

# 暗号(2)

## —共通鍵暗号—

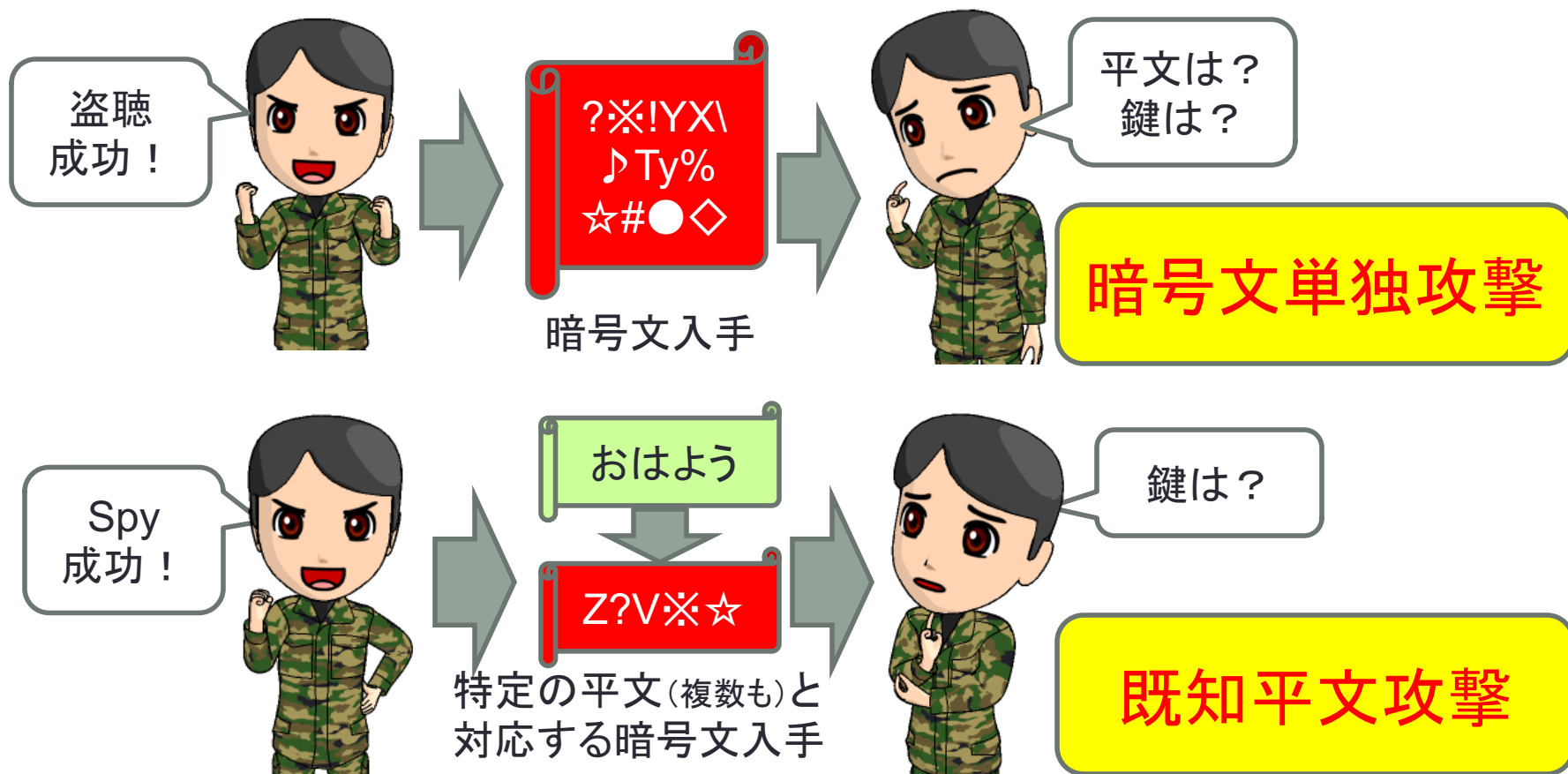
---

野口 拓

Taku NOGUCHI

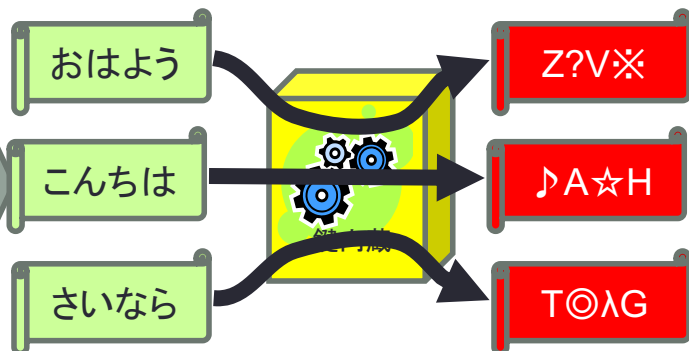
# 暗号化手法への「攻撃」

- Rule: 「暗号化手法」はわかっている  
「鍵」はわからない→鍵を求めるのが「攻撃」



# 暗号化手法への攻撃

暗号器  
入手!



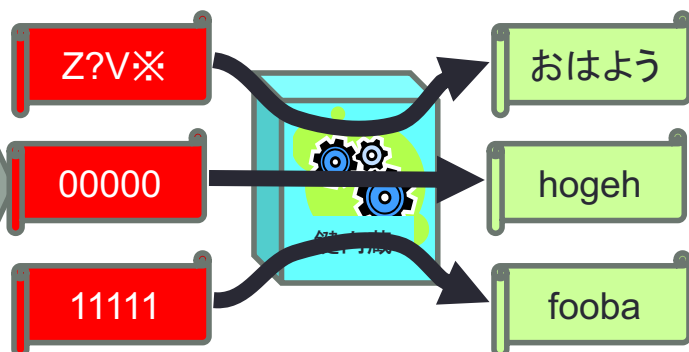
任意の平文に対し暗号文が得られる

鍵は?



選択平文  
攻撃

復号器  
入手!



任意の暗号文に対し平文が得られる

鍵は?

