128ビット共通鍵ブロック暗号のASIC実装のポイント（KASUMI付）

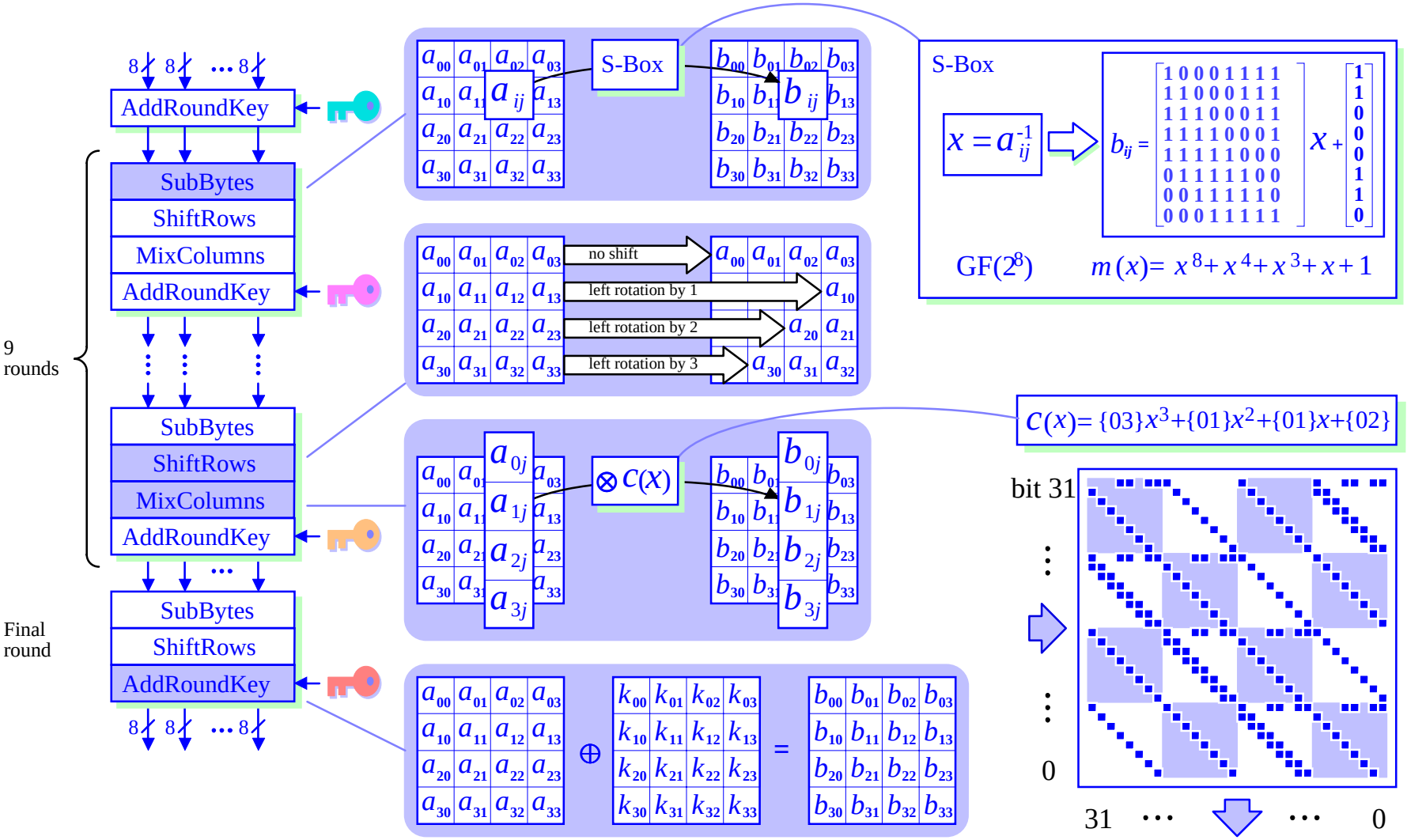
日本アイ・ビー・エム株式会社
東京基礎研究所
佐藤証

目次

- AES
- Camellia
- SC2000
- Hierocrypt－3
- MISTY＆KASUMI
- まとめ

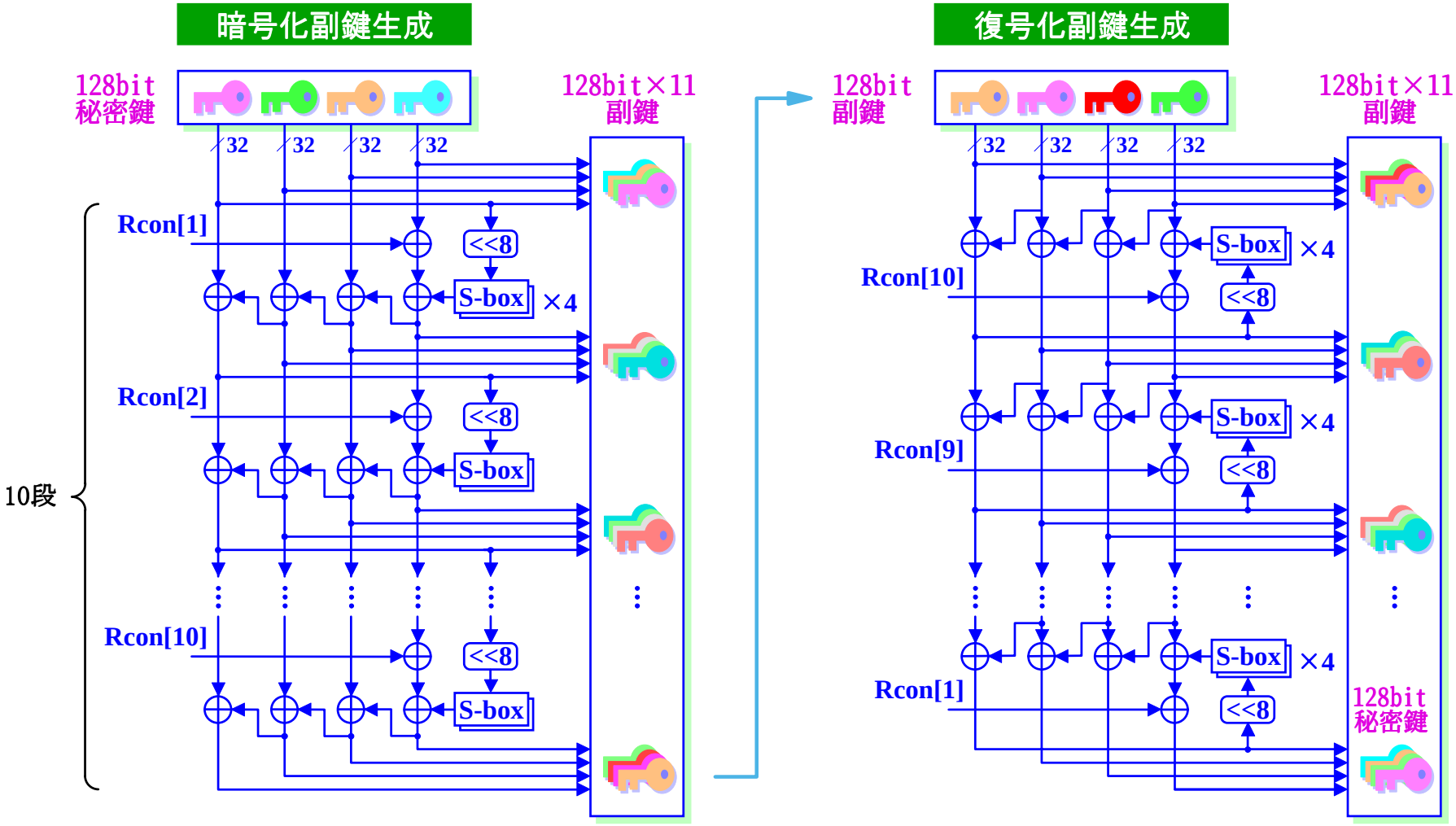
AES

- 暗号化アルゴリズム
  - 多くのコンポーネントがXORアレイで記述できるシンプルな構成
  - S-BoxはGF(2⁸)の乗法逆元演算を利用



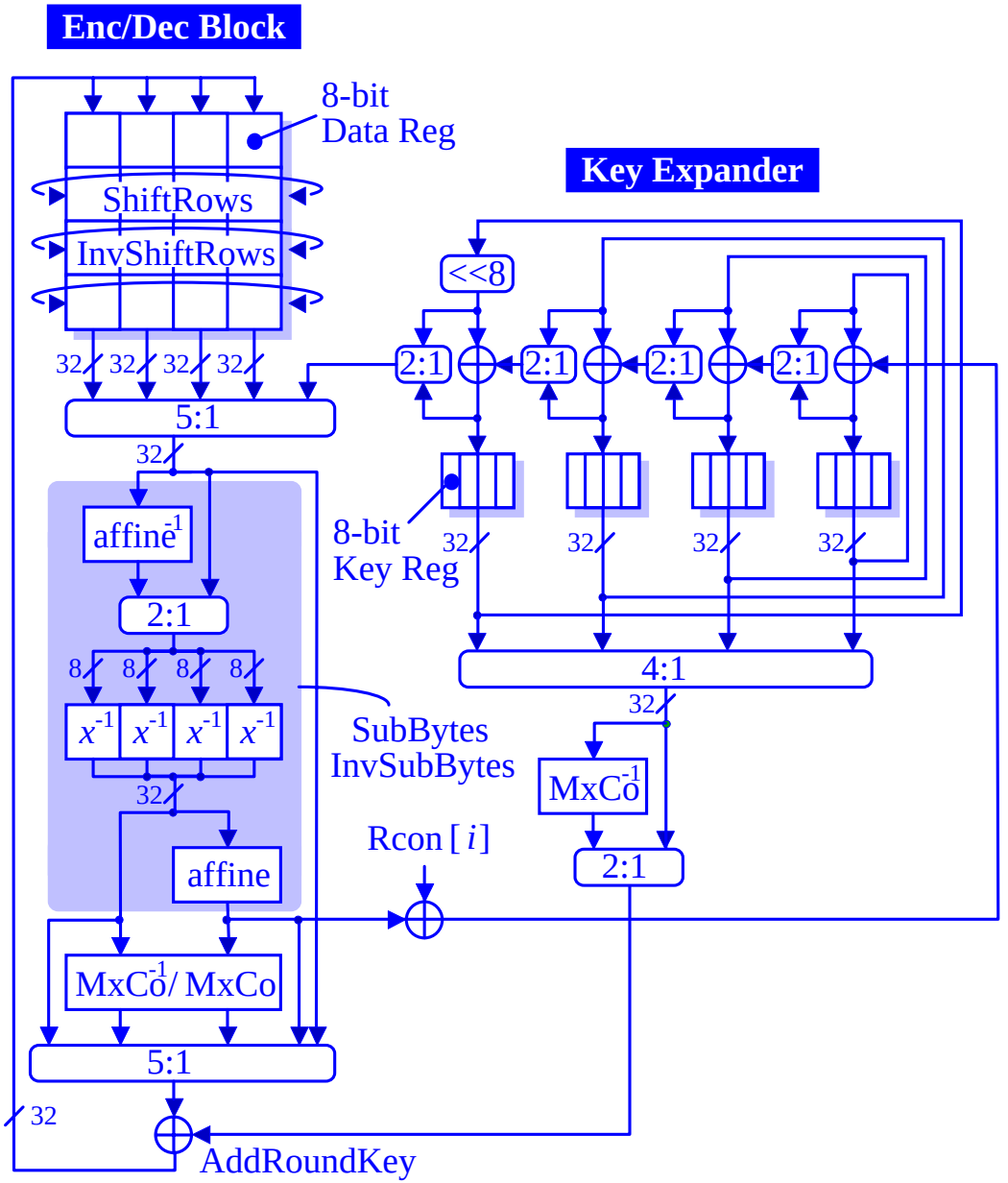
AES

- 鍵スケジューリング
  - 一段の128ビット副鍵生成に暗号化の8ビットS-BOXを4つ使用
  - 暗号化と復号化の副鍵は同じものを逆順に使用するので，On-the-Flyで復号化の副鍵を生成するには暗号化の最終段の副鍵を一旦生成



AES

- ループ処理のデータパス アーキテクチャ
  - 暗号化/復号化/鍵スケジューリングにおいて各種コンポーネントを共有
  - 128ビットのデータを4分割して32ビット毎に処理することでコンポーネントを小型化
  - GF(((2²)²)²)の演算を利用した効率的なS-Box実装
  - 演算処理のマージと共通項のくくりだしによる論理圧縮

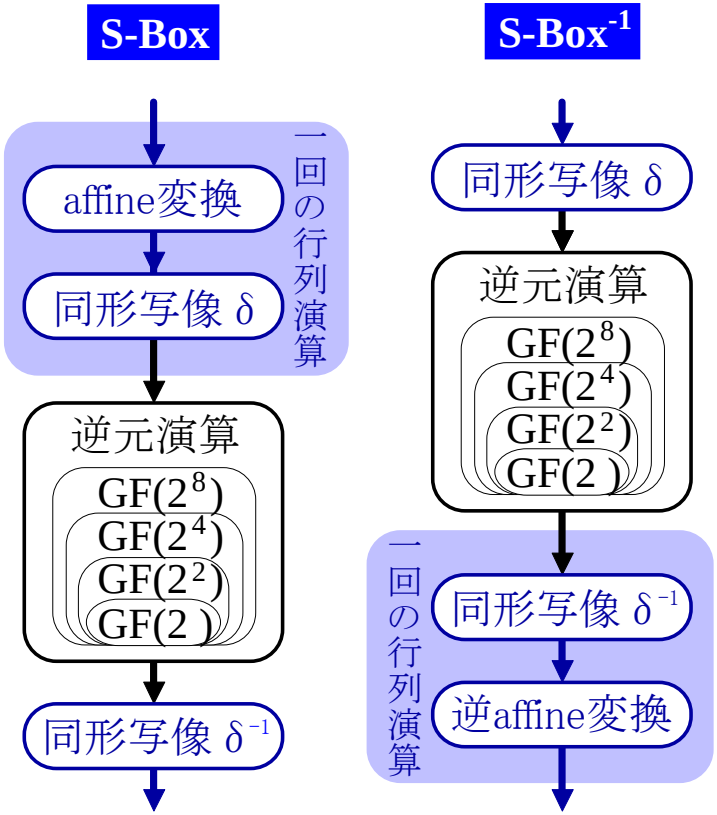


AES

- SubBytes/InvSubBytes
  - GF(2^8)の逆元演算をGF(((2^2)^2)^2)上で実行し，SubBytesとInvSubBytesで共有
  - GF((2^4)^2)上の演算よりも小型化に有利
  - 体の変換を行う同形写像はS-Boxのアファイン変換とマージ

