efforts  
spent in the past 20 years,  
GOST is still not broken.”

## 6.2. Structure of GOST

$$Enc_k = \mathcal{D} \circ \mathcal{S} \circ \mathcal{E} \circ \mathcal{E} \circ \mathcal{E}$$


## Self-Similar Key Schedule

Periodic Repetition + Inversed Order

rounds	1	8	9	16	17	24	25	32
keys	$k_0 k_1 k_2 k_3 k_4 k_5 k_6 k_7$		$k_0 k_1 k_2 k_3 k_4 k_5 k_6 k_7$		$k_0 k_1 k_2 k_3 k_4 k_5 k_6 k_7$		$k_7 k_6 k_5 k_4 k_3 k_2 k_1 k_0$	

Table 1. Key schedule in GOST

We write GOST as the following functional decomposition (to be read from right to left) which is the same as used at Indocrypt 2008 [29]:

$$Enc_k = \mathcal{D} \circ \mathcal{S} \circ \mathcal{E} \circ \mathcal{E} \circ \mathcal{E} \quad (1)$$

Where  $\mathcal{E}$  is exactly the first 8 rounds which exploits the whole 256-bit key,  $\mathcal{S}$  is a swap function which exchanges the left and right hand sides and does not depend on the key, and  $\mathcal{D}$  is the corresponding decryption function with  $\mathcal{E} \circ \mathcal{D} = \mathcal{D} \circ \mathcal{E} = Id$ .

## Last 16 Rounds of GOST

$$Enc_k = \boxed{D \circ S \circ \mathcal{E} \circ \mathcal{E} \circ \mathcal{E}}$$

←

“Theorem Which Won World War 2”,


[I. J. Good and Cipher A. Deavours, afterword to: Marian Rejewski, "How Polish Mathematicians Deciphered the Enigma", Annals of the History of Computing, 3 (3), July 1981, 229-232]

P and

$$Q^{-1} \circ P \circ Q$$

have the same cycle structure

## Last 16 Rounds of GOST

$$Enc_k = \boxed{D \circ S \circ \mathcal{E} \circ \mathcal{E} \circ \mathcal{E}}$$


“Theorem Which Won World War 2”,

⇒ Has **exactly**  $2^{32}$  fixed points (order 1)  
and  $2^{64} - 2^{32}$  points of order 2.

⇒ A lot of fixed points (very few for DES).

# Black Box Reductions

## Reflection Attack

## Reflection – Happens $2^{32}$ Times - KPA

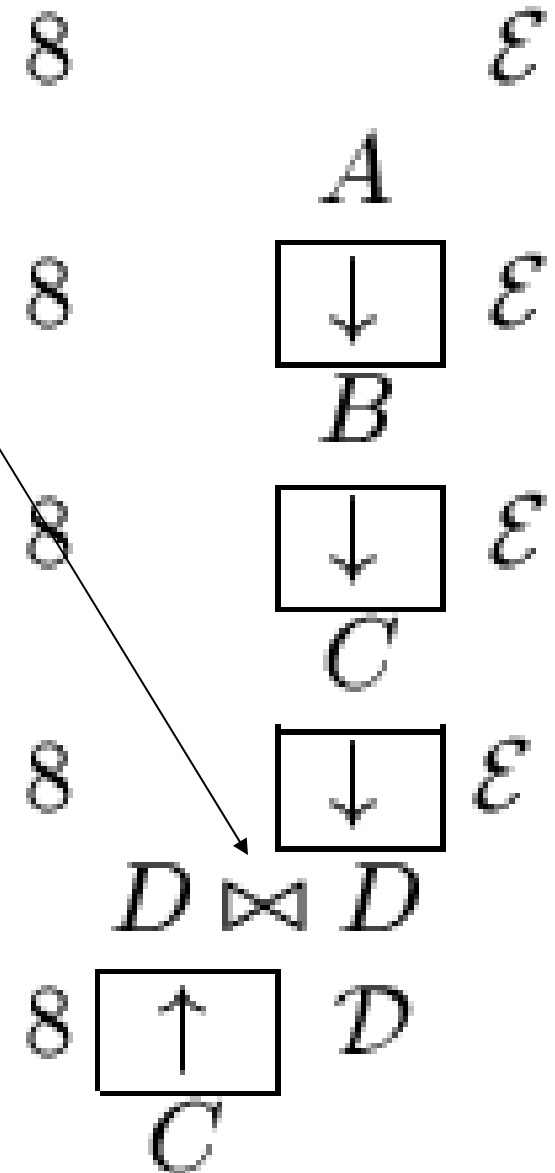
- guess A det C  
info=64 cost= $2^{-32}$
- guess B  
info=64+64 cost= $2^{-64}$
- [guess D  
info=64 cost= $2^{-32}$ ]