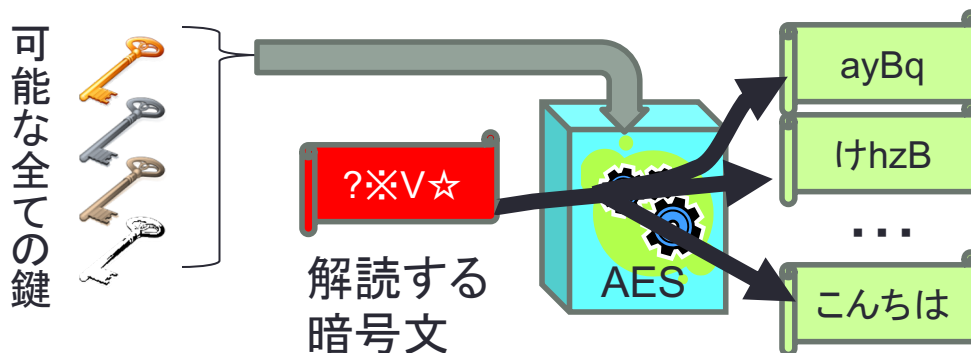


選択暗号文  
攻撃

現代暗号はこれら全ての攻撃に耐えたものが使われる

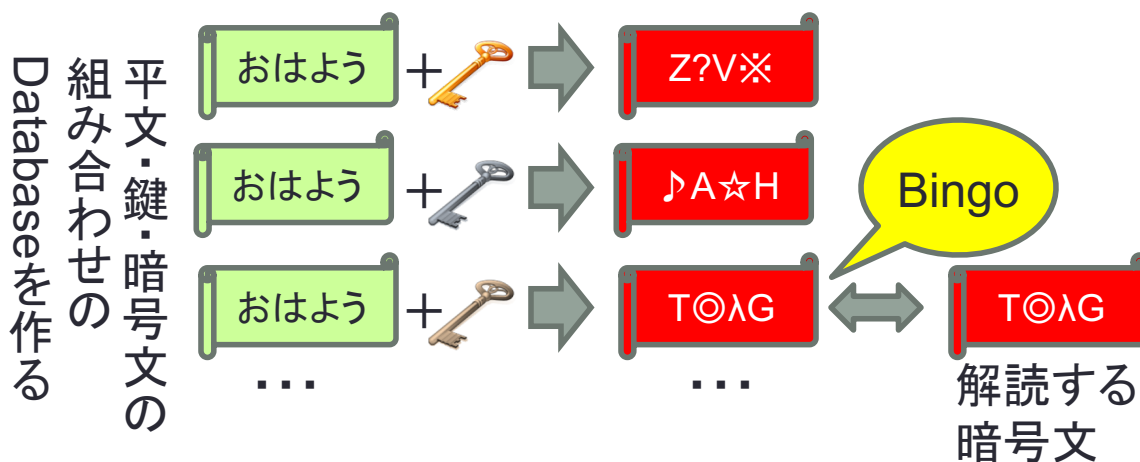
# 攻撃手法いろいろ

- 総当たり攻撃(Brute Force Attack)



鍵長  $n$  bits  
→  $2^n$  の試行

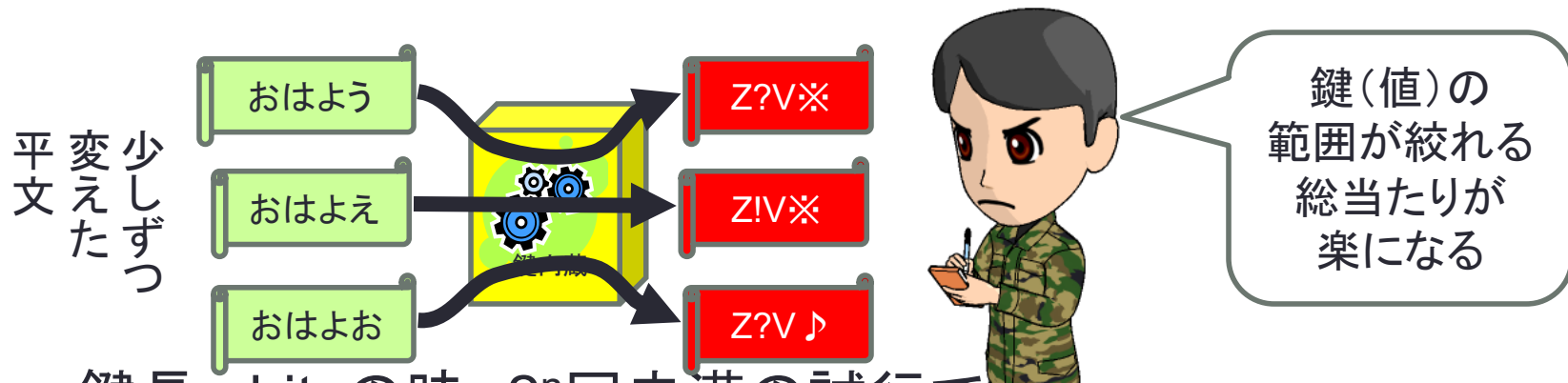
- 辞書攻撃(Dictionary Attack)



総当たりと同じだが  
Databaseを使えば  
速度が上がる  
(その代わり記憶容量が)  
**Time-space tradeoff**

# もう少し「賢い」方法

- 差分解読法(選択平文攻撃の一つ)



- 鍵長 $n$  bitsの時、 $2^n$ 回未満の試行で鍵が特定できる手法が見いだされればその暗号化手法は「解読された」という

# 暗号の安全性

シャノン : Shannon

- 情報理論的安全性

- (情報理論の父と言われる)シャノンにより証明(1949年)
- 何もない状態で平文を(あてずっぽうで)推測するのと、暗号文を見てから推測するのとで推測が当たる可能性が変わらない場合を情報理論的に安全であるという
- これを実現するには  
平文のビット長以上の鍵長必要であることが証明されている
  - つまり通信として意味がない！

- 計算量的安全性

- 鍵長が $k$ ビットである場合、 $2^k$ 個の鍵を総当りすれば必ず解読できるので、 $k$ を大きく取って「実用上」解読を不可能にする
- $2^k$ 未満の回数で解読が可能な攻撃が見つければ「解読」

# 「解読」≠「使い物にならない」

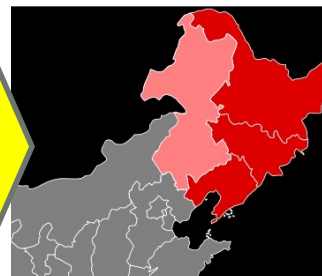
- 現代暗号の世界での鍵の解読は  
「どこかにおちている金貨を探す」ようなもの