1,396gates→294gates

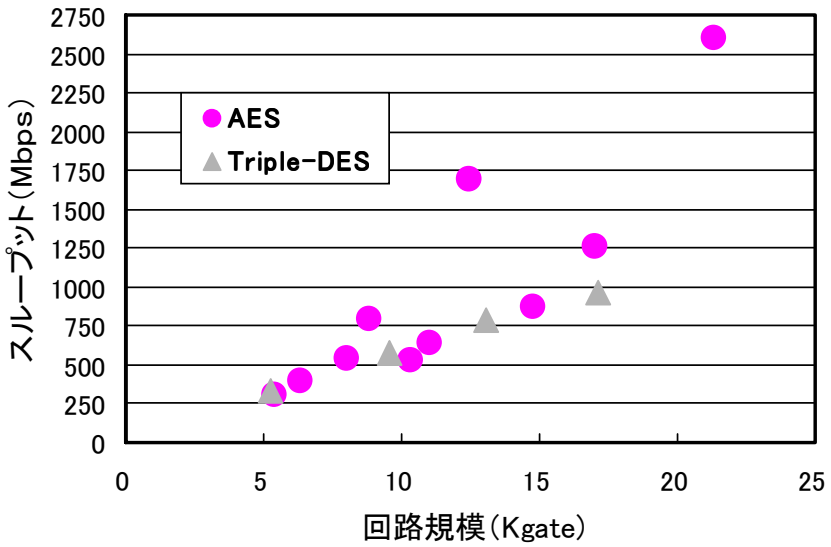
	逆元		S-Box		S-Box^{-1}	
	面積 (gates)	遅延 (ns)	面積 (gates)	遅延 (ns)	面積 (gates)	遅延 (ns)
GF(((2^2)^2)^2)	173	2.55	294	3.69	←共有	←共有
GF((2^4)^2)	341	2.50	362	3.75	←共有	←共有
Table	—	—	696	2.71	700	2.29

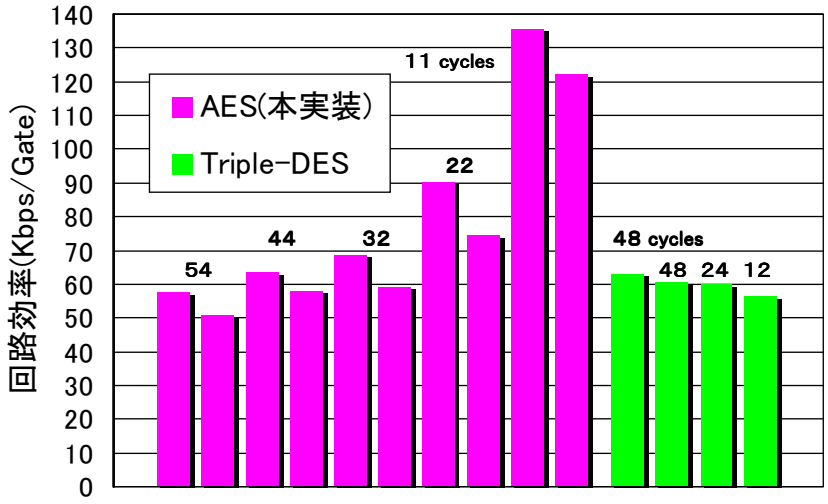


AES vs Triple-DES

- RijndaelはS-Boxサイズに応じてバス幅32/64/128ビットを実装
- DESは1/2/4段分のラウンド関数をデータパスに実装
- 回路効率(Kbps/Gate)は小型実装ではほぼ同等だが，高速実装ではAESIが優れている

	Cycles	回路規模 (gates)	動作周波数 (MHz)	スループット (Mbps)	スループット/回路規模 (Kbps/gates)	備考
AES	54	5,398	131.2	311	57.63	回路規模優先
	44	10,990	219.3	638	58.05	動作速度優先
	32	14,777	218.8	875	59.23	〃
	22	17,016	217.9	1,268	74.49	〃
	11	21,337	224.2	2,609	122.28	〃
DES	48	5,273	250.0	333	63.22	回路規模優先
	48	9,536	434.8	580	60.82	動作速度優先
	48	13,053	294.1	784	60.08	2段分実装
	48	17,140	181.8	970	56.57	4段分実装



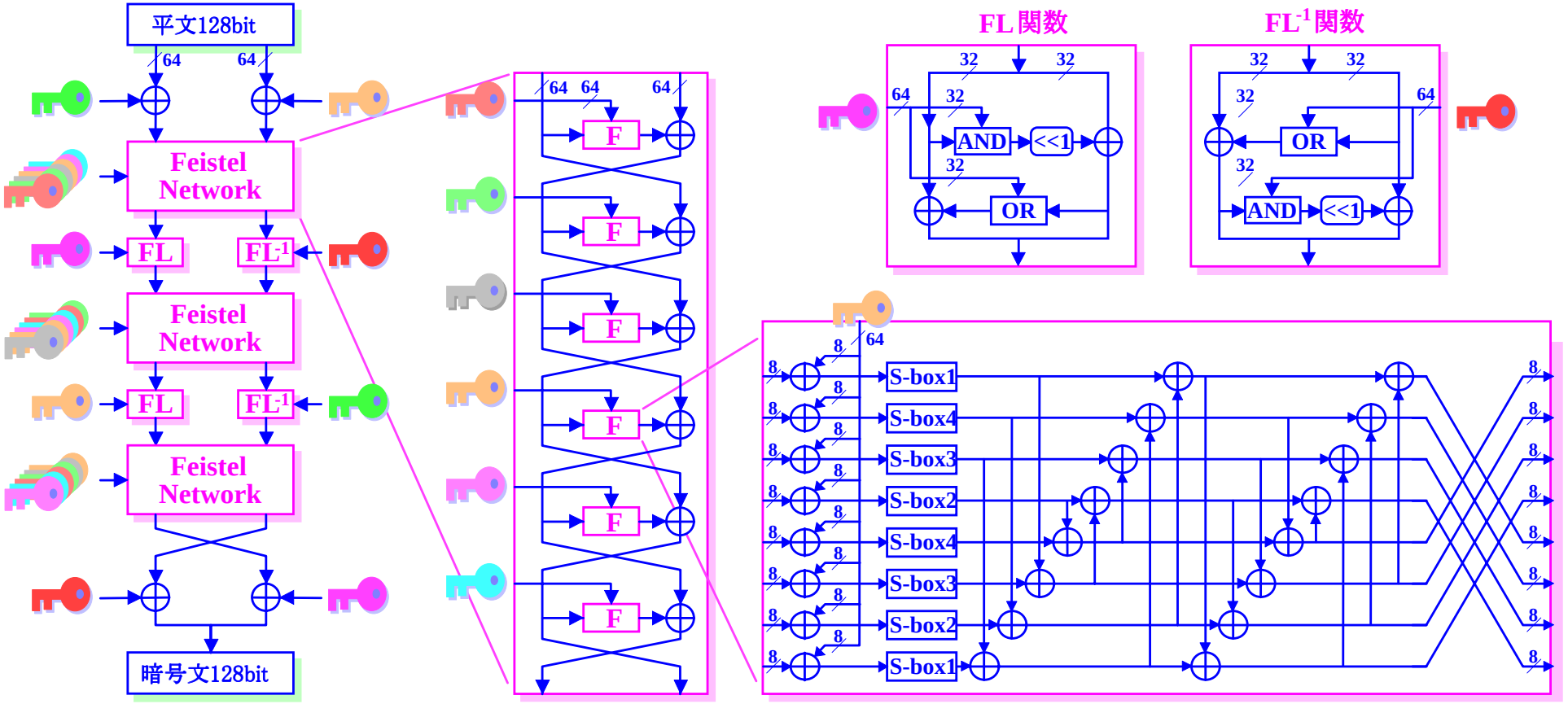


AES

- H/W化のポイント
  - ガロア体の演算を多用しているため，コンポーネントの多くがXORアレイとして記述可能で論理圧縮が容易.
  - 部分体の演算を用いてS-Boxが大幅に小型化可能.
  - 鍵スケジューリングにラウンド関数のS-Boxが流用でき，処理が簡単なためOn-the-Fly動作が可能.
  - SPN構造であるが，暗号化と復号化で多くのコンポーネントが共有あるいはマージ可能.
  - データパスを容易に32ビット単位で分割処理可能なことから，小型実装(5.4Kgates)から高速実装(2.6Gbps)まで柔軟に対応.
  - 128ビット鍵の高速実装では暗号化/復号化とも10ないし11サイクルで実行.
  - SubBytesとMixColumuns，InvSubBytesとInvMixColumnsをマージしたT-Boxアーキテクチャを用いれば10Gbpsの超高速実装も可能.

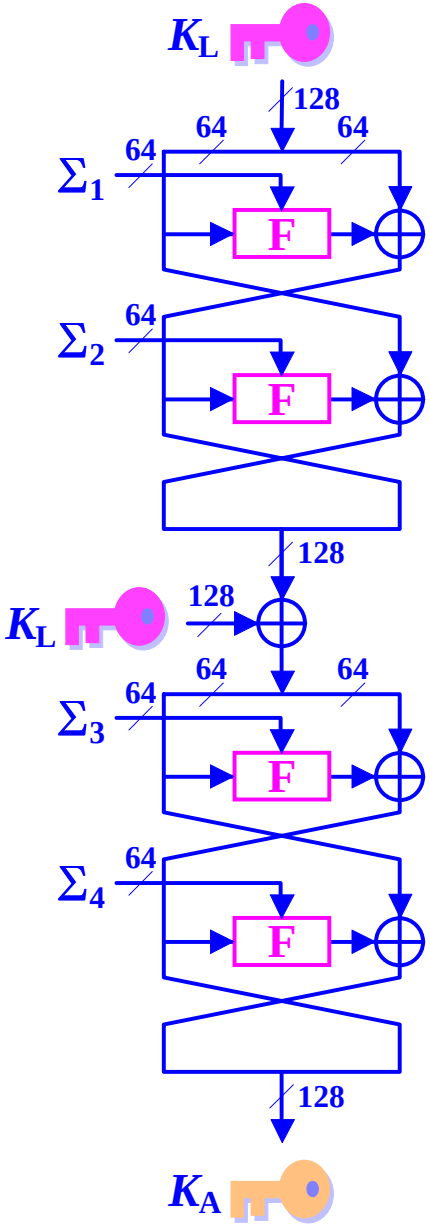
Camellia

- 暗号化／復号化アルゴリズム
  - Feistel構造にFL関数を組み込んで未知の攻撃へ対処
  - FL関数はMISTYを基に1ビット巡回シフトを加えてバイト単位の解読を困難に
  - S-boxはGF(2^4)の二次拡大体GF((2^4)^2)を利用してH/Wの実装効率を向上
  - H/Wサイズは128ビット暗号としては最小レベルの7.8Kゲート



Camellia

- 鍵スケジューリング
  - 暗号化と同じF関数を用い，XORを間に挟んだFeistel構造
  - F関数には鍵の変わりに定数 Σ_1 ～ Σ_4 を用いる
    - Σ_i は2から始まる i 番目の素数の平方根の16進表現の少数第2位からの16桁
  - 128ビット入力鍵 K_L から128ビットデータ K_A を生成
  - K_L と K_A を巡回シフトして左右半分の64桁ビットを各段の副鍵とする

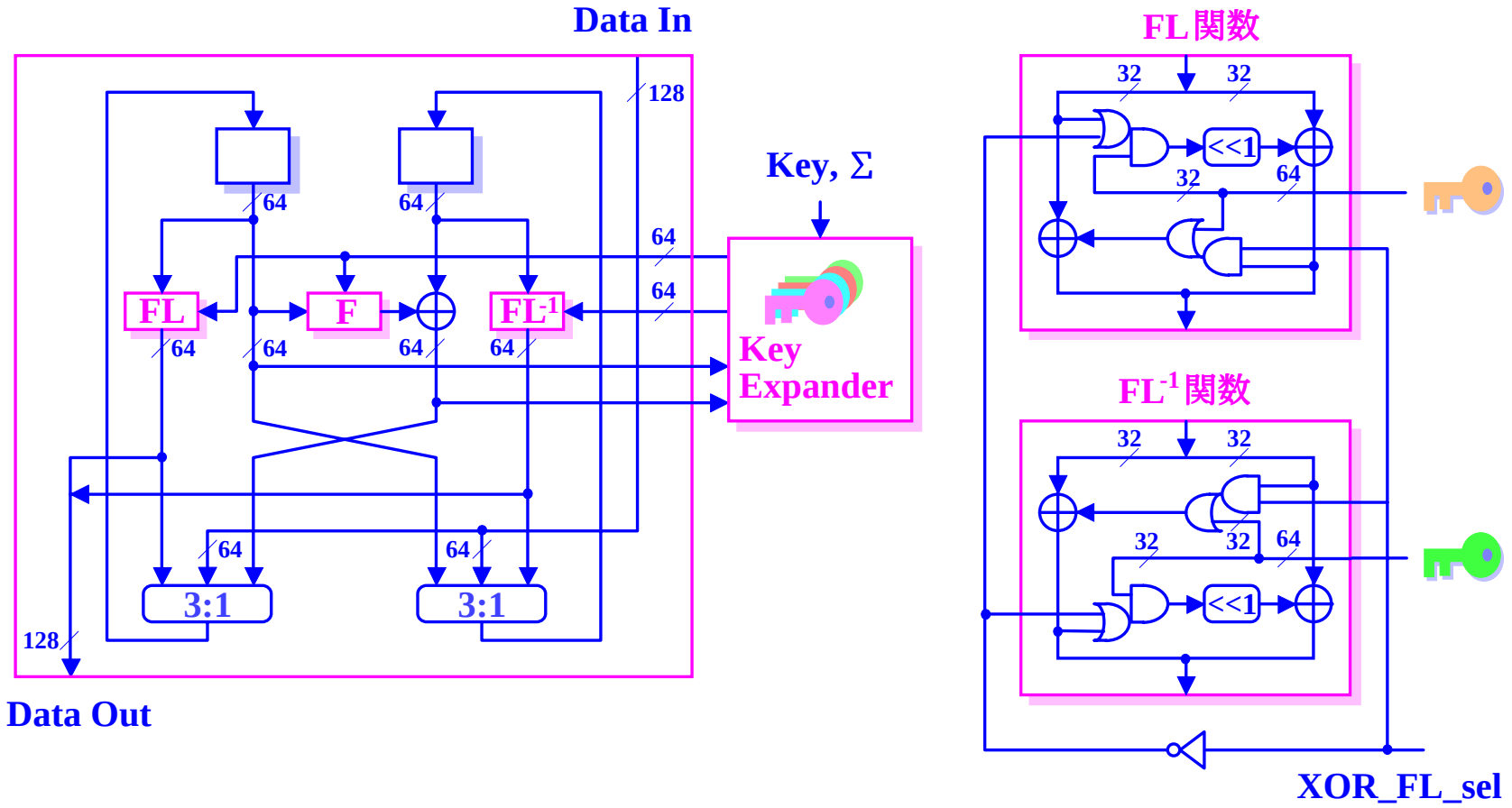


入力段	副鍵	値
初期 XOR	kw_1 kw_2	$(k_L <<< 0)_L$ $(k_L <<< 0)_R$
F(1段) F(2段) F(3段) F(4段) F(5段) F(6段)	k_1 k_2 k_3 k_4 k_5 k_6	$(k_A <<< 0)_L$ $(k_A <<< 0)_R$ $(k_L <<< 15)_L$ $(k_L <<< 15)_R$ $(k_A <<< 15)_L$ $(k_A <<< 15)_R$
FL FL^{-1}	kl_1 kl_2	$(k_A <<< 30)_L$ $(k_A <<< 30)_R$
F(7段) F(8段) F(9段) F(10段) F(11段) F(12段)	k_7 k_8 k_9 k_{10} k_{11} k_{12}	$(k_L <<< 45)_L$ $(k_L <<< 45)_R$ $(k_A <<< 45)_L$ $(k_L <<< 60)_R$ $(k_A <<< 60)_L$ $(k_A <<< 60)_R$
FL FL^{-1}	kl_3 kl_4	$(k_L <<< 77)_L$ $(k_L <<< 77)_R$
F(13段) F(14段) F(15段) F(16段) F(17段) F(18段)	k_{13} k_{14} k_{15} k_{16} k_{17} k_{18}	$(k_L <<< 94)_L$ $(k_L <<< 94)_R$ $(k_A <<< 94)_L$ $(k_A <<< 94)_R$ $(k_L <<< 111)_L$ $(k_L <<< 111)_R$
最終 XOR	kw_3 kw_4	$(k_A <<< 111)_L$ $(k_A <<< 111)_R$