Summary: we get 2/3 KP for 8R for the price of  $2^{-96}/2^{-128}$ .

break 8R 2KP  $2^{127}$   
 $\Rightarrow$  break 32R D= $2^{32}$  T= $2^{223}$

break 8R 3KP  $2^{110}$   
 $\Rightarrow$  break 32R D= $2^{32}$  T= $2^{238}$

$\mathcal{E}^3(X_i)$  is symmetric



## 6.8. Double Reflection Attack



## 2x Reflection, Happens About Once:

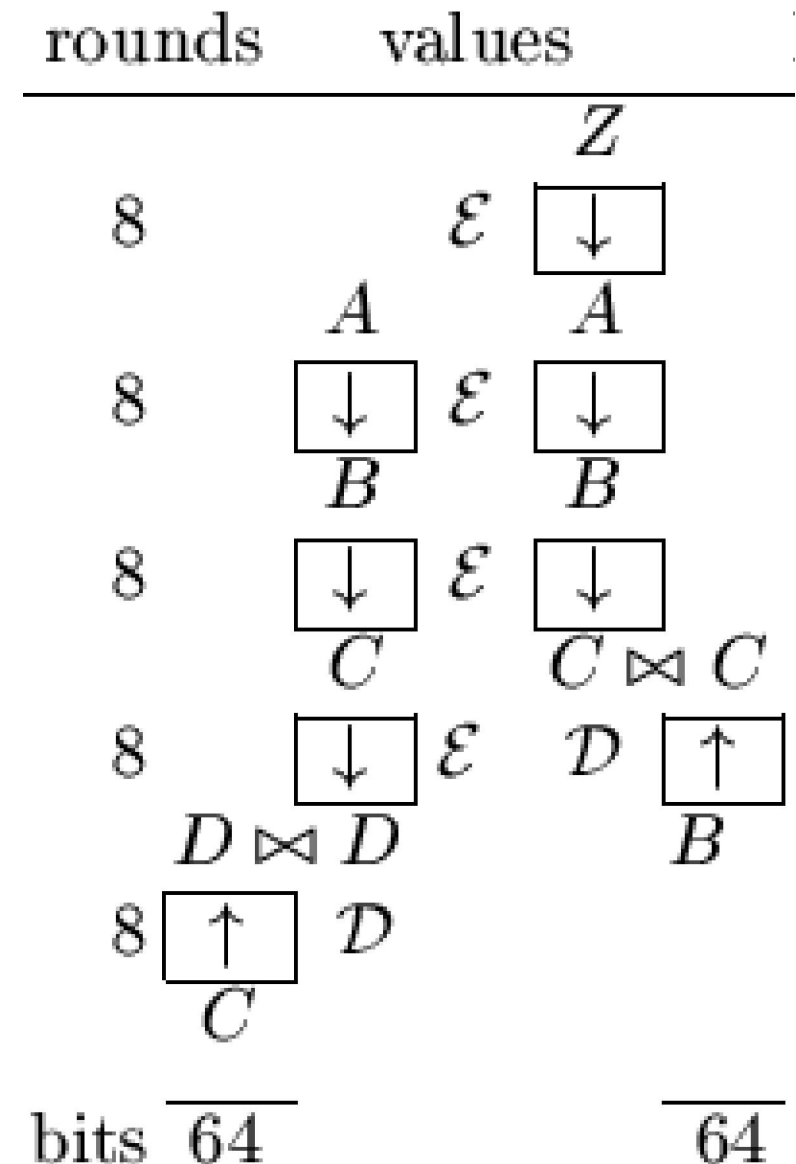
$$\begin{array}{l} \mathcal{E}^2(X_i) \text{ symmetric} \\ \mathcal{E}^3(X_i) \text{ symmetric} \end{array}$$

