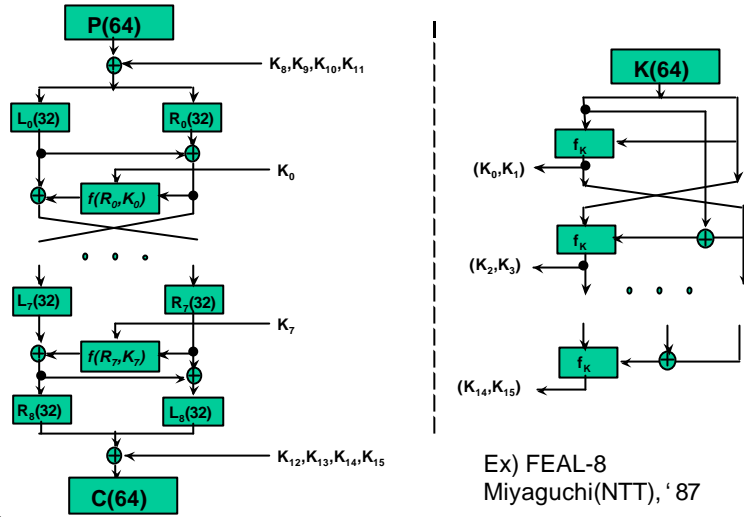


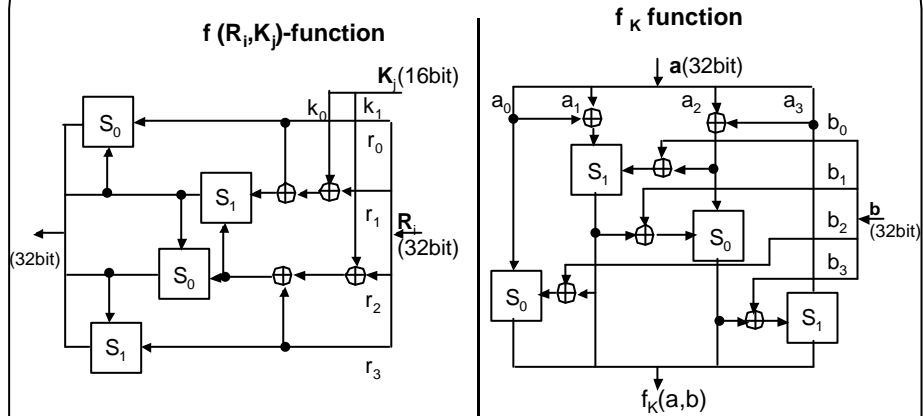
# FEAL(Fast data Encipherment ALg.)



©ICU Kwangjo Kim

1

# FEAL(II)



$S_d(X_1, X_2) = \text{Rot}_2((X_1 + X_2 + d) \bmod 256)$  where,  $d=0$  or  $1$ ,  $X_1$  and  $X_2$  : 8bit,  
 $\text{Rot}_2(Y)$  : 2-bit left rotation of 8-bit  $Y$   
 $K_{2(r-1)}$  : Left half of  $f_K(a,b)$ ,  $K_{2(r-1)+1}$  : right half of  $f_K(a,b)$

©ICU Kwangjo Kim

2

## FEAL(III)

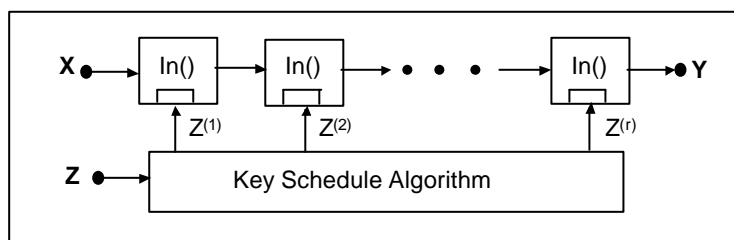
FEAL-n Attack Method	Data Complexity		Storage Complexity	Processing Complexity
	Known	Chosen		
FEAL-4 (LC)	5	-	30Kbytes	6min.
FEAL-6 (LC)	100	-	100Kbytes	40min.
FEAL-8 (LC)	$2^{24}$			10min.
FEAL-8 (DC)		$2^7$ pairs	280Kbytes	2min.
FEAL-16 (DC)		$2^{29}$ pairs		$2^{30}$ op.
FEAL-24 (DC)		$2^{45}$ pairs		$2^{46}$ op.
FEAL-32 (DC)		$2^{66}$ pairs		$2^{67}$ op.