Σ_1	0xA09E667F3BCC908B
Σ_2	0xB67AE8584CAA73B2
Σ_3	0xC6EF372FE94F82BE
Σ_4	0x54FF53A5F1D36F1C

Camellia

- ループ処理のデータパスアーキテクチャ
  - F関数，FL/FL^{-1}関数（鍵加算），データ入力は3:1セレクタで切り替える
  - 鍵加算のXORをFL/FL^{-1}関数のコンポーネントと共有し，AND-ORとOR-ANDゲートで切り替える
  - 鍵スケジューリング部はシフトレジスタで構成し，暗号化／復号化部のF関数とFL関数（Σ 加算に使用）を共有

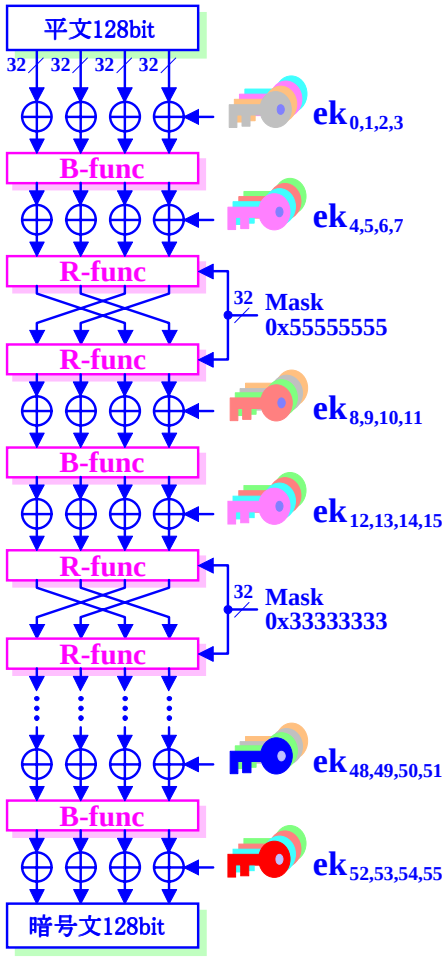


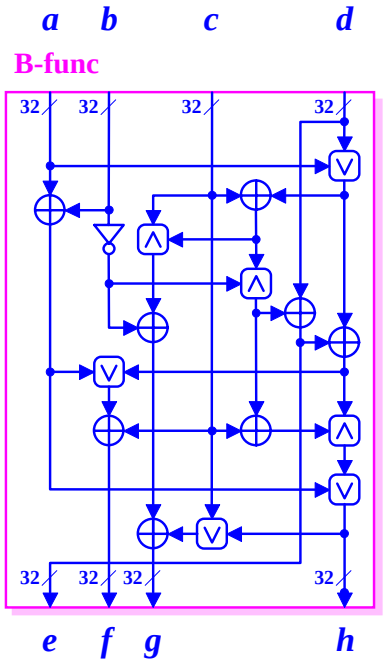
Camellia

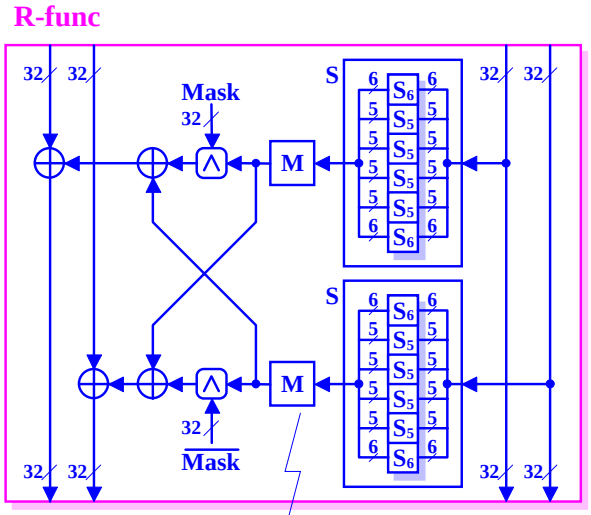
- H/W化のポイント
  - 暗号化と復号化のデータパスが簡単に共有できるため，SPN構造をもつSC2000やHierocrypt-3に比べ小型化に適している.
  - S-Boxで小型化に向くGF($(2^4)^2$)の乗法逆元演算を使用.
  - F関数が単純で動作周波数は3つのアルゴリズム中で最も高いと思われる.
  - 64ビット処理のF関数を32ビット単位の処理ブロックに分けることも可能.
  - 拡大鍵生成部が簡単でラウンド関数を流用できる．中間鍵から拡大鍵を生成するの処理は巡回シフトなのでOn-the-Flyで可能
  - 鍵レジスタが256ビット分必要.
  - Feistelのためスループットは低い.
  - 暗号化/復号化とも18～22サイクルで実行可能.

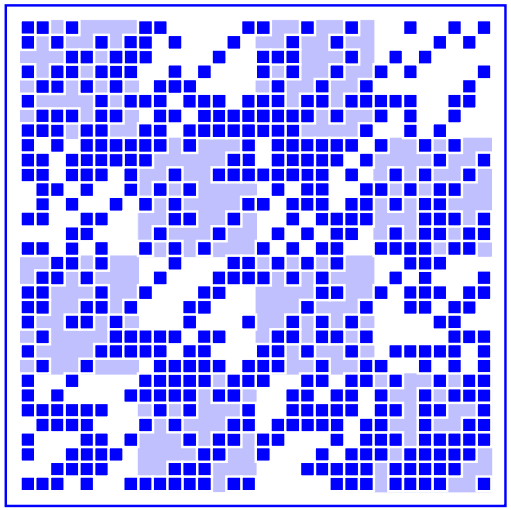
SC2000

- 暗号化アルゴリズム
  - SPNとFeistelを融合した特殊な構造
  - 暗号化と復号化で同じR-funcを使用
  - 暗号化はB-funcを復号化はB^{-1}-funcを使用





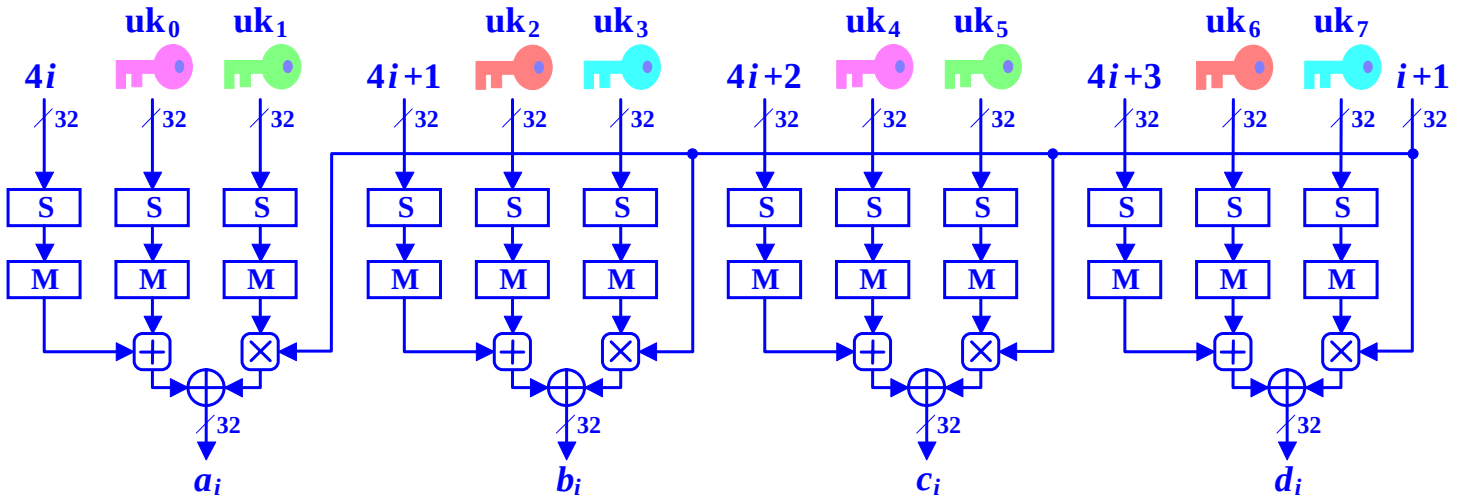


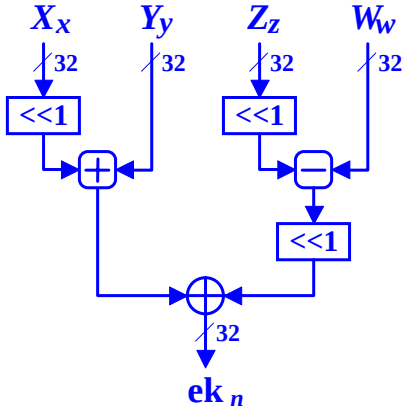


32*32
X0R
matrix

SC2000

- 鍵スケジューリング
  - S-Box，MDSマトリクス，カウンタ，加減算，乗算，巡回シフト，XOR演算を使用
  - 中間鍵 $\{a_i, b_i, c_i, d_i\}$ ($i = 0,1,2$) に384ビットのレジスタが必要
  - 拡大鍵の生成に中間鍵の使用順を決めるテーブルを使用しており，On-the-Flyでの鍵生成は巨大なスイッチングボックスが必要なため難しい





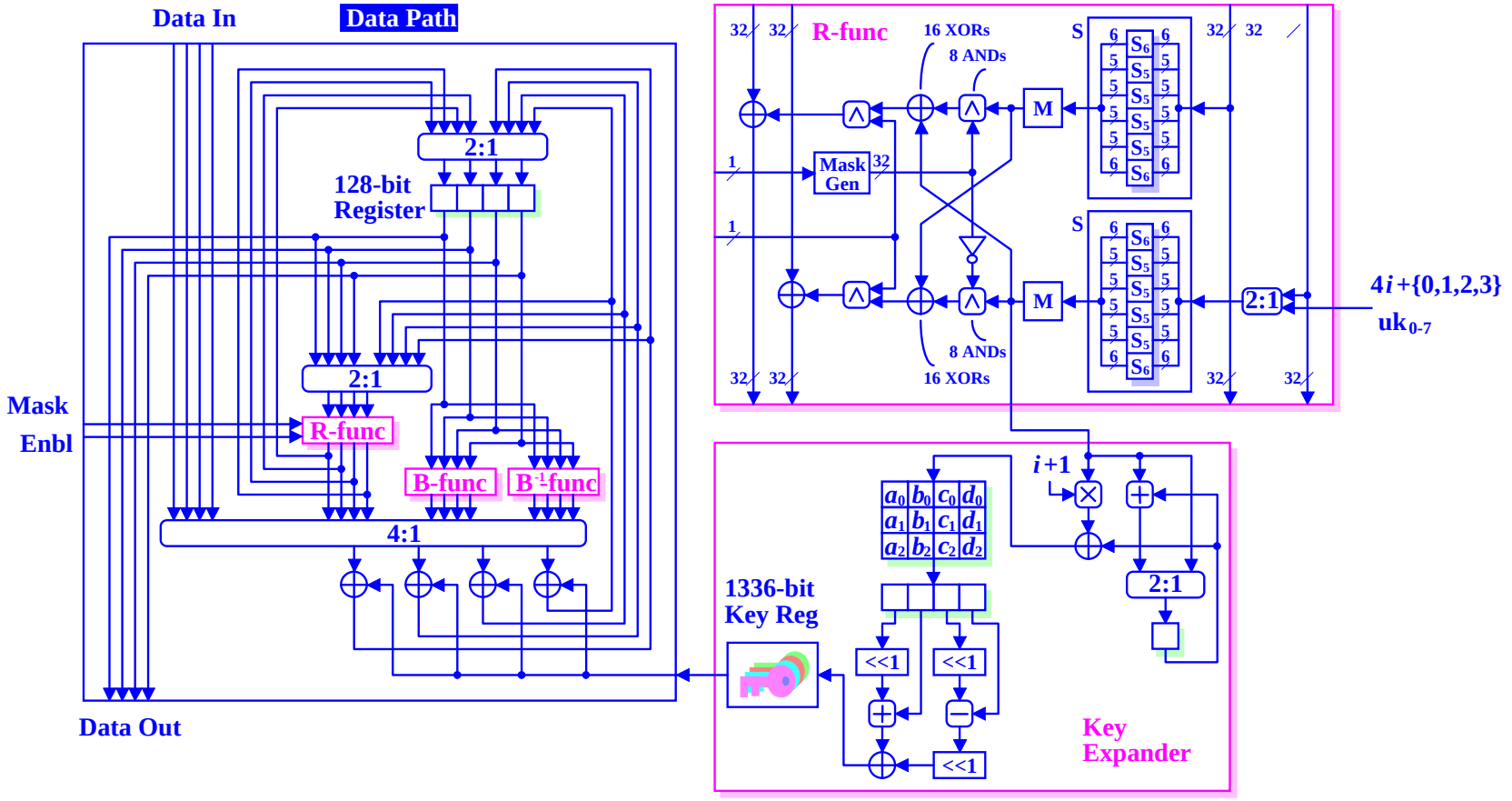
$t \equiv n + \lfloor n/36 \rfloor \pmod{12}$

t	X	Y	Z	W
0	a	b	c	d
1	b	a	d	c
2	c	d	a	b
3	d	c	b	a
1	a	c	d	b
5	b	d	c	a
6	c	a	b	d
7	d	b	a	c
8	a	d	b	c
9	b	c	a	d
10	c	b	d	a
11	d	a	c	b