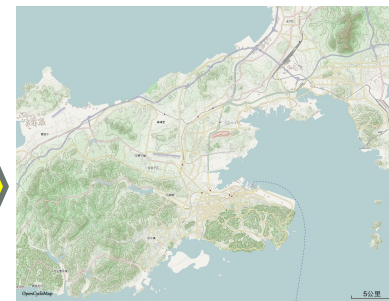
X  
年  
後



Y  
年  
後



Z  
年  
後



解読される前  
=「世界のどこか」

ある解析手法により  
=「北半球のどこか」

別の手法により  
=「中国東北部」

さらに別の手法により  
=「大連市」

計算量的  
に安全

解読された  
がまだ持つ

そろそろ危険  
移行を準備

危険!  
使用しない

本当に使い物にならなくなるまでには時間的猶予がある場合が多い

# 信頼できる暗号化手法の評価: NIST / CRYPTREC

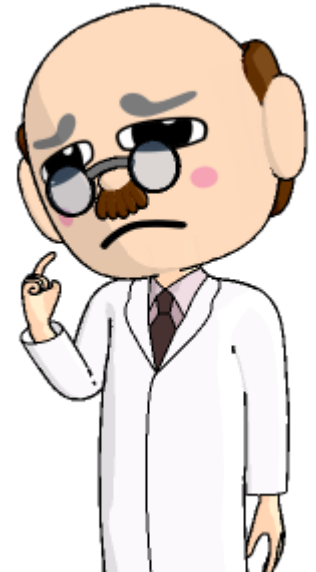
アメリカ : United States of America

- アメリカ(美国)標準技術局(NIST)
  - 暗号化手法等の安全性を定期的に評価:公表
  - アメリカ政府で使用する暗号化手法を決定  
事実上の世界標準
  - FIPS (連邦情報処理標準規格)を発行
- CRYPTREC <https://www.cryptrec.go.jp/>
  - IPA+NICTと経産省+総務省の合同Project
  - 日本において暗号化手法を独自に評価
  - 「電子政府推奨暗号」を決定



# 暗号が「解読」されるとき

- 計算機が十分速くなったとき
  - 時代と共に避けられないこと
  - 鍵長を十分長くして対処
- (特定の暗号化手法に対して)  
攻撃法が発見されたとき
  - 新しい解読手法の発見
    - ただし「暗号化手法」が公表されている限りは世界中で研究が行われるのでその動向を見ていれば「解読されそう」になれば気配がわかる
  - 新しい解読デバイスの開発？
    - 量子コンピュータとか??



# 「暗号解読の相場観」

- もし「解読法」が見つからなければ？
  - 鍵がkビットなら総当たりは $2^k$ 必要
    - $k=32$ で約40億  $64$ で $1.8 \times 10^{19}$   $128$ で340週間 $=3.4 \times 10^{38}$
  - 現在の世界最高速スパコンが400Pflops級  
つまり1秒間に40京回( $4 \times 10^{17}$ )程度浮動小数点演算
  - 1年はせいぜい $3 \times 10^8$ 秒くらいなので  
世界最高速スパコンを使うと  
64ビット近辺は危ない(1分未満...)が128ビットになると  
ざっと $10^{13}$ 年かかる計算
  - ただし計算機は「毎年倍速になる」ので注意
- 画期的な方法が出ると急激に危なくなる
  - 新しい解読法、新しい計算機(量子計算機など)  
...これらは解読にかかる時間をケタ違いに短縮することが

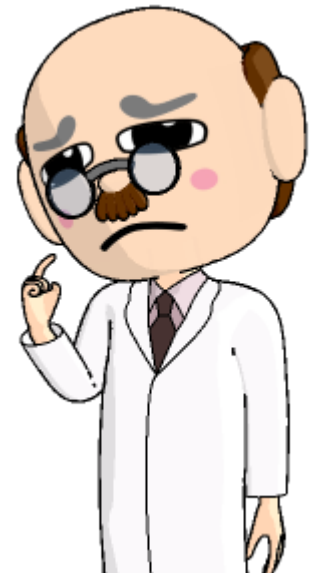
# 2022年6月のTOP500

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	<b>Frontier</b> - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65	21,100
2	<b>Supercomputer Fugaku</b> - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
3	<b>LUMI</b> - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	1,110,144	151.90	214.35	2,942
4	<b>Summit</b> - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148.60	200.79	10,096
5	<b>Sierra</b> - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94.64	125.71	7,438
6	<b>Sunway TaihuLight</b> - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93.01	125.44	15,371