©ICU Kwangjo Kim

3

## Lai's Classification of "E/D-Similar" Iterated Ciphers

(I) Involution Ciphers Only

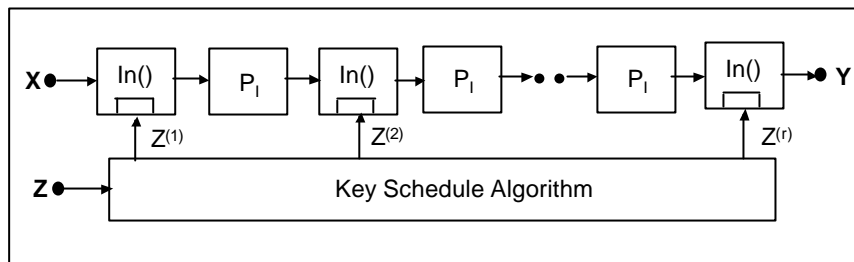


To decrypt : Reverse the Key Schedule

©ICU Kwangjo Kim

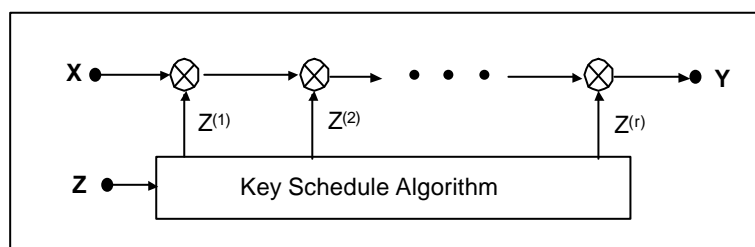
4

(II) Involution Ciphers and Involutory Permutations



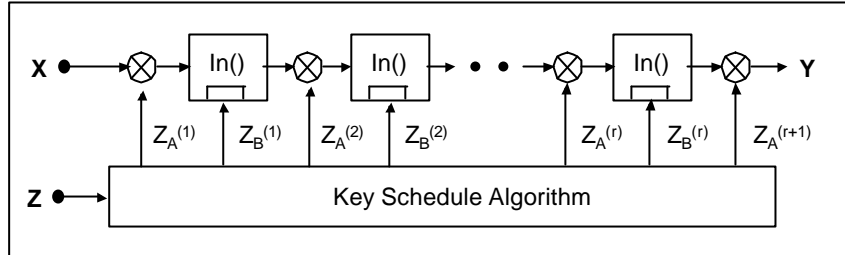
To decrypt : Reverse the Key Schedule  
Ex) DES, FEAL, LOKI etc

\* Group Ciphers Only



To decrypt : Reverse the Key Schedule and replace each subkey by its group inverse.

### (III) Group Ciphers and Involution Ciphers

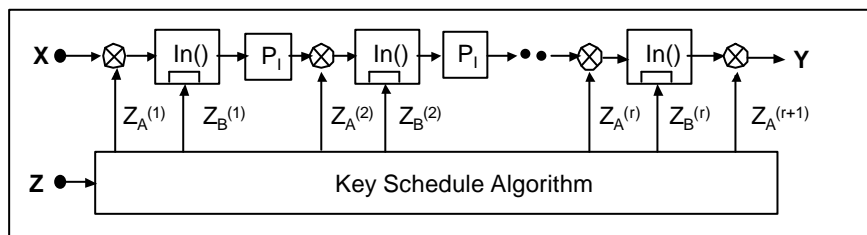
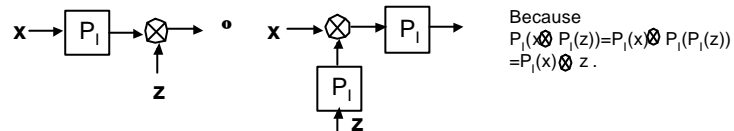