$s \equiv n \pmod{9}$

s	x	y	z	w
0	0	0	0	0
1	1	1	1	1
2	2	2	2	2
3	0	1	0	1
1	1	2	1	2
5	2	0	2	0
6	0	2	0	2
7	1	0	1	0
8	2	1	2	1

- ループ処理のデータパスアーキテクチャ
  - R-funcにANDゲートを挿入し Enbl＝0 のときPass Throughに
  - マスクパターンは0x33333333と0x55555555の2種類なのでコントロールは1ビットとし，AND→XOR部のゲート数が削減可能
  - R-funcのS→M変換を中間鍵 $\{a_i, b_i, c_i, d_i\}$ の生成にも用いる

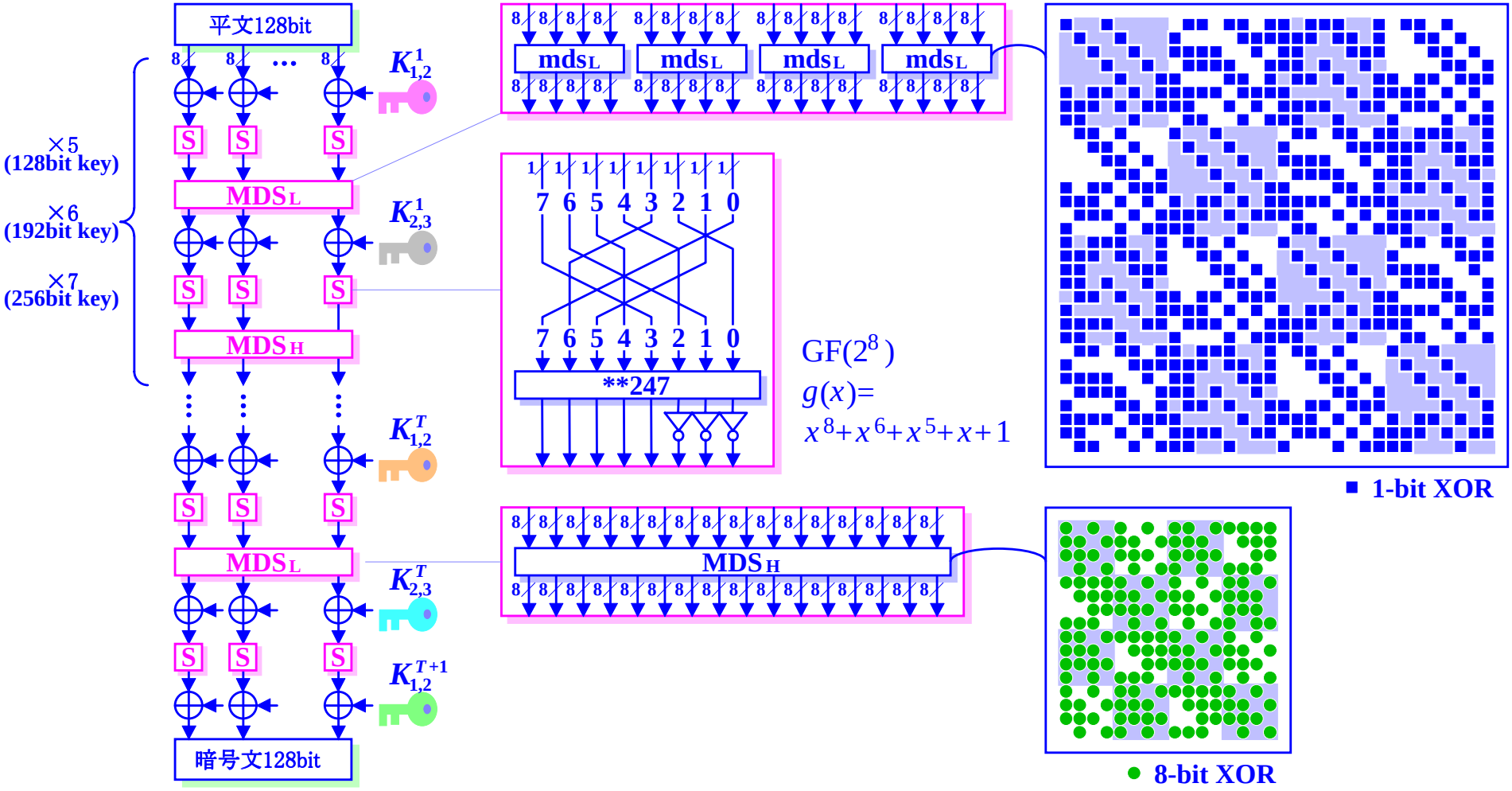


SC2000

- H/W化のポイント
  - 処理単位の異なる二つのラウンド関数R-funcとB-func/B^{-1}-funcを用いており，128ビットデータを分割して小さい単位で効率的に処理することは難しい．
  - 暗号化と復号化でB-func/B^{-1}-func以外のコンポーネントは共有可能だが，そのためのセレクタがやや多く，回路規模と動作速度に影響．
  - 中間鍵生成にR-funcのS→M変換を流用できるが，それに加え加算器，減算器，XORなどが必要とされる．
  - 12個の32ビット中間鍵を4つづつ組み合わせて拡大鍵を生成するが，組み合わせ決定のテーブルが複雑で，On-the-Flyで実行するためには巨大なスイッチングマトリクスが必要となる．
  - あるいは拡大鍵の事前計算に1,336ビットのレジスタが必要となる．
  - ラウンド関数は共有できる部分も多いが3つのアルゴリズム中で最も大きいと思われる．
  - ラウンド関数部が複雑なため，遅延時間はもっとも長い（動作周波数が低い）と思われる．
  - 暗号化/復号化とも14サイクルで実行可能

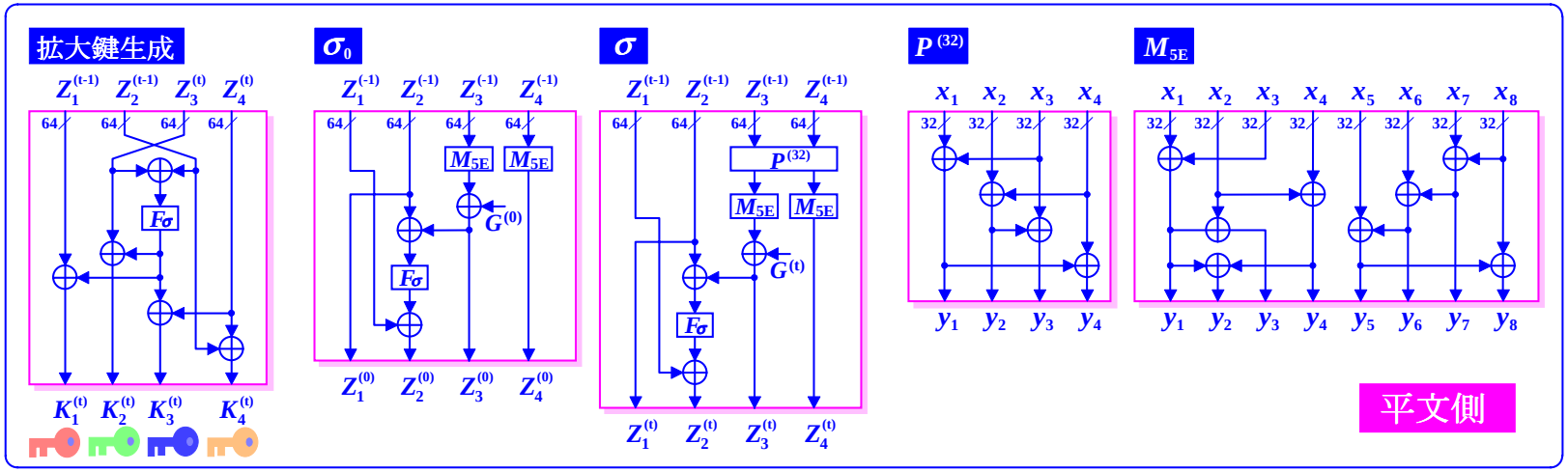
Hierocrypt-3

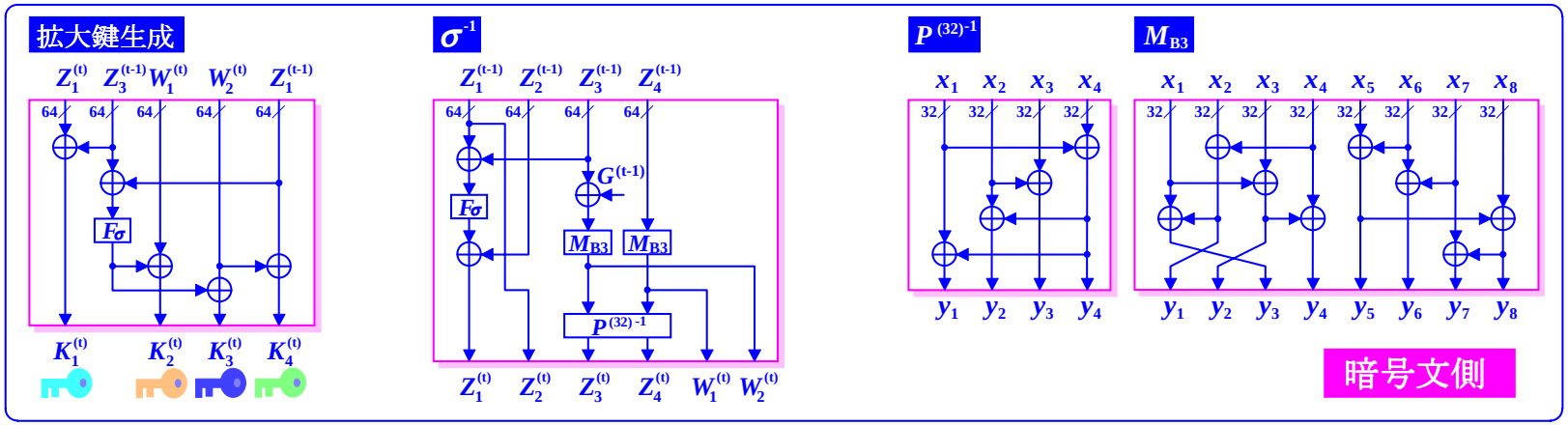
- 暗号化アルゴリズム
  - MDS_LとMDS_Hの二種類の拡散層を持つSPN構造
  - S-Boxは$GF(2^8)$上の演算を利用