1102PFlop/sくらい

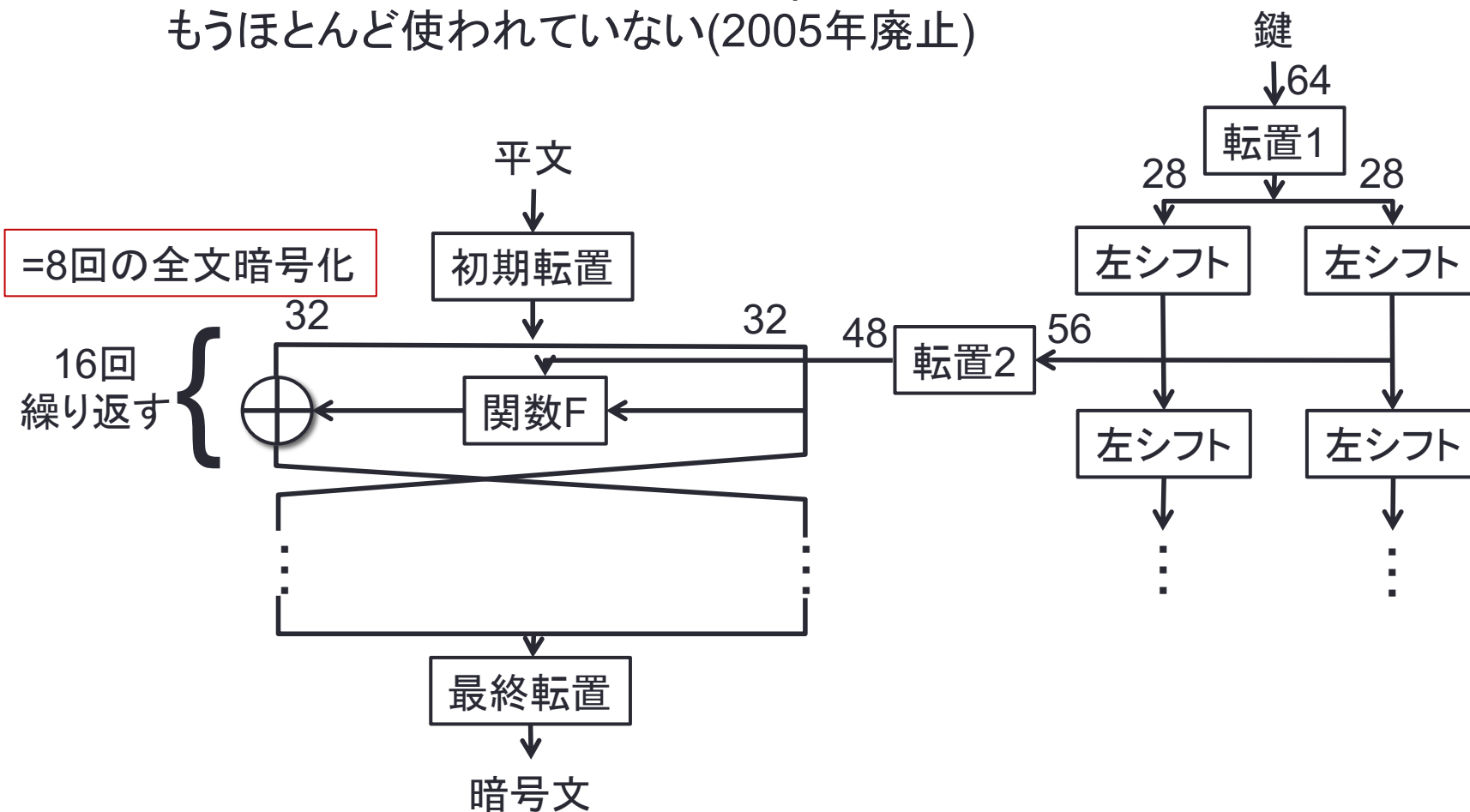
# 暗号世界の「都市伝説」

- 某社  
「アメリカ政府の決めた暗号には  
『Backdoor』があるかも知れません！」  
「自社開発の独自のAlgorithmの採用により、絶対に安心な暗  
号化手法を！」
- 世界中の暗号学者がさまざまな  
攻撃を試して耐え抜いている手法と  
Algorithmが「秘密」  
＝評価を受けていない手法の  
どっちが安全？

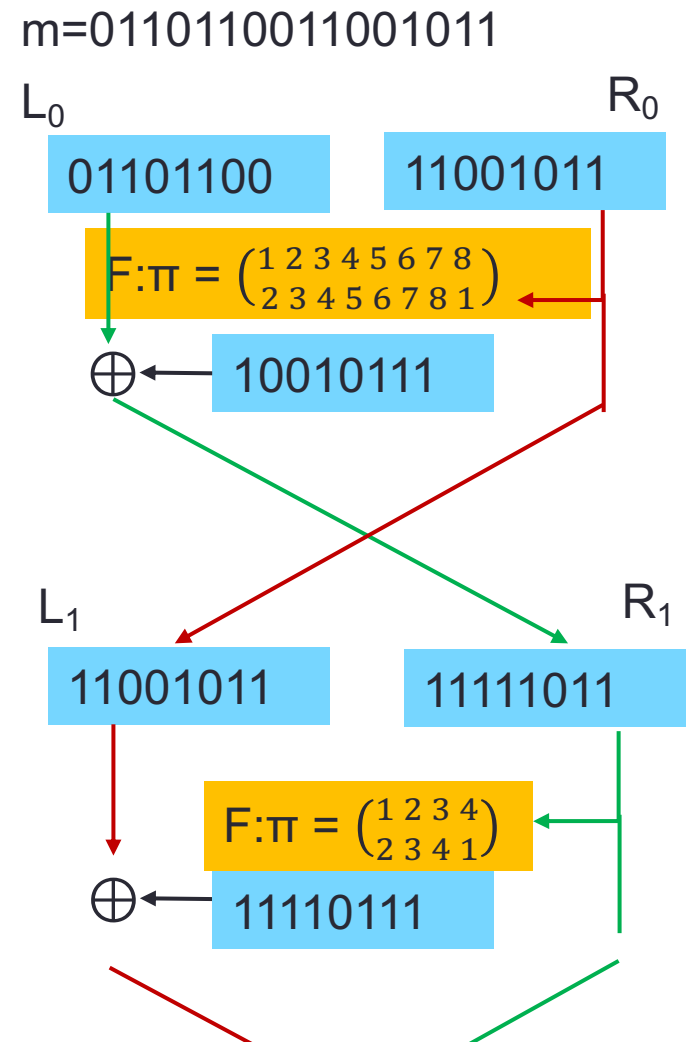
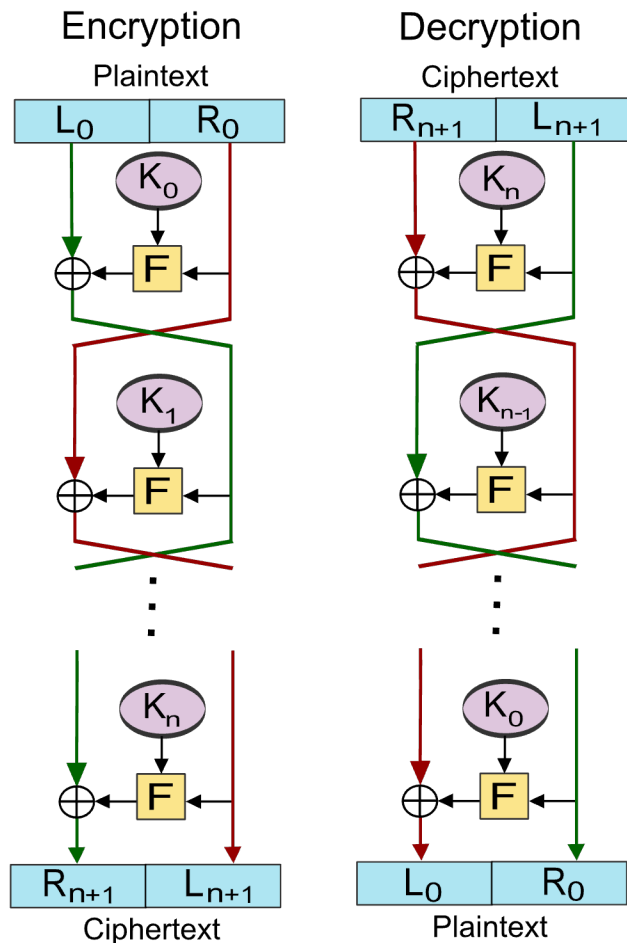