- guess C det A  
info=64 cost= $2^{-32}$
- guess B det Z  
info=64+64+64 cost= $2^{-64}$
- [guess D  
info=64 cost= $2^{-32}$ ]

Summary: we get 3/4 KP for  
8R for the price of  $2^{-96}/2^{-128}$

break 8R 3KP  $2^{110}$   
=> break 32R D= $2^{64}$  T= $2^{206}$

break 8R 4KP  $2^{94}$   
=> break 32R D= $2^{64}$  T= $2^{222}$

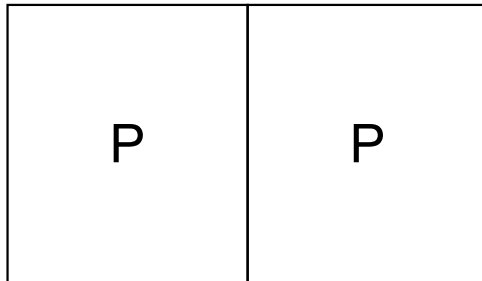


# Fixed Point Attack

(already seen for KeeLoq last step Attack 3)

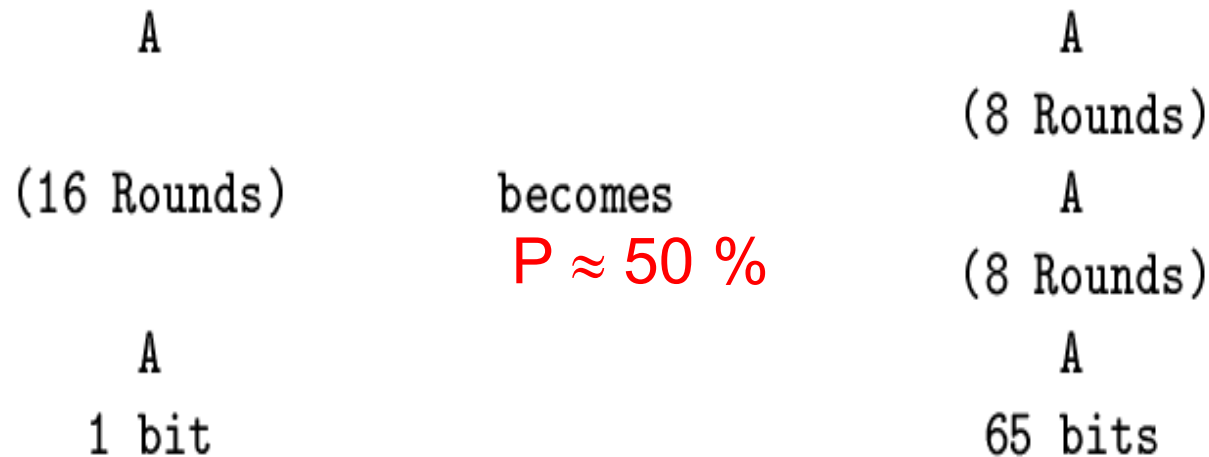
# First 16 Rounds of GOST

- Same perm, same key



## First 16 Rounds of GOST

1 point, first 16 rounds of GOST



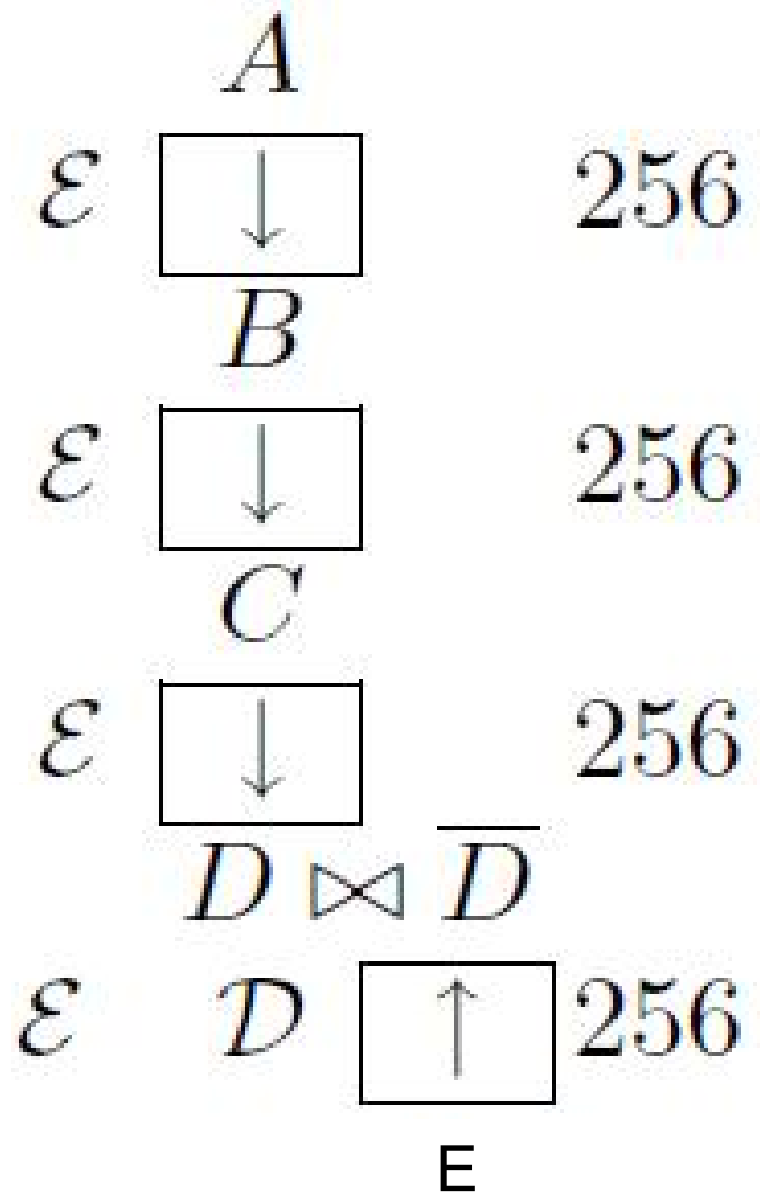
$A$  is an arbitrary unknown value

**Fig. 29.** Fixed points in the first 16 rounds of GOST seen as an Induction property: the value in the middle is obtained nearly for free instead of  $2^{-64}$