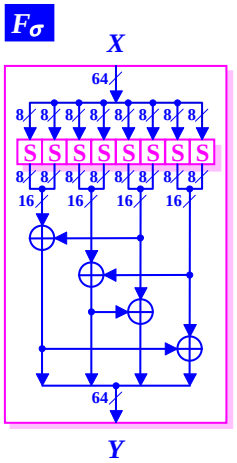


Hierocrypt-3

- 鍵スケジューリング
  - 2種類の鍵生成回路を用いる
  - On-the-Fly鍵生成には約2,300個ものXORが必要



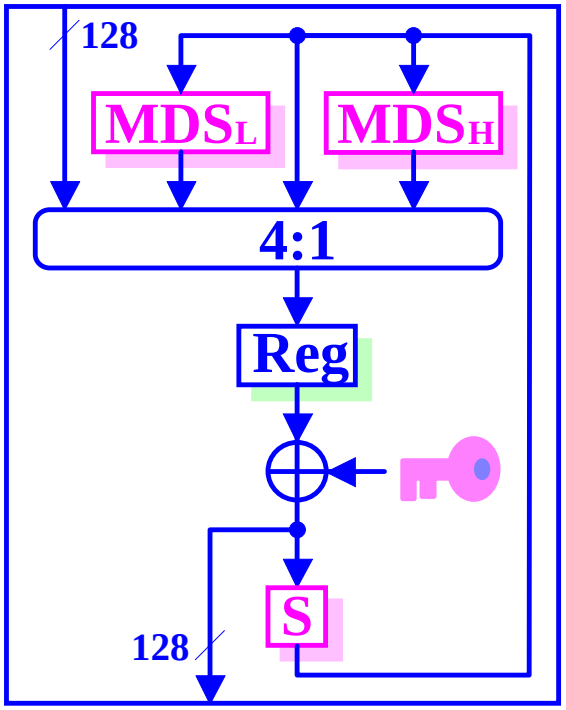




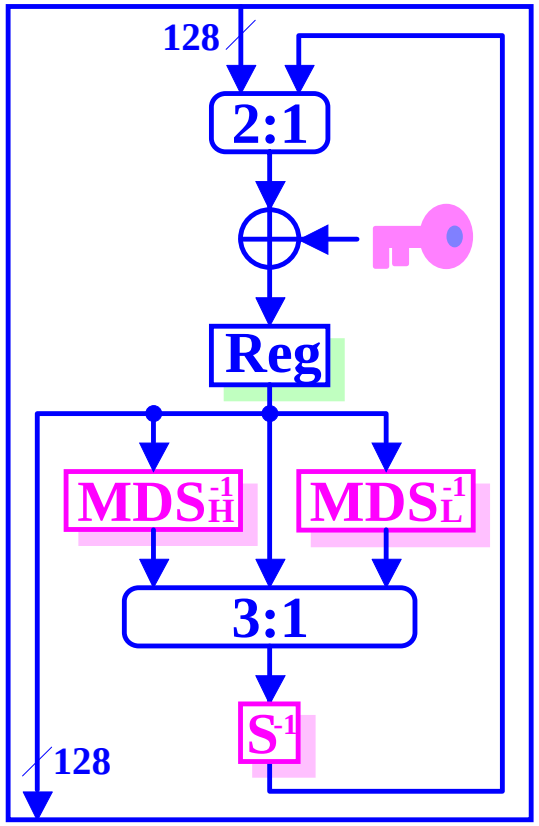
Hierocrypt-3

- ループ処理のデータパスアーキテクチャ
  - S-Boxは鍵スケジューリングにも流用可能だが，そのときラウンド関数の処理は同時に行えない．

暗号化



復号化

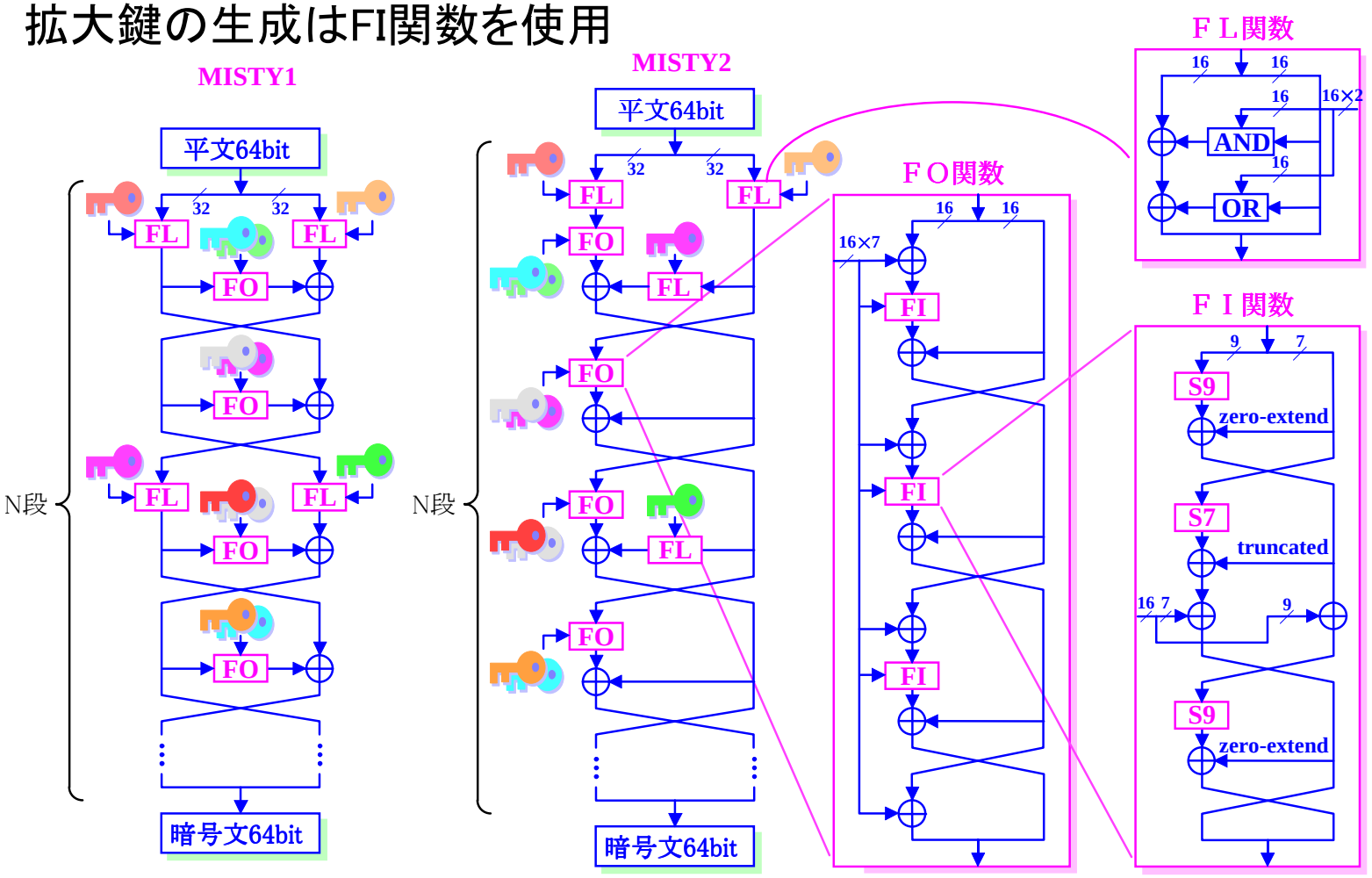


Hierocrypt-3

- H/W化のポイント
  - 高速なラウンド関数を持つSPN構造であるためスループットは最も高いと思われる.
  - S-BoxにGF(2^8)の演算を用いており，暗号化が247(=-8)乗，復号化が223(=-32)乗のため，正規基底を用いた小型共有化が可能.
  - 拡散層MDS_LとMDS_H処理単位が異なるため，データパスを分割して小型化することは難しい.
  - 暗号化と復号化のデータパス共有化は，AESよりも多くのセレクタが必要で配線領域が大きくなり，動作周波数が低下する.
  - 鍵スケジューリングに約2,300個のXORが必要．S-Boxはラウンド関数を流用可能だが，ラウンド関数を同時に処理することはできない.
  - あるいは鍵スケジューリングのXORを共有すると，256ビット拡大鍵の生成が2サイクルでおさまらず，On-the-Fly生成が不可能.
  - ラウンド数は鍵長に応じて13,15,17段

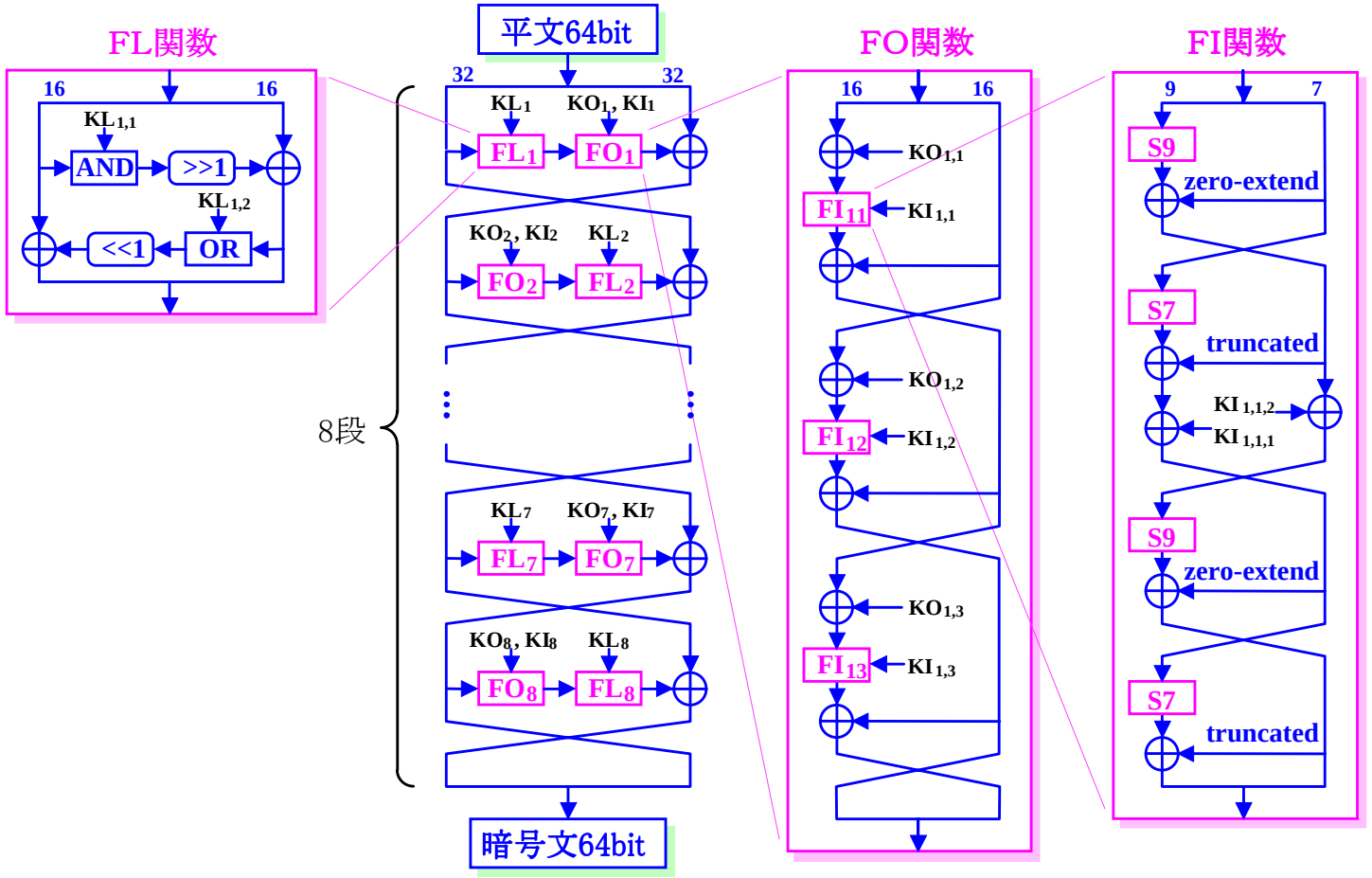
MISTY

- 三菱電機が1995年に開発した鍵長128bitの64bitブロック暗号
  - MISTY1はFO関数とFL関数を含むFeistel構造
  - MISTY2はFO関数が並列処理可能で12段が推奨されている
  - FO関数は入れ子構造を持ち，S-BOXはガロア体上のべき乗算を採用
  - 差分解読法と線形解読法に対する安全性を考慮して設計
  - 拡大鍵の生成はFI関数を使用



KASUMI

- 暗号化アルゴリズム
  - MISTYと同じFL，FO，FI関数およびS-Box S7を使用．（S9は多少異なる）
  - FL関数の位置がMISTY2と異なる
  - 鍵スケジューリングをシフトとXORに簡略化
  - 3GPPでは復号化だけ使用するため復号化はスペックに定義されていない



KASUMI

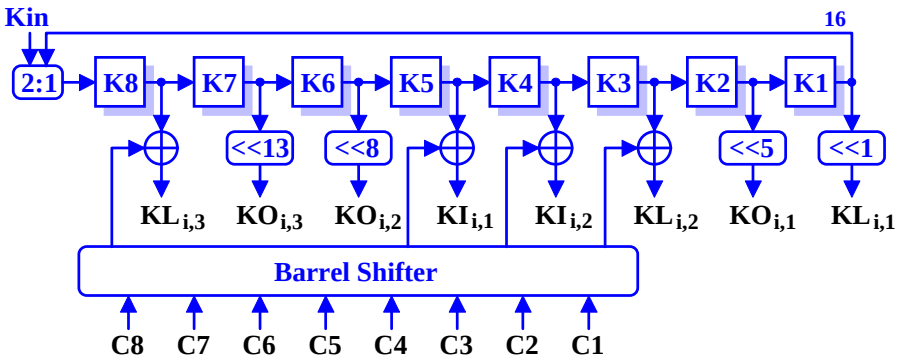
- 鍵スケジューリング
  - 定数*Cj*とのXORとシフト演算だけなので回路が高速で小型

$$Kj' = Kj \oplus Cj \quad (1 \le j \le 8)$$

{C1, C2, C3, C4, C5, C6, C7, C8}

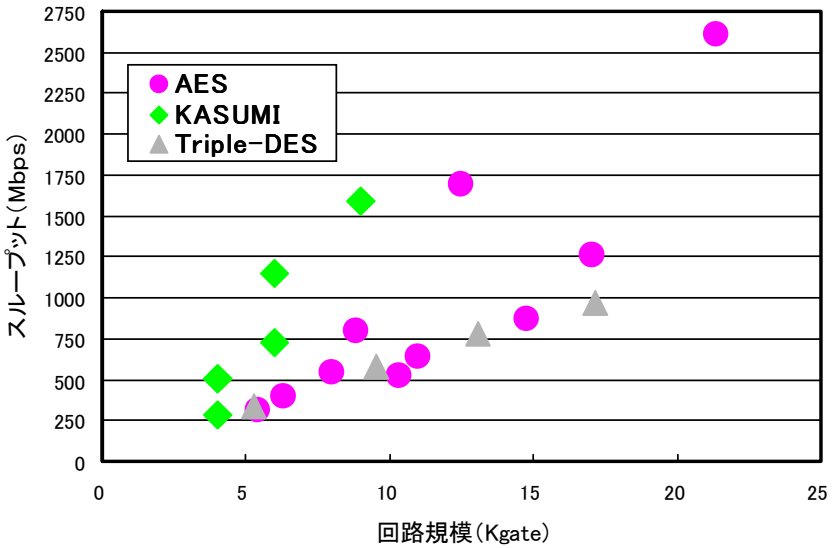
= {0x0123, 0x4567, 0x89AB, 0xCDEF, 0xFEDC, 0xBA98, 0x7654, 0x3210}

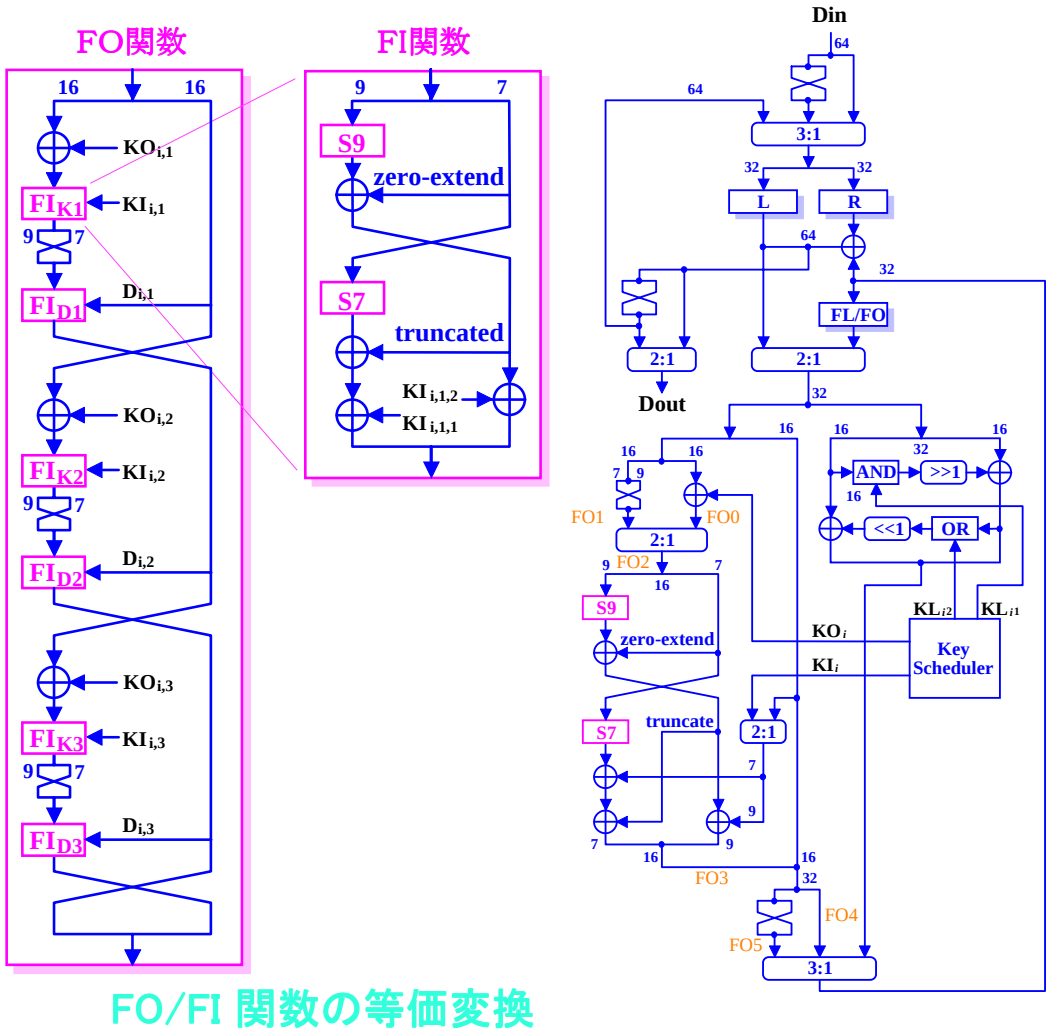
i	KLi,1	KLi,2	KOi,1	KOi,2	KOi,3	KIi,1	KIi,2	KIi,3
1	K1<<1	K3'	K2<<5	K6<<8	K7<<13	K5'	K4'	K8'
2	K2<<1	K4'	K3<<5	K7<<8	K8<<13	K6'	K5'	K1'
3	K3<<1	K5'	K4<<5	K8<<8	K1<<13	K7'	K6'	K2'
4	K4<<1	K6'	K5<<5	K1<<8	K2<<13	K8'	K7'	K3'
5	K5<<1	K7'	K6<<5	K2<<8	K3<<13	K1'	K8'	K4'
6	K6<<1	K8'	K7<<5	K3<<8	K4<<13	K2'	K1'	K5'
7	K7<<1	K1'	K8<<5	K4<<8	K5<<13	K3	K2'	K6'
8	K8<<1	K2'	K1<<5	K5<<8	K6<<13	'K4'	K3'	K7'