# DES (Data Encryption Standard)

- 米国政府標準で定められていたblock暗号(1977年制定)
  - 64bit(8bytes)単位でデータを暗号化するブロック暗号アルゴリズム
  - 3種の鍵を用いて3回繰り返すTriple-DESが使われるようになったがもうほとんど使われていない(2005年廃止)

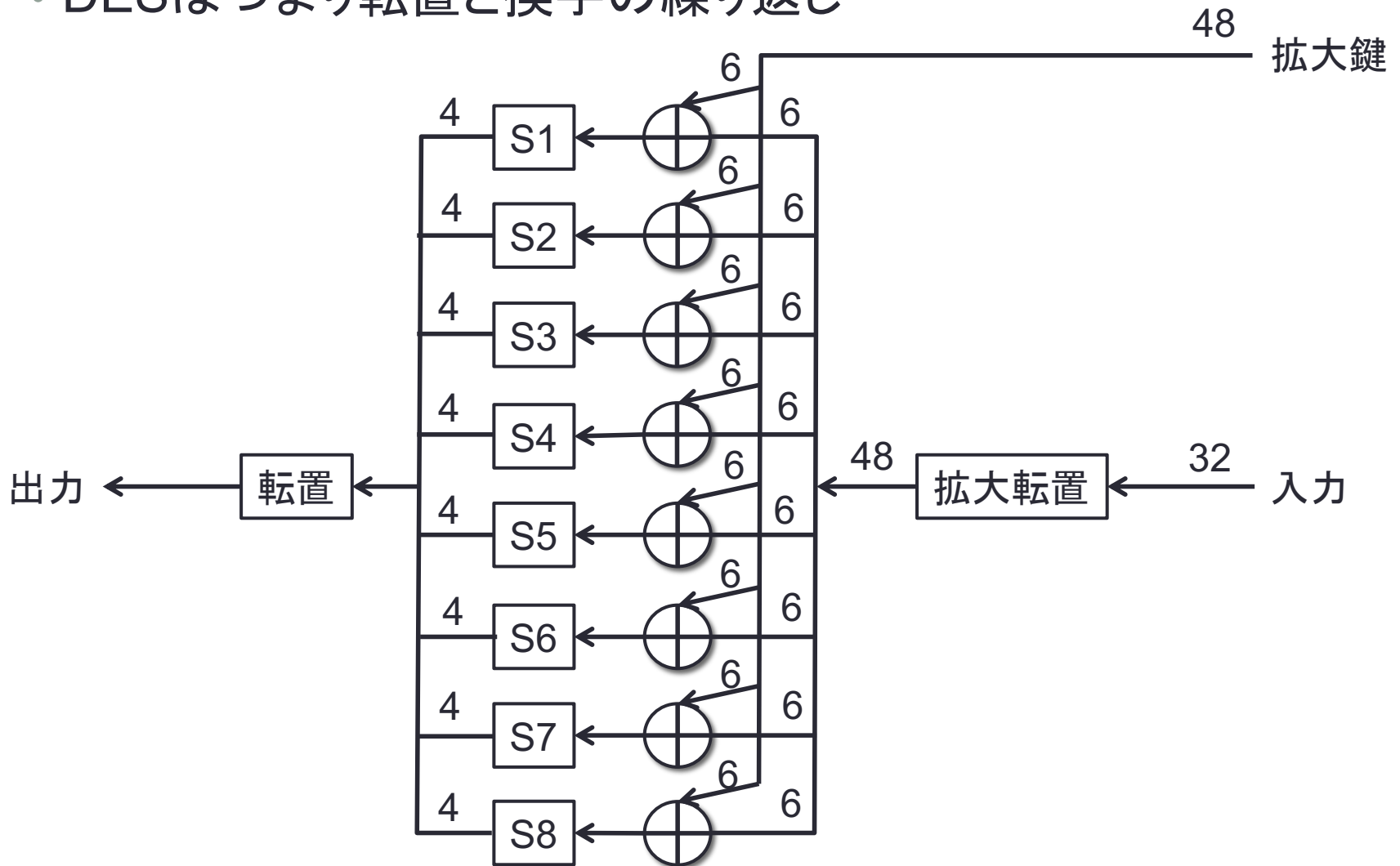


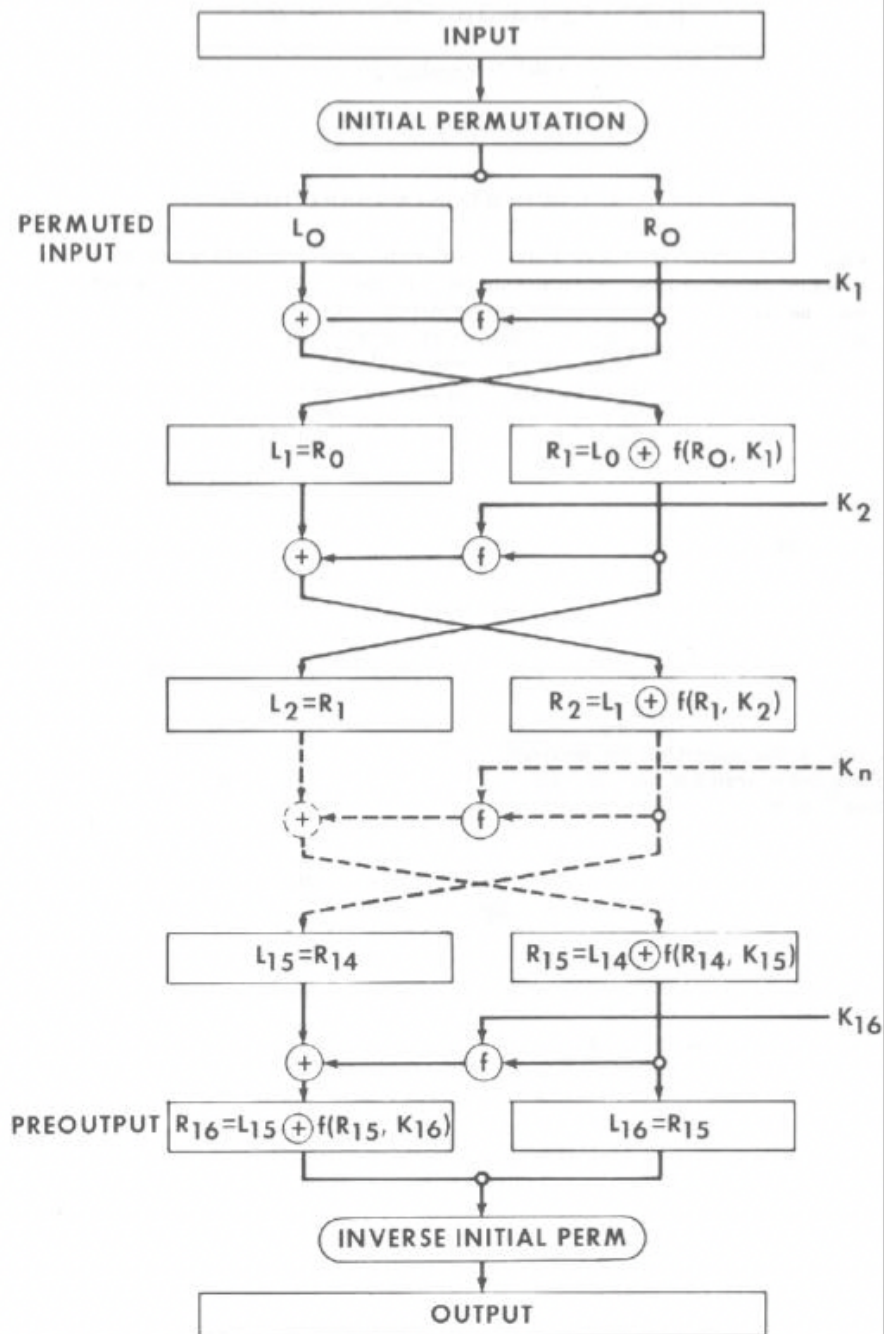
# Feistel 構造 (ファイステル構造)