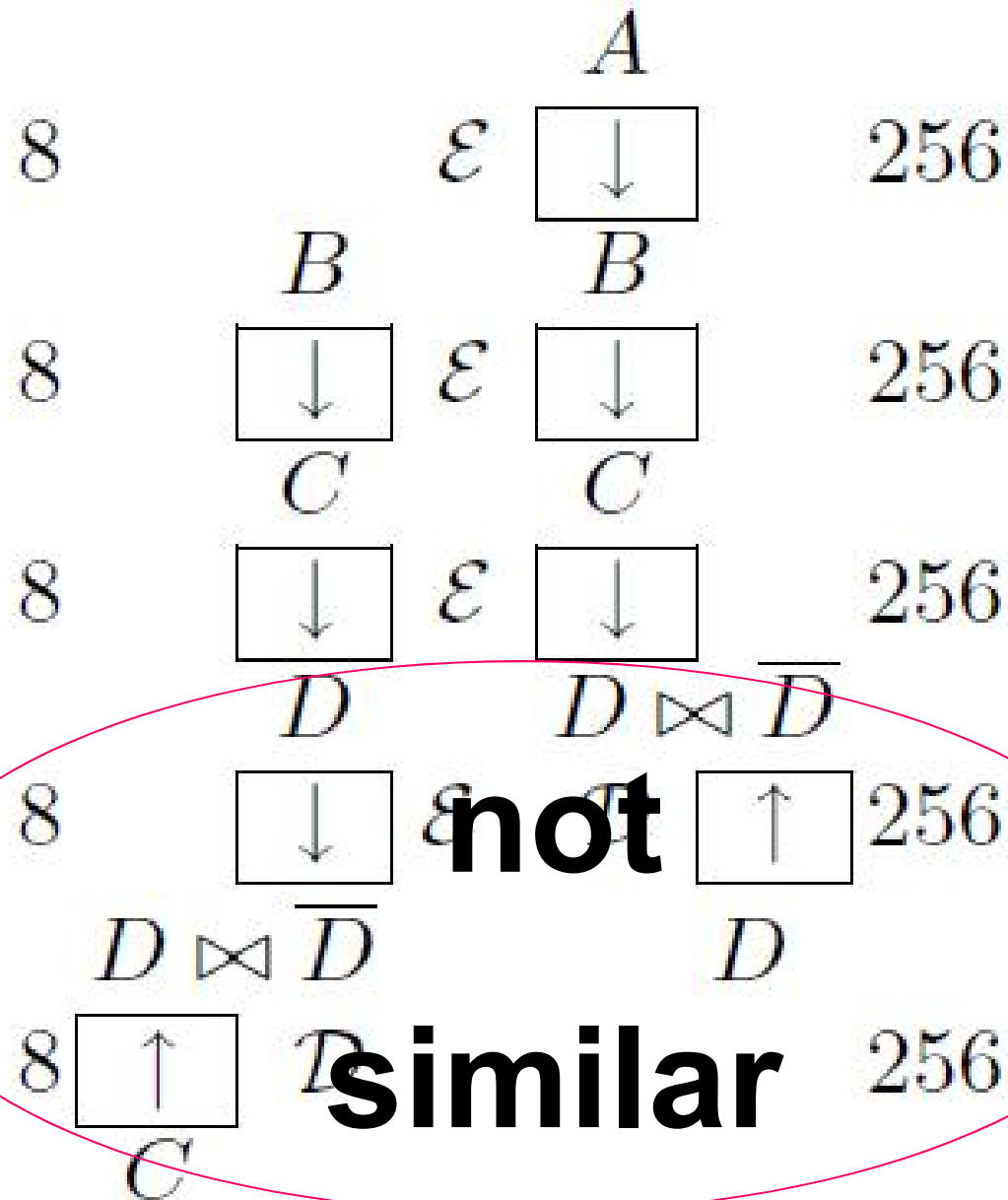
Breaking Full GOST  
Black Box Reduction:  
Pseudo-Sliding Attack  
[Cryptologia Jan 2012]