KASUMI

- データパスアーキテクチャ
  - FI関数を二分割してS9とS7を一つづつ含む関数に変換
  - 小型実装は1ラウンドでこの関数を6回使用
  - FI関数の個数を増やすことで高速化が可能





FO/FI 関数の等価変換

KASUMI

- H/W化のポイント
  - S-BoxがS7，S9一つずつで済むためLook-up-table方式でOK
  - XOR数も少なく共有が可能
  - FL関数と鍵スケジューラーのシフト演算量は固定ビットなので配線をねじるだけでよい
  - ラウンド関数部と鍵スケジューラーで共有部品がない
  - 入れ子構造のどこでデータパスを切るかで，小型（3.5Kゲート）～高速（1.6Gbps)までの様々な回路実装が可能
  - 2Gbpsを超える高速実装は難しいが，AESよりも回路使用効率が高い
  - ラウンド数はアーキテクチャにより8, 32, 56段
  - とにかくコンポーネントとその組み合わせがシンプル

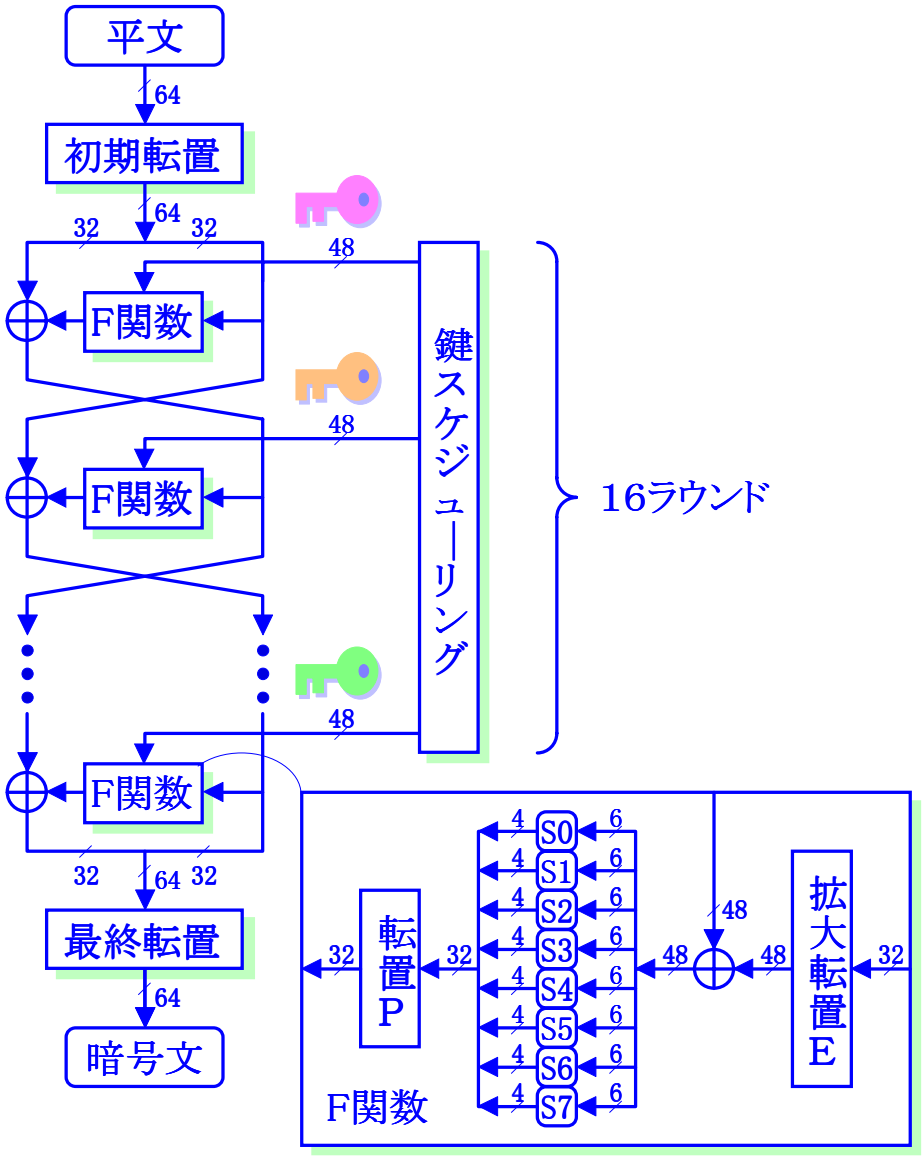
小型・高速化のポイント

- セレクタと配線に要注意
  - 不用意な関数の共有や複数関数の導入はH/Wでは　×ダメ×ダメ
  - シフト演算の多用も　×ダメ×ダメ
- 鍵スケジューリングを侮るな
  - 不用意にラウンド関数を使うのは　×ダメ×ダメ
  - テンポラリレジスタの多用も　×ダメ×ダメ
  - 下手すると鍵スケジューリングのほうが大きくなったりして

Backup

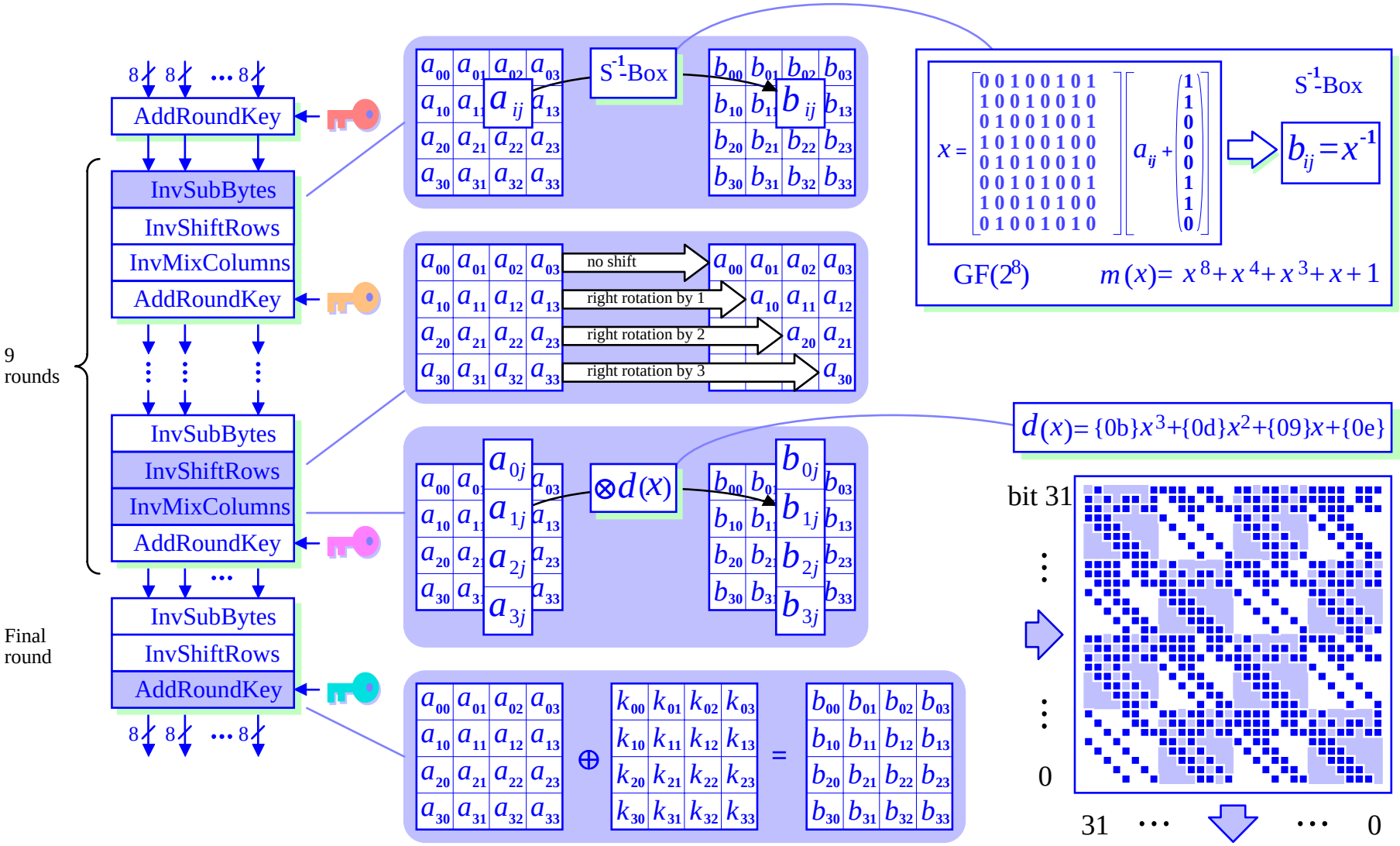
Triple-DES

- 暗号化／復号化アルゴリズム
  - 64ビットデータを32ビットずつ処理するFeistel構造
  - 暗号化では56ビット鍵を3つ用いて16ラウンドのDESを，暗号化-復号化-暗号化と3回繰り返し，トータルは48ラウンド
  - S-boxは6ビット入力4ビット出力の乱数テーブルを8つ使用
  - 鍵スケジューリングは単純巡回シフト
  - 構成がシンプルでありH/Wサイズは1round/clock処理で約5Kゲート
  - 0.11umCMOSで300Mbps以上



AES

- 復号化アルゴリズム
  - SPN構造だが，暗号化と復号化で多くのコンポーネントが共有可能

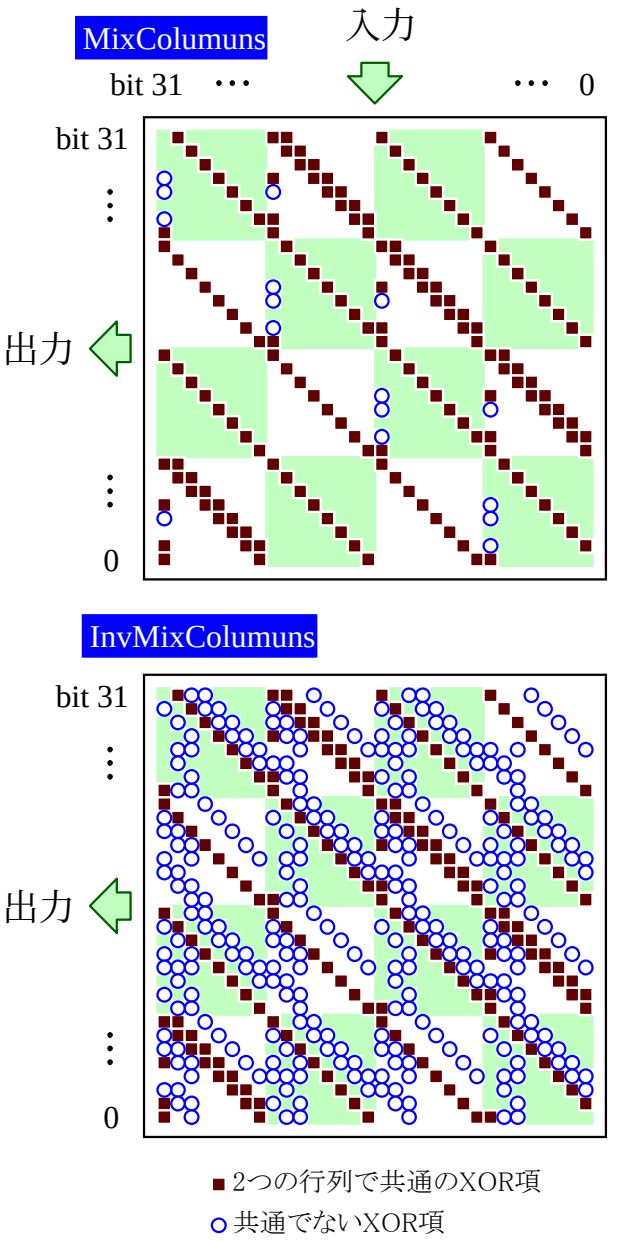


AES

- MixColumns／InvMixColumns
  - GF(2^8)の4つの元を3次の4項式とみなし，それにC(x)，C^{-1}(x)を乗じる定数乗算
  - 定数乗算はXORアレイで記述できる
  - MixColumnsのほとんどのXOR項はInvMixColumnsに含まれるので両者を簡単にマージ可能
  - マージしたアレイ内部でもさらに共通項のくくり出しによって小型化を行う