To decrypt : Reverse the Key Schedule and replace each A-key by its group inverse. Ex) PES  
 (\*) Never should be used ...equivalent to one encipherment with  $Z^{(1)} \otimes Z^{(2)} \otimes \dots \otimes Z^{(r)}$

©ICU Kwangjo Kim

7

### (IV) Group Ciphers, Involution Ciphers and Involutionary Permutations such that $P_i(a \otimes b) = P_i(a) \otimes P_i(b)$

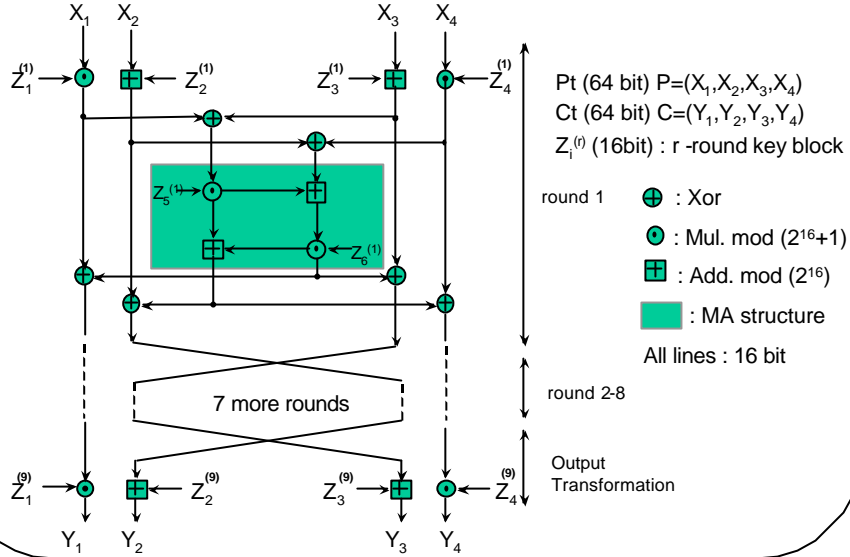


There is no  $P_i$  in the last round.  
 To decrypt : Reverse the Key Schedule and replace  $Z_A^{(1)}$  and  $Z_A^{(r+1)}$  by its group inverse and replace all other A-subkeys by the group inverses of their involutory permuted values [ $P_i(Z^{-1}) = (P_i(Z))^{-1}$ ]. Ex) IDEA

©ICU Kwangjo Kim

8

## IDEA(Int'l Data Enc. Al.)



©ICU Kwangjo Kim

9

## IDEA(II)

- Divide 128-bit key into 52 16-bit word
- (1) Divide 128-bit into 8 blocks and use 6 blocks for round key
  - (2) 25-bit left rotate of 128-bit
  - (3) Repeat steps (1) and (2) till 8-round 8