## One Encryption

$$Enc_k = D \circ S \circ \mathcal{E} \circ \mathcal{E} \circ \mathcal{E}$$



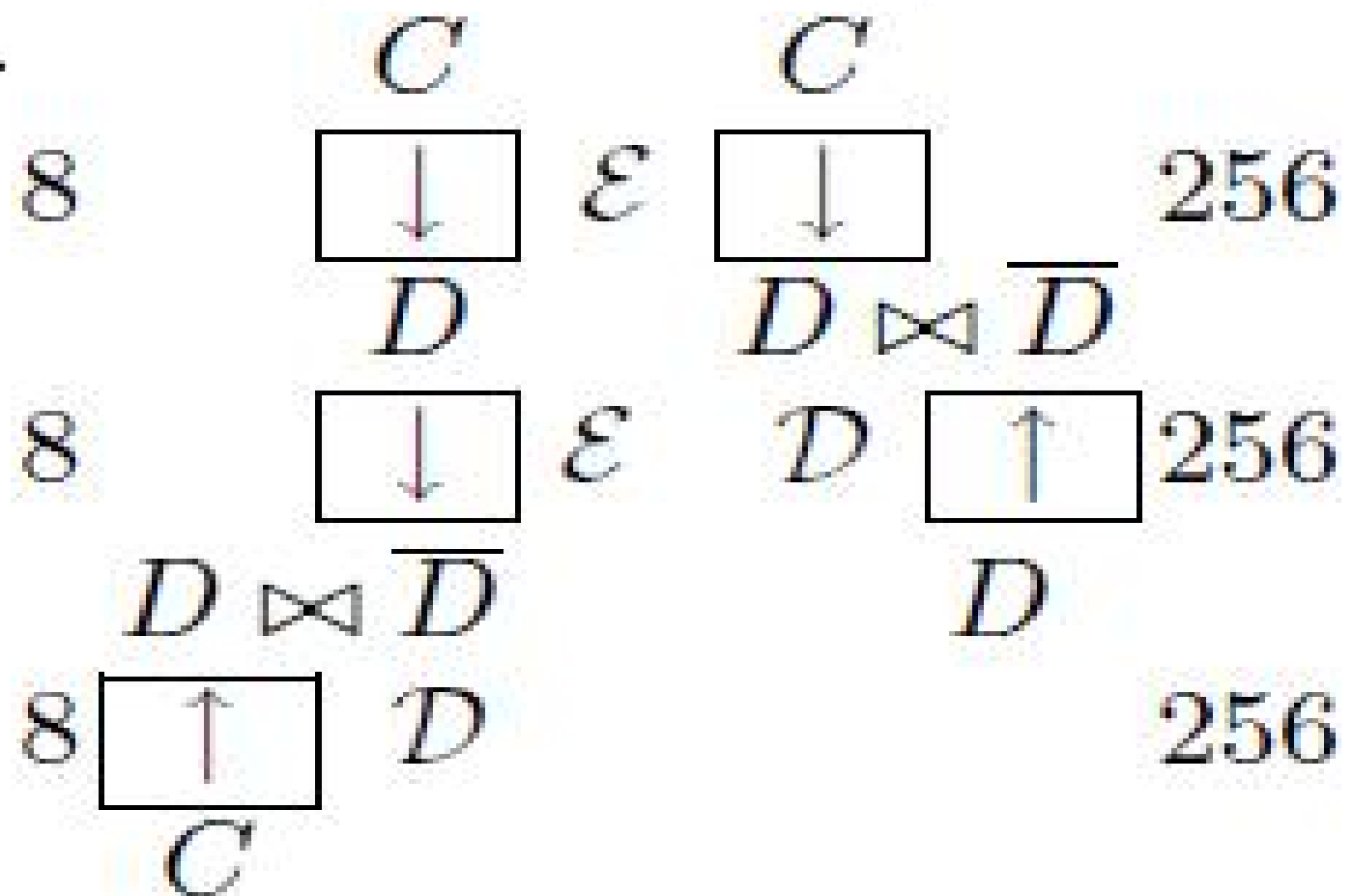
## Two Encryptions with A Slide



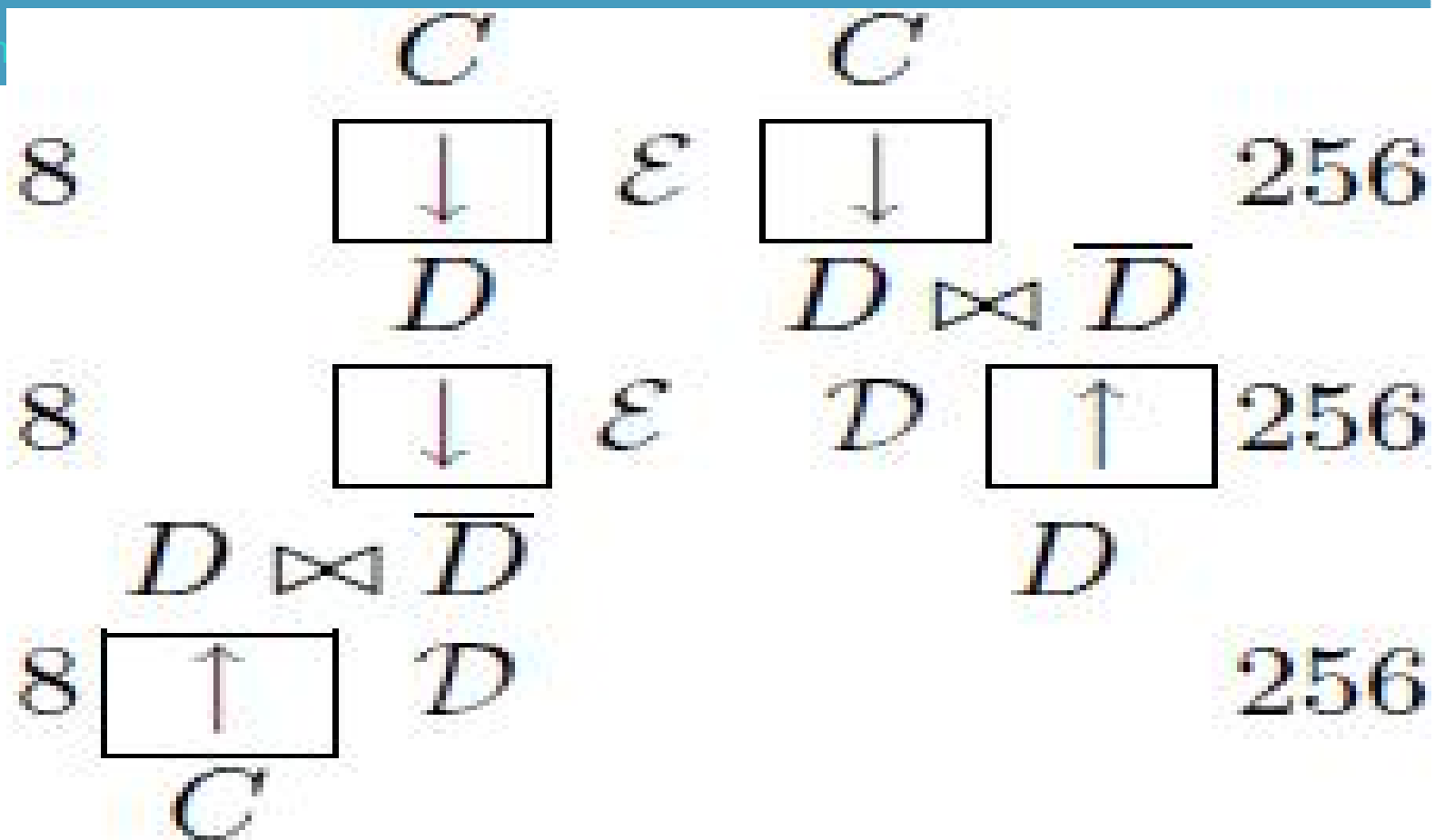
## Assumptions

We proceed as follows. We consider plaintexts with a very peculiar property:

**Assumption 1 (Assumption W).** Let  $A$  be such that  $\mathcal{E}(D) = \overline{D}$  where  $D$  is defined as  $D = \mathcal{E}^3(A)$ .







**Fact 2 (Property W).** Given  $2^{64}$  KP there is on average one value  $A$  which satisfies the Assumption. For 63% of all GOST keys at least one such  $A$  exists.