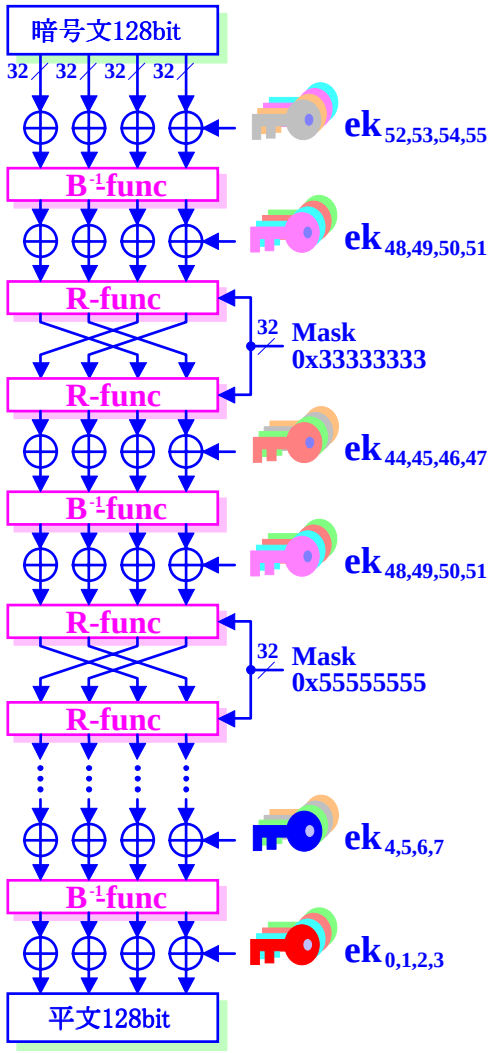
592XORs→210XORs

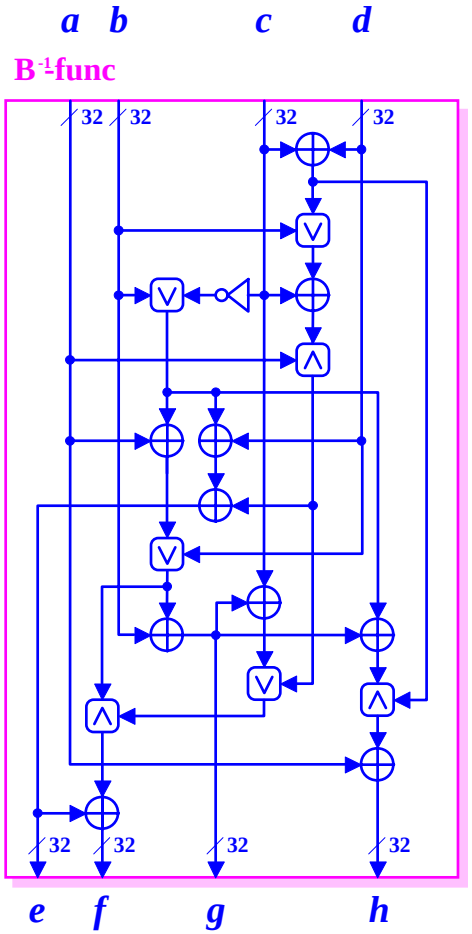
	MixCo		InvMixCo		MixCo+ InvMixCo	
	基本	共有	基本	共有	基本	共有
XOR数	152	108	440	177	592	210
遅延段数	3	3	5	7	5	7