### Round Key block for Encryption / Decryption

Round	Encryption Key Block	Decryption Key Block
1	$Z_1^{(1)} Z_2^{(1)} Z_3^{(1)} Z_4^{(1)} Z_5^{(1)} Z_6^{(1)}$	$Z_1^{(9)-1} -Z_2^{(9)} -Z_3^{(9)} Z_4^{(9)-1} Z_5^{(8)} Z_6^{(8)}$
2	$Z_1^{(2)} Z_2^{(2)} Z_3^{(2)} Z_4^{(2)} Z_5^{(2)} Z_6^{(2)}$	$Z_1^{(8)-1} -Z_2^{(8)} -Z_3^{(8)} Z_4^{(8)-1} Z_5^{(7)} Z_6^{(7)}$
3	$Z_1^{(3)} Z_2^{(3)} Z_3^{(3)} Z_4^{(3)} Z_5^{(3)} Z_6^{(3)}$	$Z_1^{(7)-1} -Z_2^{(7)} -Z_3^{(7)} Z_4^{(7)-1} Z_5^{(6)} Z_6^{(6)}$
4	$Z_1^{(4)} Z_2^{(4)} Z_3^{(4)} Z_4^{(4)} Z_5^{(4)} Z_6^{(4)}$	$Z_1^{(6)-1} -Z_2^{(6)} -Z_3^{(6)} Z_4^{(6)-1} Z_5^{(5)} Z_6^{(5)}$
5	$Z_1^{(5)} Z_2^{(5)} Z_3^{(5)} Z_4^{(5)} Z_5^{(5)} Z_6^{(5)}$	$Z_1^{(5)-1} -Z_2^{(5)} -Z_3^{(5)} Z_4^{(5)-1} Z_5^{(4)} Z_6^{(4)}$
6	$Z_1^{(6)} Z_2^{(6)} Z_3^{(6)} Z_4^{(6)} Z_5^{(6)} Z_6^{(6)}$	$Z_1^{(4)-1} -Z_2^{(4)} -Z_3^{(4)} Z_4^{(4)-1} Z_5^{(3)} Z_6^{(3)}$
7	$Z_1^{(7)} Z_2^{(7)} Z_3^{(7)} Z_4^{(7)} Z_5^{(7)} Z_6^{(7)}$	$Z_1^{(3)-1} -Z_2^{(3)} -Z_3^{(3)} Z_4^{(3)-1} Z_5^{(2)} Z_6^{(2)}$
8	$Z_1^{(8)} Z_2^{(8)} Z_3^{(8)} Z_4^{(8)} Z_5^{(8)} Z_6^{(8)}$	$Z_1^{(2)-1} -Z_2^{(2)} -Z_3^{(2)} Z_4^{(2)-1} Z_5^{(1)} Z_6^{(1)}$
Output	$Z_1^{(9)} Z_2^{(9)} Z_3^{(9)} Z_4^{(9)}$	$Z_1^{(1)-1} -Z_2^{(1)} -Z_3^{(1)} Z_4^{(1)-1}$

$Z_i^{(i)-1}$  : multiplicative inverse of  $Z_i^{(i)} \text{ mod } (2^{16}+1)$ ,  $-Z_i^{(i)}$  : additive inverse of  $Z_i^{(i)} \text{ mod } 2^{16}$

©ICU Kwangjo Kim

10

## LOKI\*

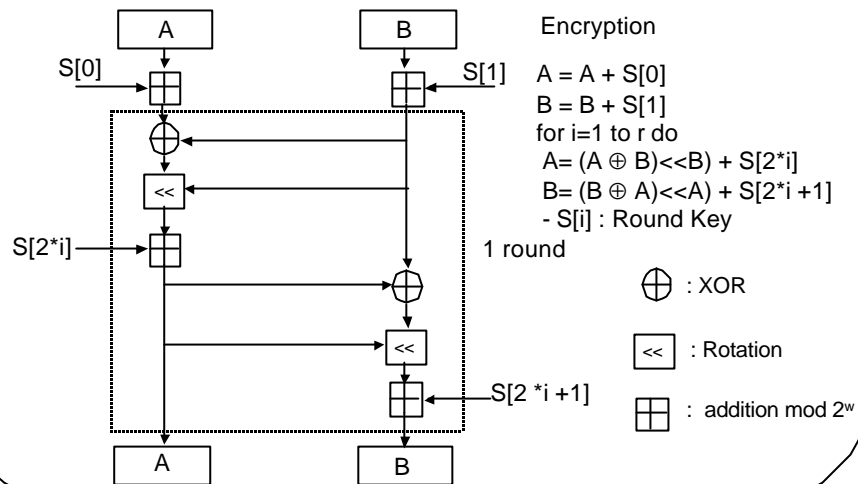
- **Brown, Pieprzyk, Seberry(Australia)**
- **Modify f-function and Key Schedule of DES**
  - **S -box : 12 bit -> 8 bit**
  - **Rotation : 12-bit or 13-bit bit rotation**
  - **Others are same as DES**
- **LOKI89, LOKI91**