# 非線形関数F

- DESはつまり転置と換字の繰り返し







# ちょっと計算してみる？できるかな？

平文  $p = 0123456789ABCDEF$

0	0	0	0	0	0	0	1
0	0	1	0	0	0	1	1
0	1	0	0	0	1	0	1
0	1	1	0	0	1	1	1
1	0	0	0	1	0	0	1
1	0	1	0	1	0	1	1
1	1	0	0	1	1	0	1
1	1	1	0	1	1	1	1

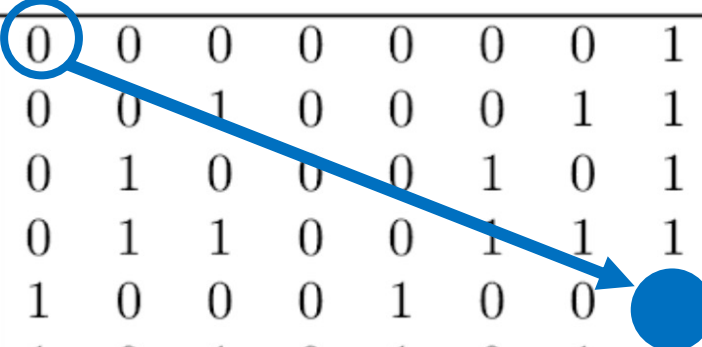
暗号鍵  $k = 133457799BBCDFF1$

0	0	0	1	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	1	0	1	1	1
0	1	1	1	1	0	0	1
1	0	0	1	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	1	1	1	1	1
1	1	1	1	0	0	0	1

# 初期転置 (Initial Permutation)

- 順番を入れ替える
- L0, R0を得る

平文  $p = 0123456789ABCDEF$



0	0	0	0	0	0	0	1
0	0	1	0	0	0	1	1
0	1	0	0	0	1	0	1
0	1	1	0	0	1	1	1
1	0	0	0	1	0	0	1
1	0	1	0	1	0	1	1
1	1	0	0	1	1	0	1
1	1	1	0	1	1	1	1

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

# 初期転置後

1	1	0	0	1	1	0	0		L0
0	0	0	0	0	0	0	0		R0
1	1	0	0	1	1	0	0		
1	1	1	1	1	1	1	1		
1	1	1	1	0	0	0	0		
1	0	1	0	1	0	1	0		
1	1	1	1	0	0	0	0		
1	0	1	0	1	0	1	0		

# K<sub>1</sub>を求める

- kからC<sub>0</sub>, D<sub>0</sub>をつくって
- それを左1ビット循環シフトか左2ビット循環シフトして
- PC2で統合する

暗号鍵 k = 133457799BBCDFF1

0	0	0	1	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	1	0	1	1	1
0	1	1	1	1	0	0	1
1	0	0	1	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	1	1	1	1	1
1	1	1	1	0	0	0	1

64bit

C <sub>0</sub>						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
D <sub>0</sub>						
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	13	28	20	12	4