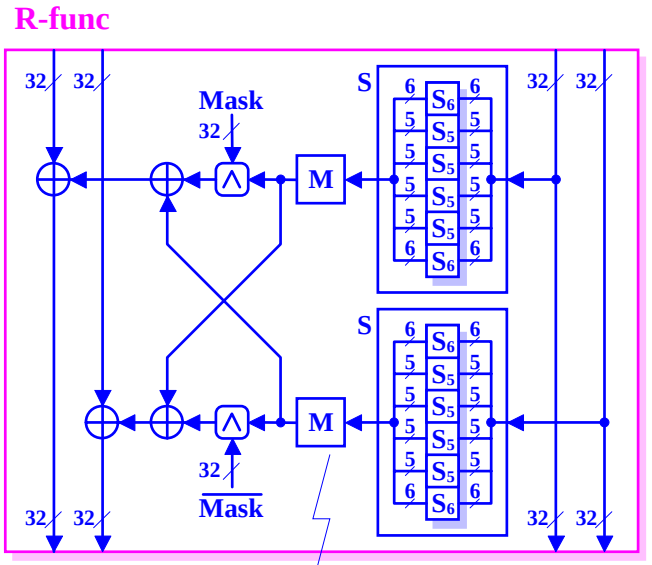


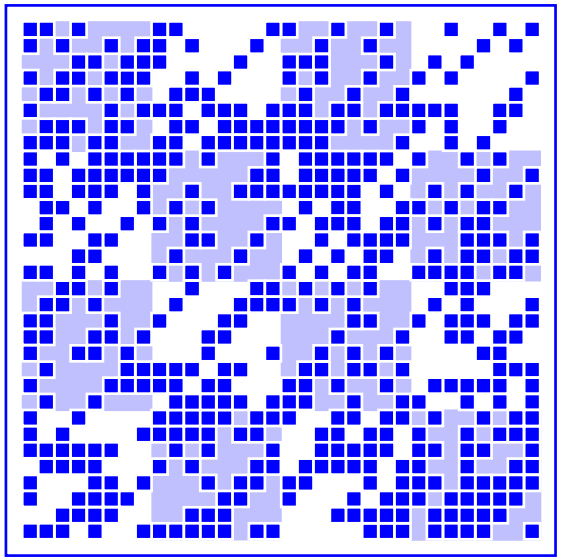
SC2000

- 復号化アルゴリズム





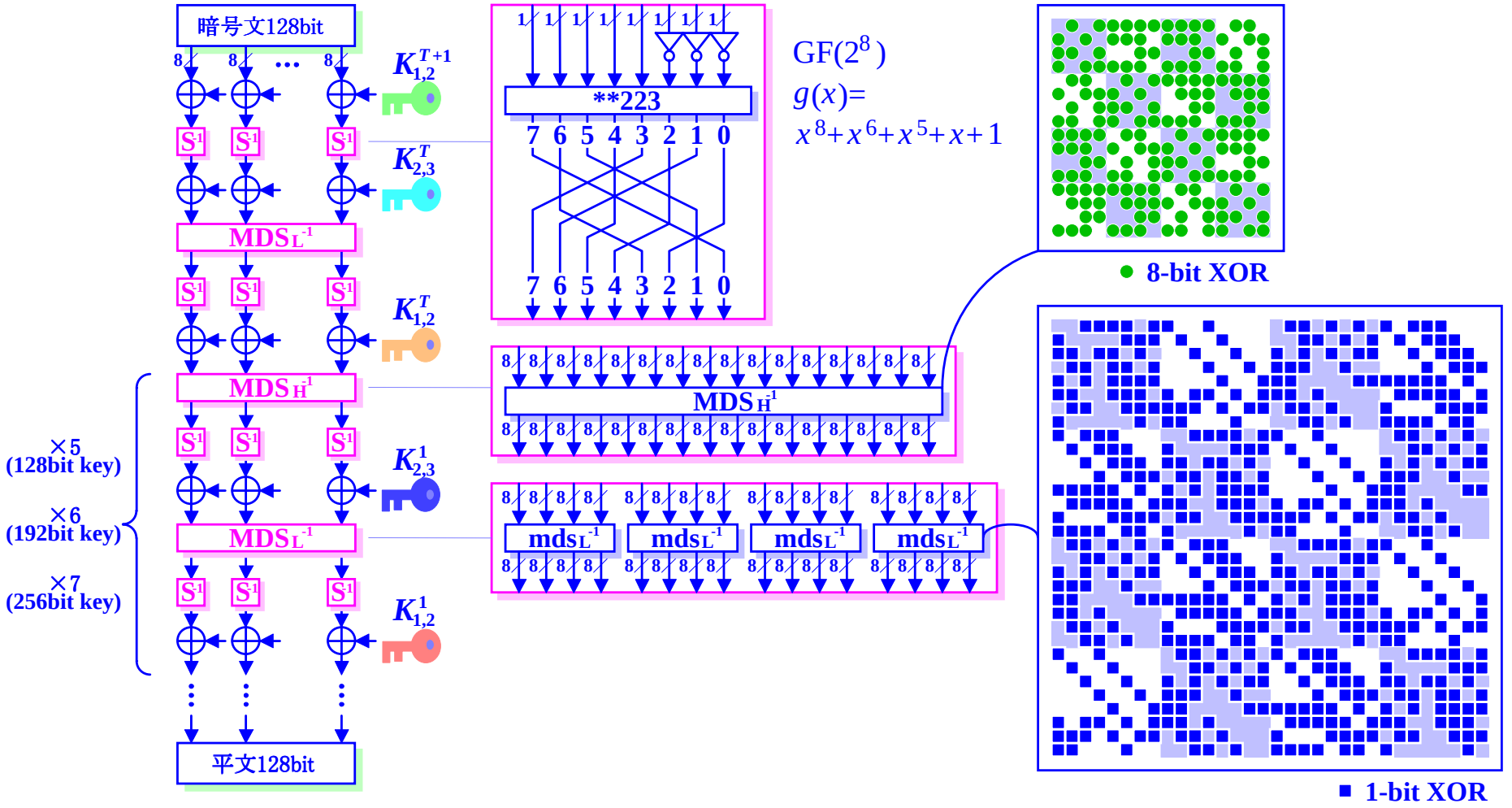




32*32
X0R
matrix

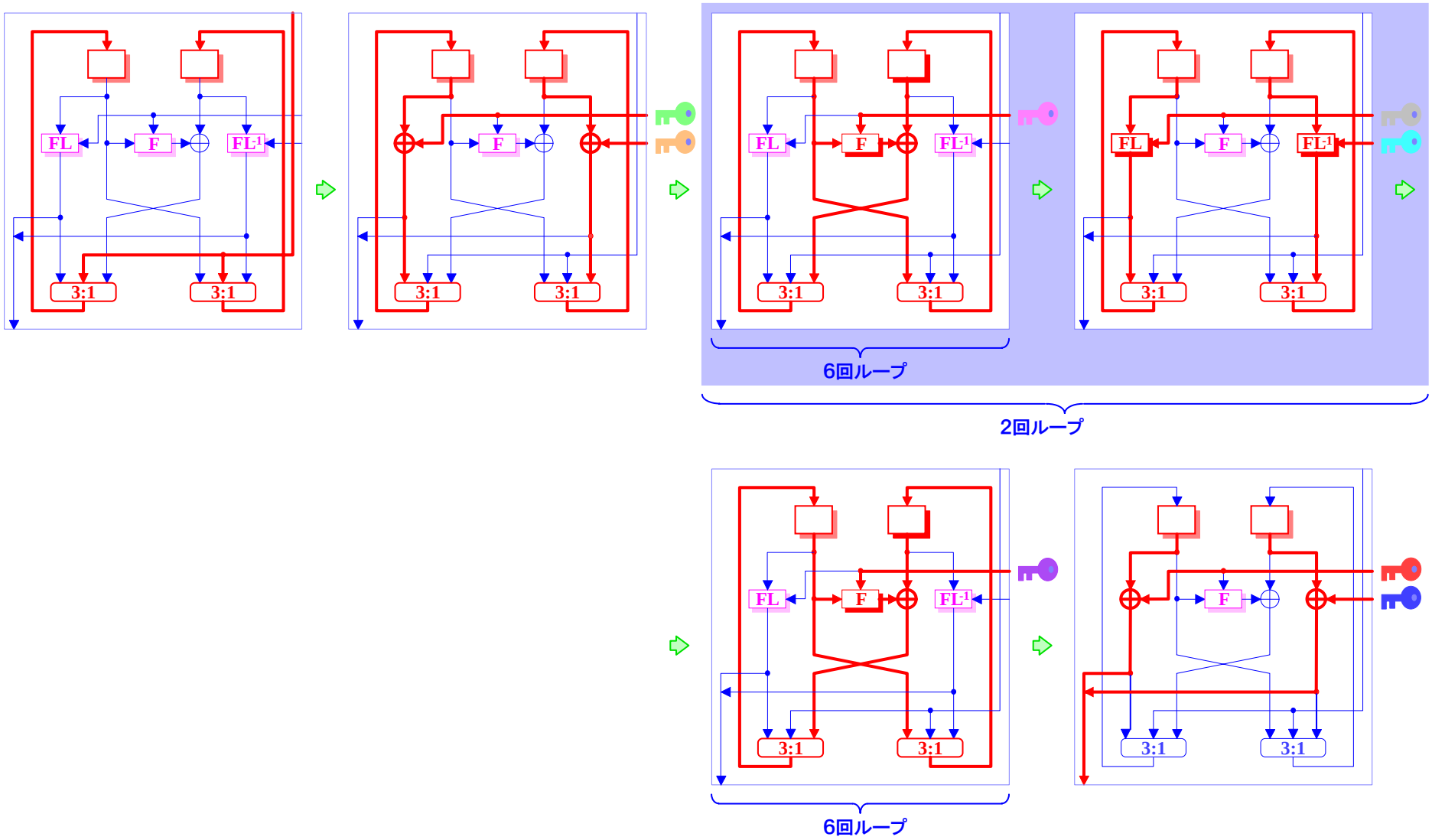
Hierocrypt-3

- 復号化アルゴリズム
  - 復号化と共有できるコンポーネントは少ない



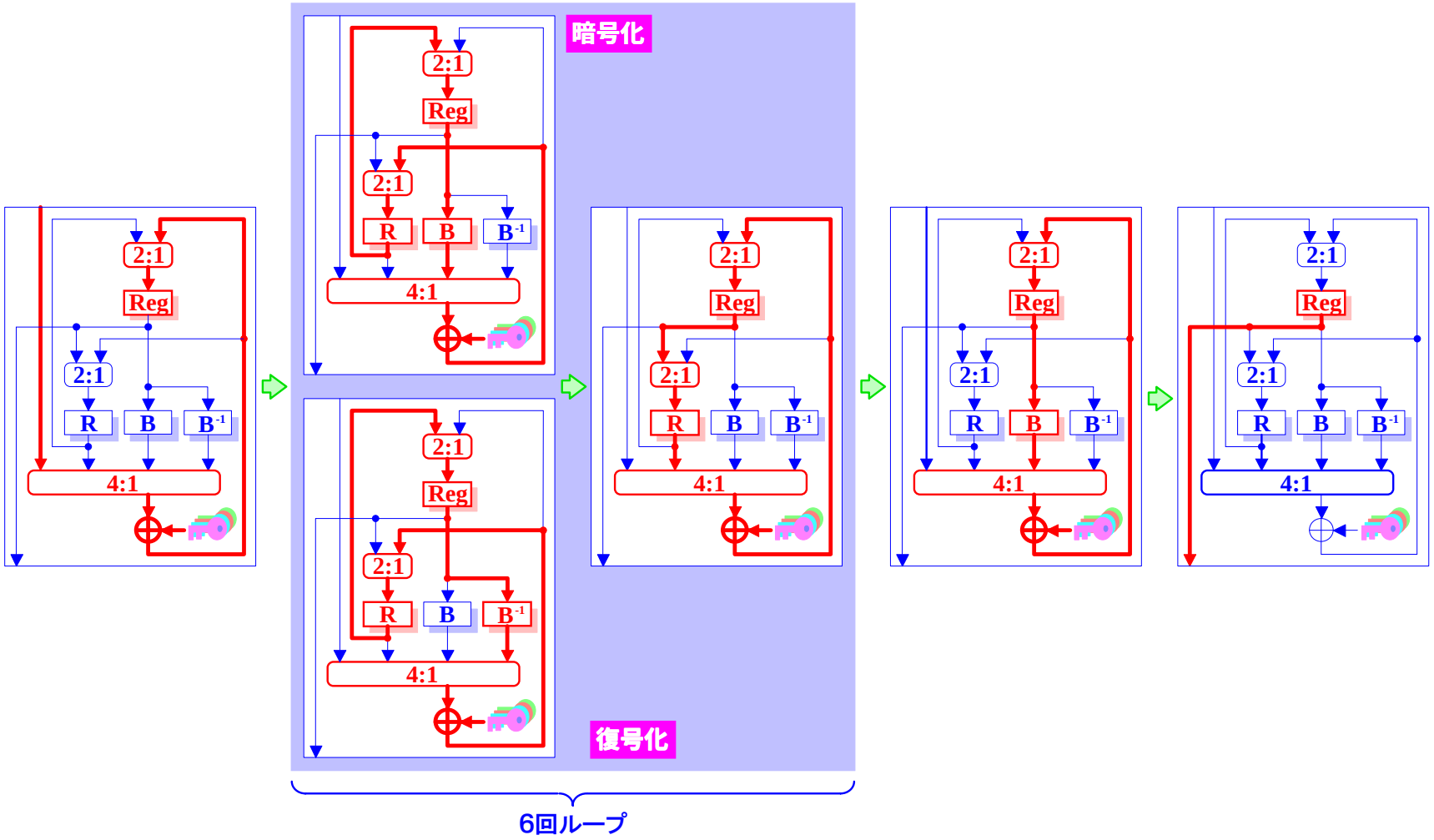
Camellia

- ECBモードのデータパス



SC2000

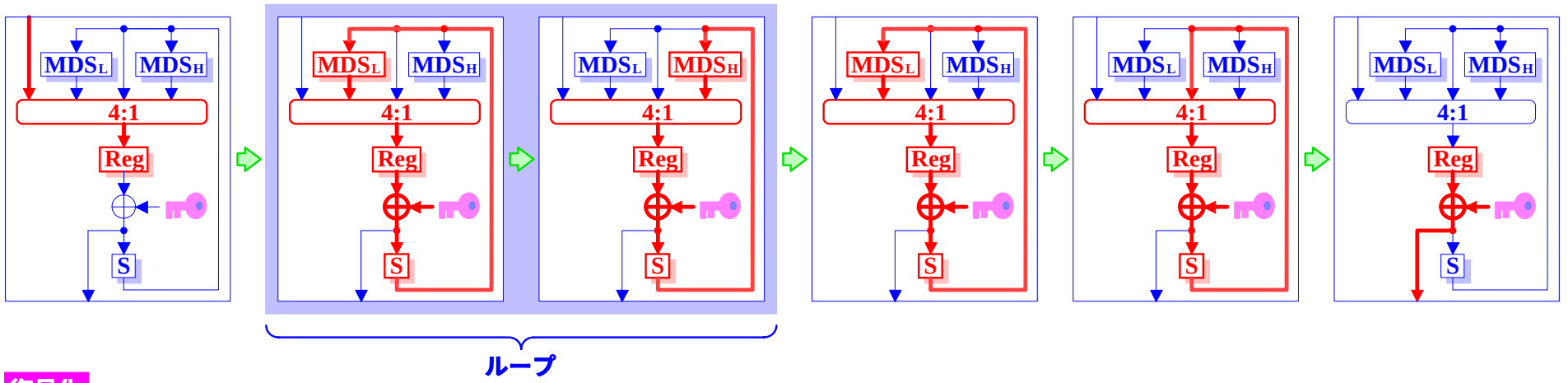
- ECBモードのデータパス
  - 暗号化と復号化はB−funcとB^{-1}−funcを切り替える



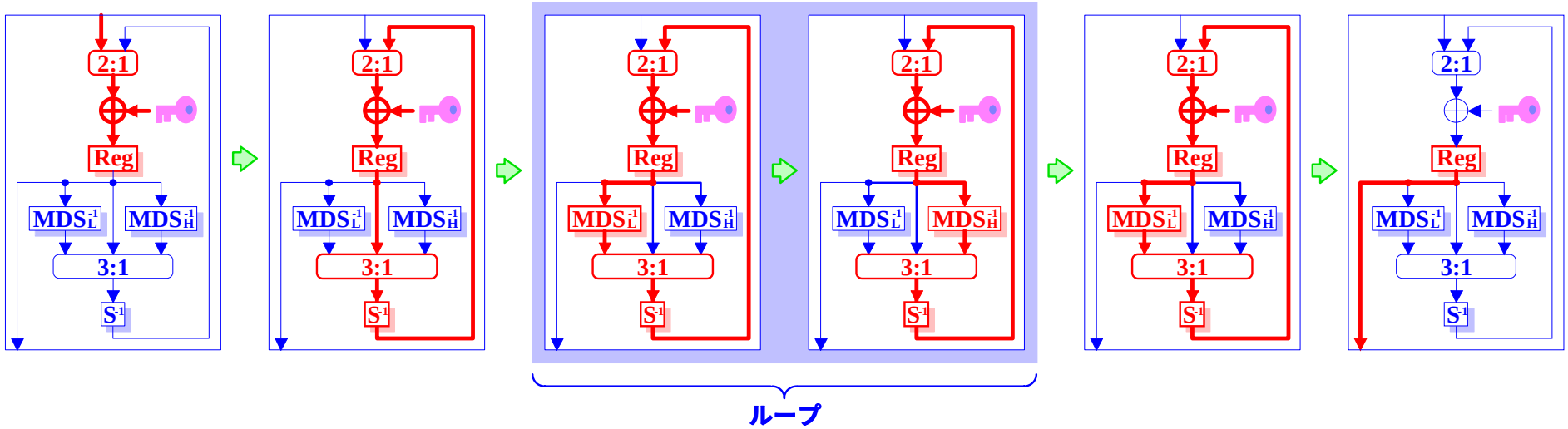
Hierocrypt-3

- ECBモードのデータパス

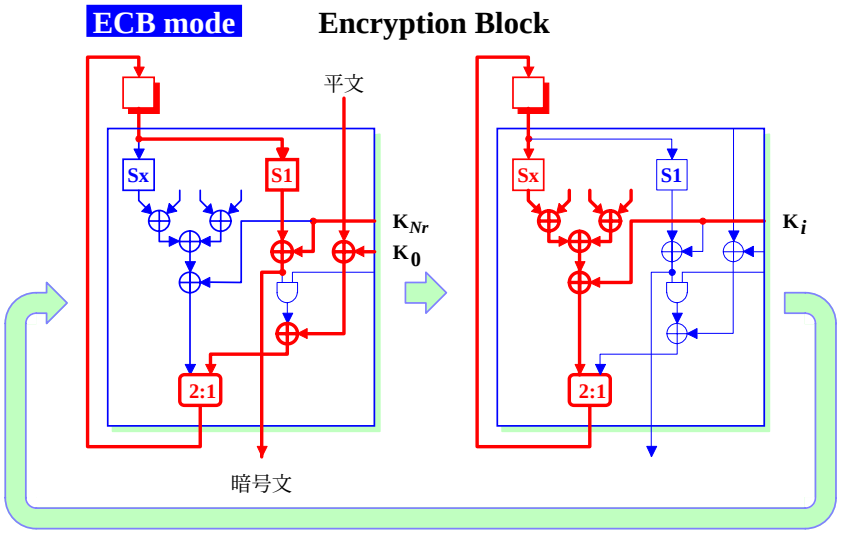
暗号化

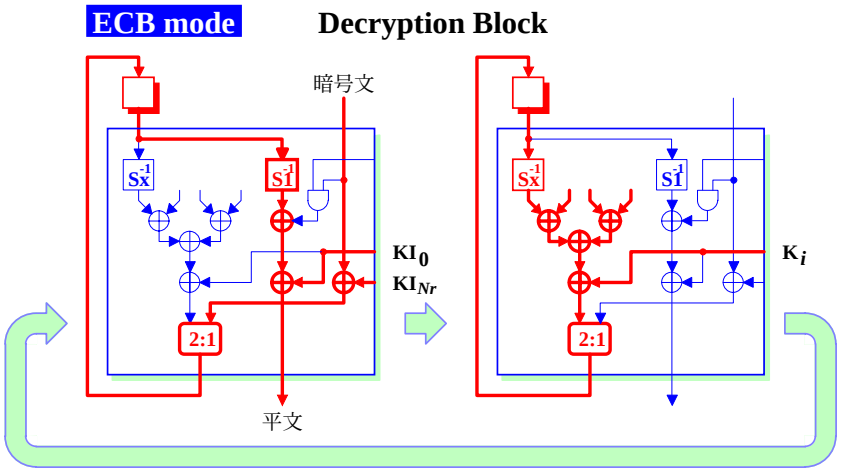


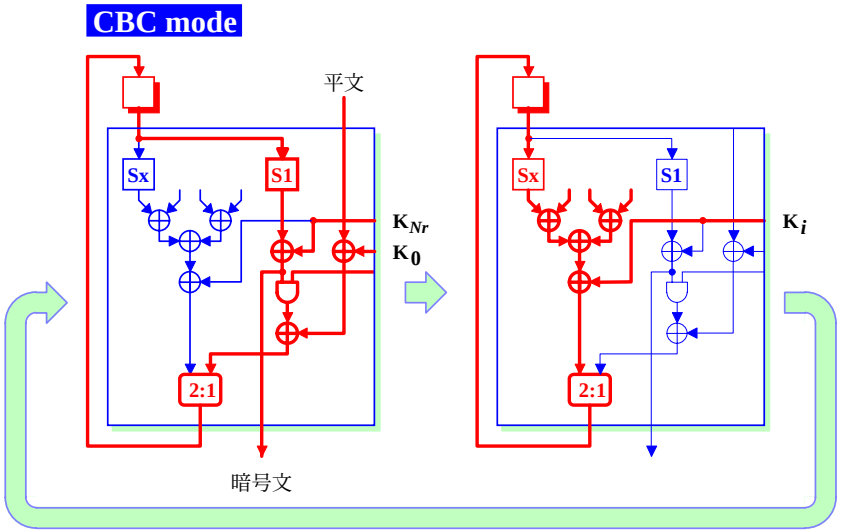
復号化

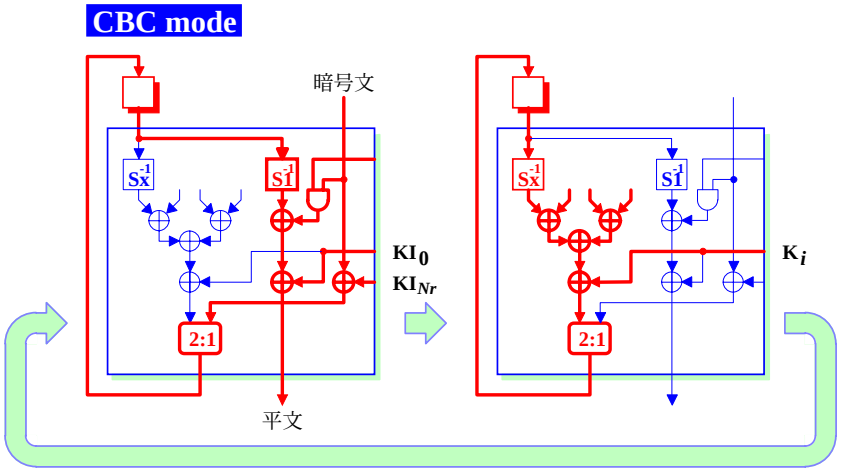


10Gbps Rijndael Architecture

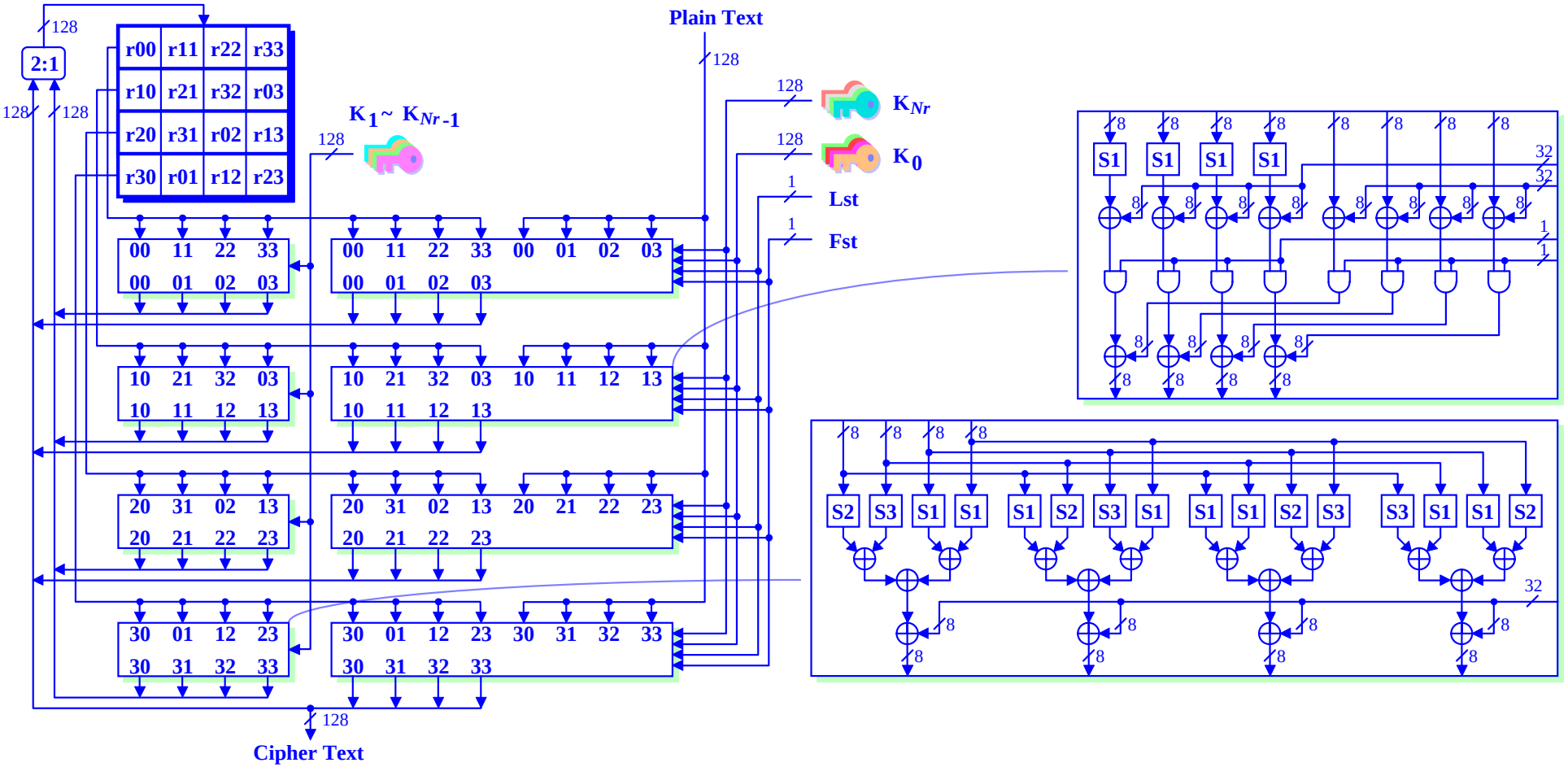








10Gbps Rijndael Architecture



10Gbps Rijndael Architecture