\* Name of strong and capricious god appeared in Greek myth

## RC-5(I)

- **Ron Rivest : Ron' s Code**
- **Ease for H/W & S/W Implementation**
- **RC5 - w/r/b**
  - **w : word length (16, 32)**
  - **r : number of round (0-255)**
  - **b : key byte (0 -255)**
  - **Ex) RC5-32/12/16 : 100Kb, 586PC**
- **To prevent weak key, user magic constant (e, p) in key scheduling**

## RC-5(II)



©ICU Kwangjo Kim

13

## GOST

- ❑ Russia: Government Standard 28147-89
- ❑ internal specification(unpublished)
- ❑ 64-bit block cipher, 32 round
- ❑ 256-bit main key & 512-bit subkey
- ❑ No Key Scheduling
- ❑ Modify f-function of DES : 4x4 k-box

©ICU Kwangjo Kim

14

## MISTY

- ❑ Matsui, Ichigawa, Sorimachi, Tokita, Yamagishi in Mitsubishi co.
- ❑ Pt/Ct:64-bit, Key:128-bit, 8 Round
- ❑ Provable security against DC and LC
- ❑ By using recursive structure 30Mb/s (HP9735, PA7150-125MHz)
- ❑ To be KASUMI for 3GPP group

©ICU Kwangjo Kim

15

## Skipjack

- ❑ Classified 64-bit block for replacing -  
> Internal structure published ' 98
- ❑ 80-bit key, 32 round
- ❑ Used for Clipper Project with Key Escrow
- ❑ ECB, CBC, 64 bitOFB, {1/8/16/ 32} bit CFB

©ICU Kwangjo Kim

16



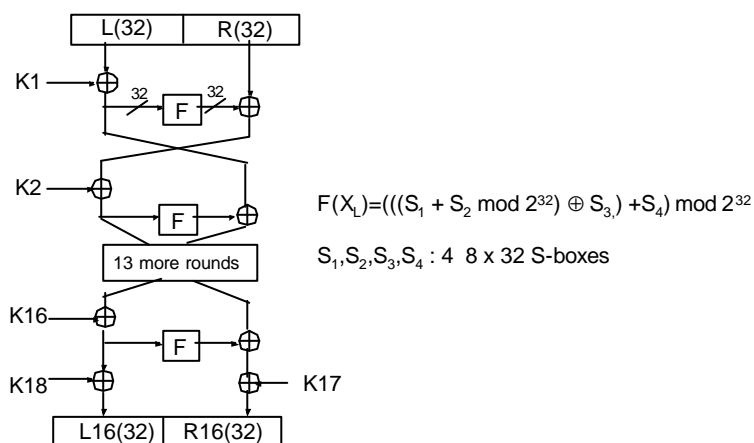
# Blowfish

- ❑ Bruce Schneier, et al
- ❑ XOR, Addition for 32 bit CPU
- ❑ 64bit block, 448 bit key, 8 x 32 S-box
- ❑ 16 round
- ❑ Takes long time for key scheduling