PC1

56bit

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

PC2

48bit

PC2 =

K1	0	0	0	1	1	0
	1	1	0	0	0	0
	0	0	1	0	1	1
	1	0	1	1	1	1
	1	1	1	1	1	1
	0	0	0	1	1	1
	0	0	0	0	0	1
	1	1	0	0	1	0

48bit

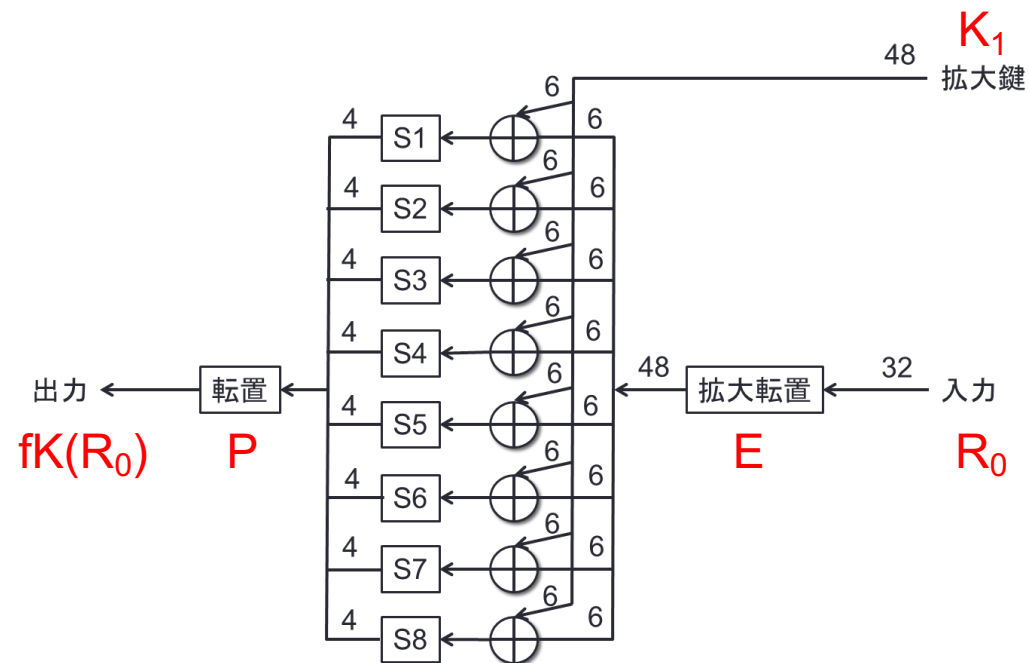
# $E(R_0)$ を求める

$E(R_0)$		0	1	1	1	1	0
		1	0	0	0	0	1
		0	1	0	1	0	1
		0	1	0	1	0	1
		0	1	1	1	1	0
		1	0	0	0	0	1
		0	1	0	1	0	1
		0	1	0	1	0	1

E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

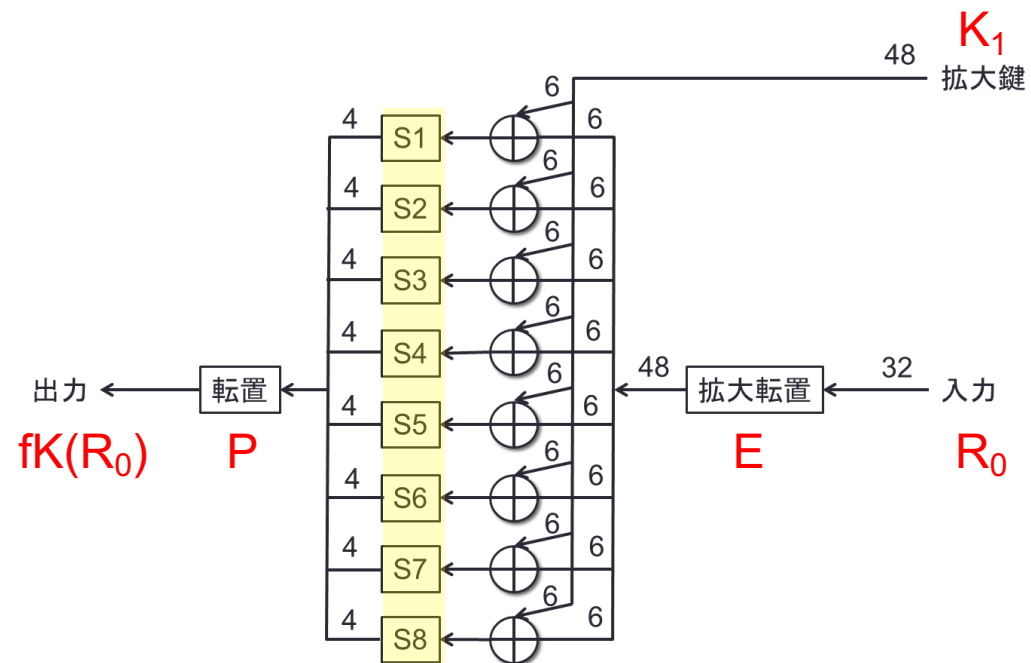
# $E(R_0) \oplus K_1$ を求める

E(R0) XOR K1							
		0	1	1	0	0	0
		0	1	0	0	0	1
		0	1	1	1	1	0
		1	1	1	0	1	0
		1	0	0	0	0	1
		1	0	0	1	1	0
		0	1	0	1	0	0
		1	0	0	1	1	1



# $S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8$ を求める

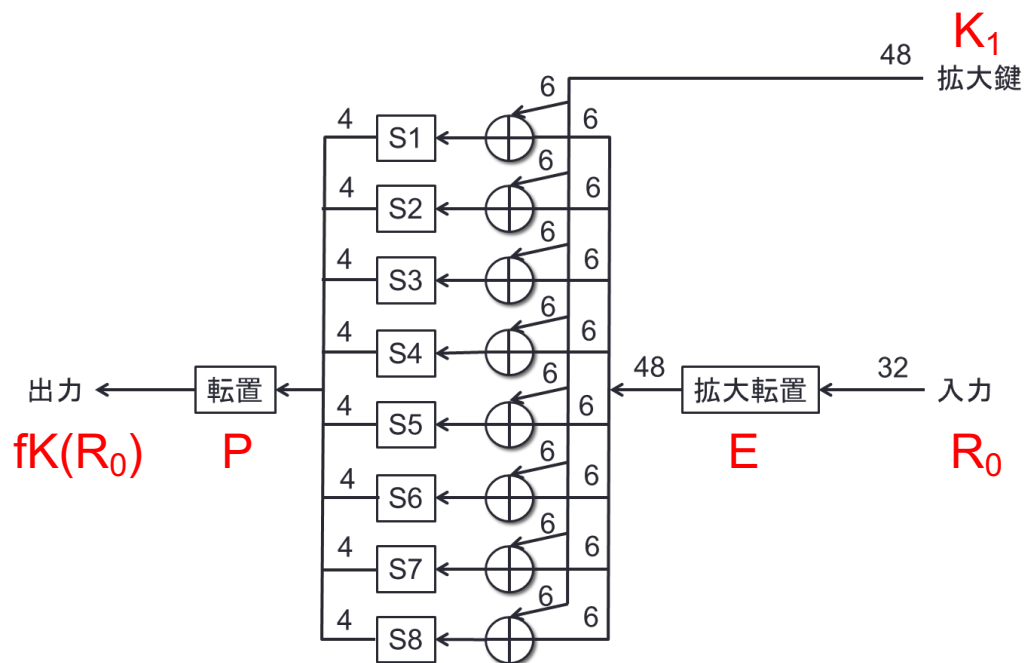
s1	0	1	0	1
s2	1	1	0	0
s3	1	0	0	0
s4	0	0	1	0
s5	1	0	1	1
s6	0	1	0	1
s7	1	0	0	1
s8	0	1	1	1





# fK(R<sub>0</sub>)を求める

s1	0	1	0	1
s2	1	1	0	0
s3	1	0	0	0
s4	0	0	1	0
s5	1	0	1	1
s6	0	1	0	1
s7	1	0	0	1
s8	0	1	1	1



fK(R<sub>0</sub>)