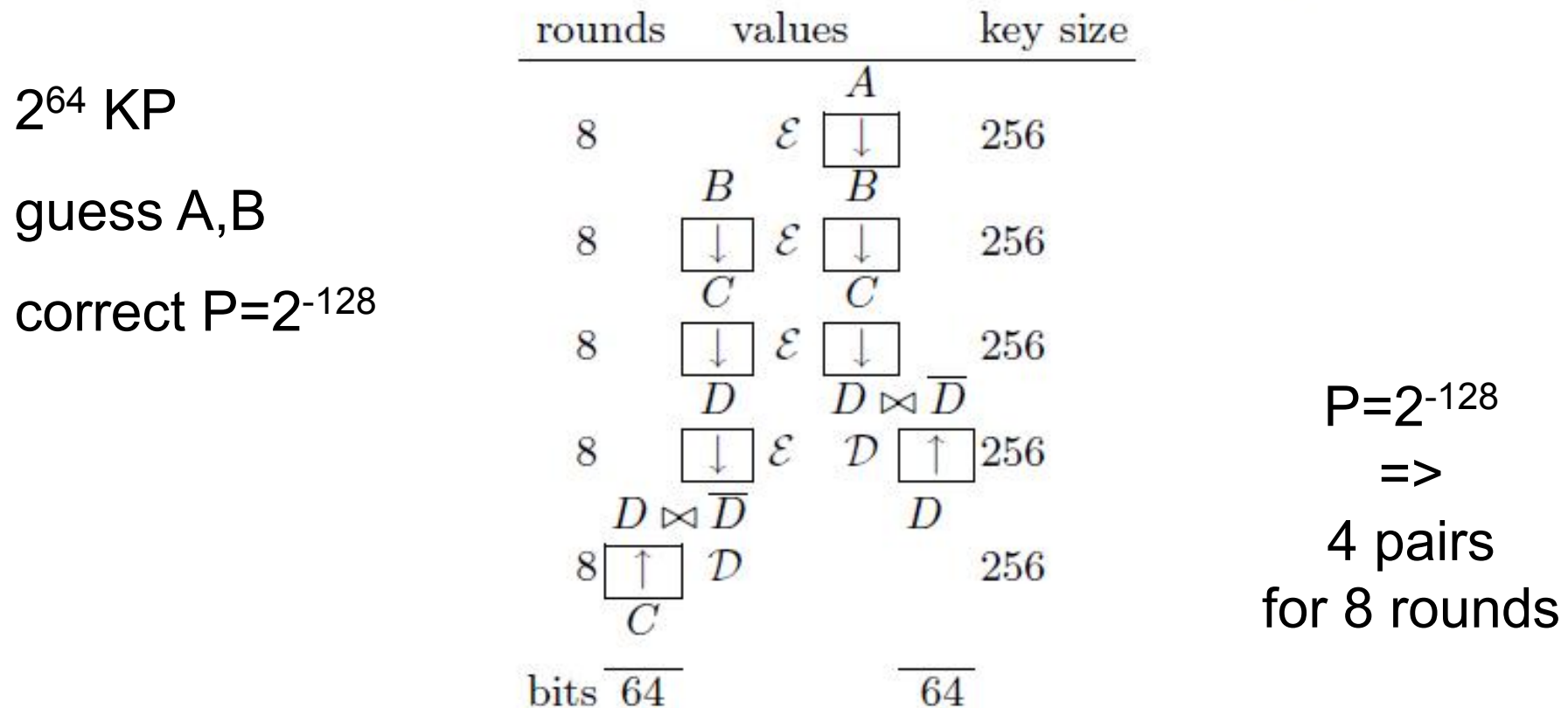
*Remark:* For the remaining 37 % of keys this attack fails. However many other attacks still work, see [12].

# Reduction

## New Attack on GOST

**Fact 3 (Consequences of Property W).** If  $A$  satisfies the Assumption W above and defining  $B = \mathcal{E}(A)$  and  $C = \mathcal{E}(B)$  we have:

1.  $Enc_k(A) = D$ . This is illustrated on the right hand side of Fig. 1.
2.  $Enc_k(B) = C$  This can be seen on the left hand side of Fig. 1.



**Fig. 1.** A black-box “Algebraic Complexity Reduction” from 32 to 8 rounds of GOST

## 6.5. Can We Solve 8R?

## Final Key Recovery 8R

4 Pairs, 8 rounds.

The key is found within

$2^{94}$  GOST computations.

## Overall Attack

$2^{128+94}$  GOST computations.

$2^{33}$  times faster than brute force.

Not the best attack yet.

## Other Attacks on GOST

Best single key attack (for any key):

$$D=2^{64} \quad T=2^{179}$$

Nicolas Courtois: [An Improved Differential Attack on Full GOST](#),  
in ``The New Codebreakers — a Festschrift for David Kahn'', LNCS 9100, Springer, 2015.  
long extend version: [eprint.iacr.org/2012/138](http://eprint.iacr.org/2012/138).