©ICU Kwangjo Kim

17

# Blowfish(II)



©ICU Kwangjo Kim

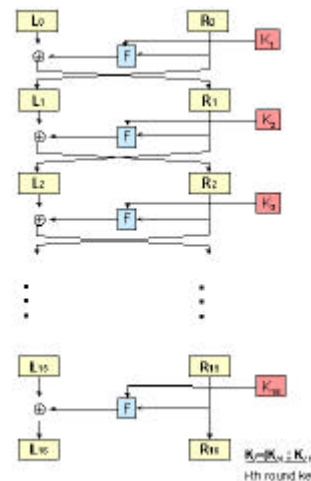
18

# SEED(I)

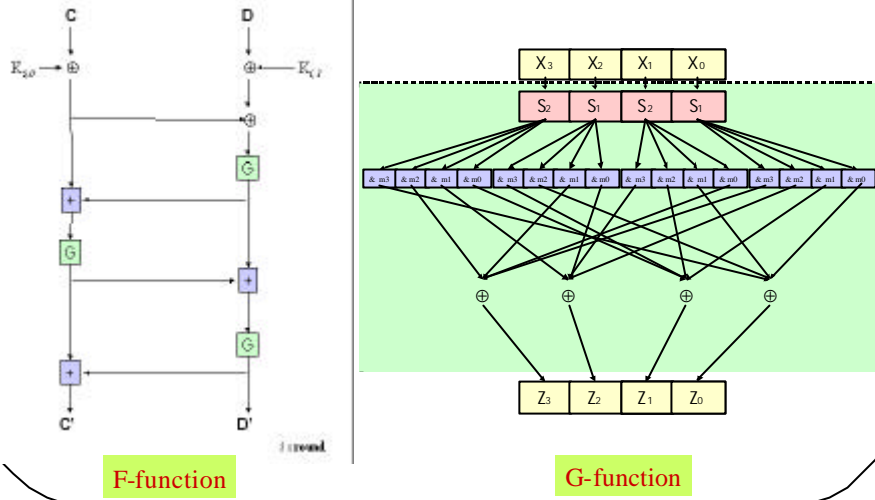
- (Korean Data Encryption Standard)
  - 128-bit symmetric block cipher
  - developed by KISA, ETRI, ADD + a
- Reference
  - KRyptoGate (Korean cRYPTOgraphers' GATEway) <http://www.cryptogate.com>
  - <http://dosan.skku.ac.kr/~sjkim/Kr-Standard.html>

# SEED(II)

- Introduction
  - Symmetric Block Cipher
  - Block size, key size : 128-bit
  - Feistel structure
  - 16 rounds
- Design criteria
  - Provide security proof
  - Secure against DC, LC, High order DC, Related key attack
  - Faster than 3-DES in softwa
  - Use nonlinear function for round-key generation



# SEED(III)



F-function

G-function

©ICU Kwangjo Kim

21

# SEED(IV)

F-function

$$C' = G[G\{G\{(C \oplus K_{10}) \oplus (D \oplus K_{11})\} \oplus (C \oplus K_{10})\} \oplus G\{(C \oplus K_{10}) \oplus (D \oplus K_{11})\}]$$

$$\oplus G[G\{(C \oplus K_{10}) \oplus (D \oplus K_{11})\} \oplus (C \oplus K_{10})]$$

$$D' = G[G\{G\{(C \oplus K_{10}) \oplus (D \oplus K_{11})\} \oplus (C \oplus K_{10})\} \oplus G\{(C \oplus K_{10}) \oplus (D \oplus K_{11})\}]$$

G-function

$$Y_3 = S_2(X_3), \quad Y_2 = S_1(X_2), \quad Y_1 = S_2(X_1), \quad Y_0 = S_1(X_0).$$

$$Z_3 = (Y_0 \& m_3) \oplus (Y_1 \& m_0) \oplus (Y_2 \& m_1) \oplus (Y_3 \& m_2)$$

$$Z_2 = (Y_0 \& m_2) \oplus (Y_1 \& m_3) \oplus (Y_2 \& m_0) \oplus (Y_3 \& m_1)$$

$$Z_1 = (Y_0 \& m_1) \oplus (Y_1 \& m_2) \oplus (Y_2 \& m_3) \oplus (Y_3 \& m_0)$$

$$Z_0 = (Y_0 \& m_0) \oplus (Y_1 \& m_1) \oplus (Y_2 \& m_2) \oplus (Y_3 \& m_3)$$