0	0	1	0	0	0	1	1
0	1	0	0	1	0	1	0
1	0	1	0	1	0	0	1
1	0	1	1	1	0	1	1

P

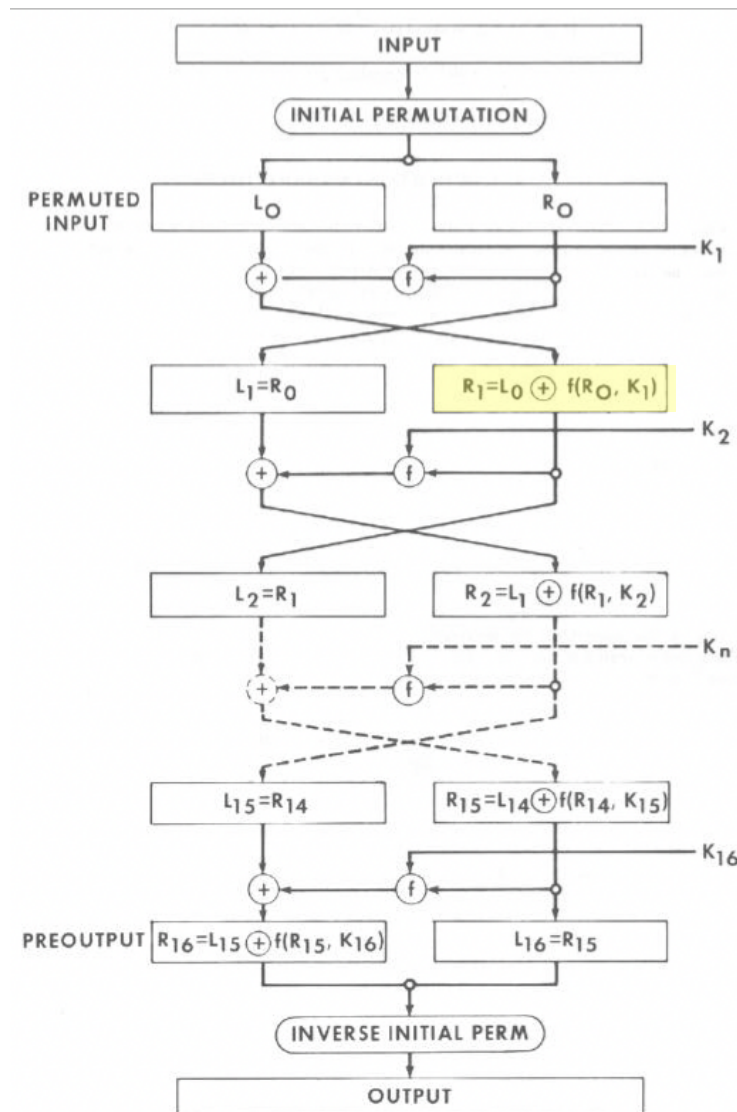
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

# R1を求める

- $fK(R_0) \oplus L_0$ で求められる

$$R1 = fK(R0) \text{ XOR } L0$$

1	1	1	0	1	1	1	1
0	1	0	0	1	0	1	0
0	1	1	0	0	1	0	1
0	1	0	0	0	1	0	0



ということを16回繰り返す

# DESは解読が進み使われなくなってきた

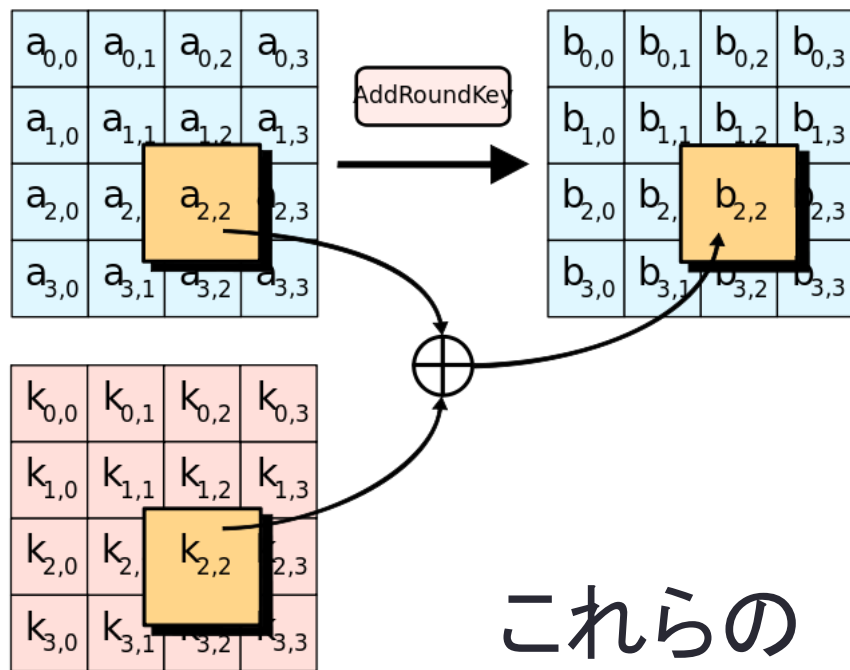
- 差分解読法（選択平文攻撃） Biham, Shamir
  - 多くのDES類似ブロック暗号に有効だがDESには適用しにくい（DESは最初から対策されていた）
    - $2^{47}$ 乗の検索が必要
- 線形解読法（既知平文攻撃） 松井充
  - 暗号アルゴリズムを線形式で近似して鍵を予想
  - 世界で初めてDESの実用的な解読に成功
    - $2^{45}$ 乗の検索で約99%  $2^{43}$ 乗でも約10%くらいで解読
    - 1994年 12台の計算機で2ヶ月半で解読することに成功
    - 1990年代終わりには1日以下で解読可能に

# DESの後継問題

- 3DES: DESの改良
  - DESを3回行う 鍵は2または3つ使う(計108/164ビット)
    - 2つの時は鍵1で暗号化、鍵2で復号化、鍵1で暗号化  
→こうすると鍵1=鍵2のとき単なるDESになる
- 米国政府 AES(Advanced Encryption Standard)を全世界に公募(1997年)
  - 最終的に2000年ベルギーのRijmenらによる Rijndaelを基にAESが成立
  - $n=128$   $k=128/192/256$
  - 128ビットを4x4バイトの行列とみなして、各バイトの置換と複雑な要素の入れ替えを繰り返す
  - 差分解読法はもちろん線形解読法にも強い

ベルギー : Kingdom of Belgium

# AESの基本構造



これらの  
繰り返し