$$(m_0 = 0xfc, \quad m_1 = 0xf3, \quad m_2 = 0xcf, \quad m_3 = 0x3f)$$

©ICU Kwangjo Kim

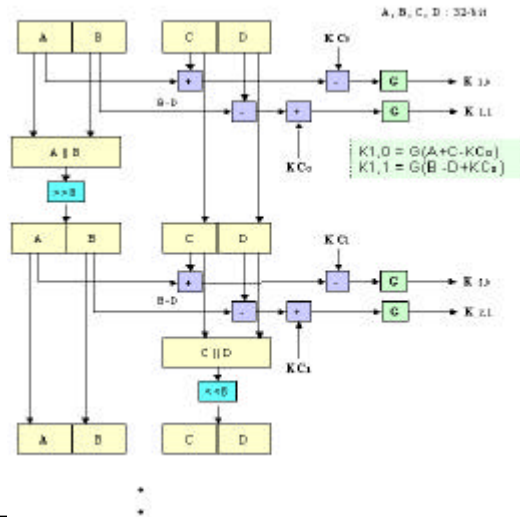
22

# SEED(V)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271
272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287
288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303
304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319
320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335
336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351
352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367
368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383
384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399
400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415
416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431
432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447
448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463
464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479
480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495
496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511
512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527
528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543
544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559
560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575
576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591
592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607
608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623
624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639
640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655
656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671
672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687
688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703
704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719
720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735
736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751
752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767
768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783
784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799
800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815
816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831
832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847
848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863
864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879
880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895
896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911
912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927
928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943
944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959
960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975
976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991
992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007
1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023
1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039
1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055
1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071
1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087
1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103
1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119
1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135
1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151
1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167
1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183
1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199
1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231
1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247
1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263
1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279
1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295
1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311
1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327
1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343
1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359
1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375
1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391
1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407
1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423
1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439

# SEED(VII)

## Round-key generation



©ICU Kwangjo Kim

25

## Others

- Multi2
- SAFER(Secure And Fast Encryption Routine)-64,128
- Lion& Bear
- TEA (Tiny Encryption Algorithm)
- CAST(Carlisle Adams & Stafford Tavares)

©ICU Kwangjo Kim

26

## Summary of block ciphers

Algorithm	Year	Country	Pt/Ct	Key	Round
DES	1977	USA	64	56	16
FEAL	1987	Japan	64	64	4,8,16,32
GOST	1989	Russia	64	256	32
IDEA	1990	Swiss	64	128	8
LOKI	1991	Australia	64	64	16
SKIPJACK	1990	USA	64	80	32
MISTY	1996	Japan	64	128	>8
SEED	1998	Korea	128	128	16

©ICU Kwangjo Kim

27

## AES requirements

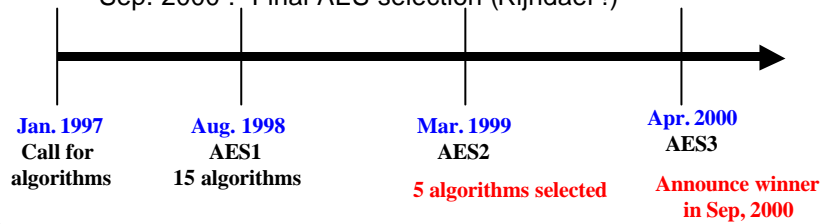
- Block cipher**
  - 128-bit blocks
  - 128/192/256-bit keys
- Worldwide-royalty free**
- More secure than Triple DES**
- More efficient than Triple DES**

©ICU Kwangjo Kim

28

# AES Calendar

- Jan. 2, 1997 : Announcement of intent to develop AES and request for comments
- Sep. 12, 1997 : Formal call for candidate algorithms
- Aug. 20-22, 1998 : First AES Candidate Conference and beginning of Round 1 evaluation (15 algorithms), Rome, Italy
- Mar. 22-23, 1999 : Second AES Candidate Conference, NY, USA
- Sep. 2000 : Final AES selection (Rijndael !)



©ICU Kwangjo Kim

29

# AES1 algorithms

15 algorithms are proposed at AES1 conference

Cipher	Submitted by	Country
CAST-256	Entrust	Canada
Crypton	Future Systems	Korea†
Deal	Outerbridge	Canada†
DFC	ENS-CNRS	France
E2	NTT	Japan
Frog*	TecApro	Costa Rica
HPC*	Schroeppe	USA
LOKI97*	Brown, Pieprzyk, Seberry	Australia
Magenta	Deutsche Telekom	Germany
Mars	IBM	USA†
RC6	RSA	USA†
Rijndael*	Daemen, Rijmen	Belgium‡
Safer+*	Cylink	USA†
Serpent*	Anderson, Biham, Knudsen	UK, Israel, Norway
Twofish*	Counterpane	USA†

\* Placed in the public domain; † and foreign designers; ‡ foreign influence

©ICU Kwangjo Kim

30

## AES Round 2 Algorithms

- After AES2 conference, NIST selected the following 5 algorithms as the round 2 candidate algorithm.

<i>Algorithm Name</i>	<i>Submitter Name(s)</i>
<a href="#">MARS</a>	IBM ( <i>represented by Nevenko Zunic</i> )
<a href="#">RC6™</a>	RSA Laboratories ( <i>represented by Burt Kaliski</i> )
<a href="#">Rijndael</a>	Joan Daemen, Vincent Rijmen
<a href="#">Serpent</a>	Ross Anderson, Eli Biham, Lars Knudsen
<a href="#">Twofish</a>	Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson

## Rijndael(I)

- Proposed by Joan Daemen, Vincent Rijmen(Belgium)
- Design choices
  - **Square type**
  - Three distinct invertible uniform transformations(Layers)
    - ◆ Linear mixing layer : guarantee high diffusion
    - ◆ Non-linear layer : parallel application of S-boxes
    - ◆ Key addition layer : XOR the round key to the intermediate state
  - Initial key addition, final key addition
- Representation of state and key
  - **Rectangular array** of bytes with 4 rows (square type)
  - Nb : number of column of the state
  - Nk : number of column of the cipher key



## Rijndael(II)

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$

State (Nb=6)

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$

Key (Nk=4)

Nr	Nb = 4	Nb = 6	Nb = 8
Nk = 4	10	12	14
Nk = 6	12	12	14
Nk = 8	14	14	14

Number of rounds (Nr)

©ICU Kwangjo Kim

33

## Rijndael(III)

```

Rijndael(State,CipherKey)
{
  KeyExpansion(CipherKey,ExpandedKey);
  AddRoundKey(State,ExpandedKey);
  For( i=1 ; i<Nr ; i++ ) Round(State,ExpandedKey + Nb*i);
  FinalRound(State,ExpandedKey + Nb*Nr);
}

```

```

Round(State,RoundKey)
{
  ByteSub(State);
  ShiftRow(State);
  MixColumn(State);
  AddRoundKey(State,RoundKey);
}

```

```

FinalRound(State,RoundKey)
{
  ByteSub(State);
  ShiftRow(State);
  AddRoundKey(State,RoundKey);
}

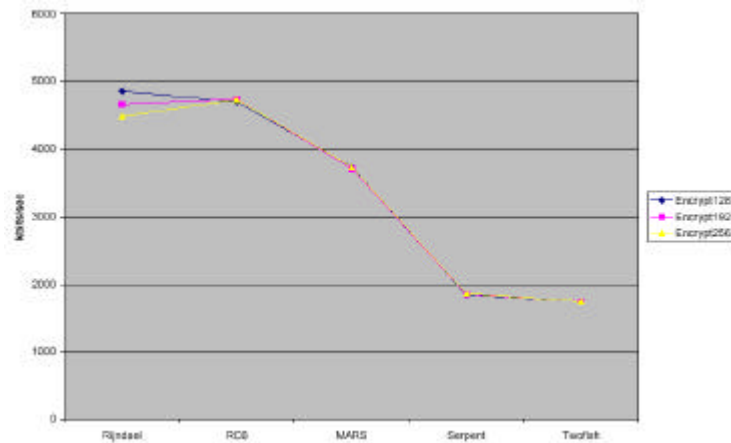
```

©ICU Kwangjo Kim

34

## Comparison of AES2 algorithms(I)

### Encryption speed analysis by NIST



©ICU Kwangjo Kim

35

## Comparison of AES2 algorithms(II)

### Java Implementation by A. Sterbenz(Graz Univ.)

Encryption Speed (kbit/s)	DES (56 bit)	Triple DES (168 bit)	IDEA	MARS	RC6	Rijndael	Serpent	Twofish
128 bit key	10508	4178	12820	19718	26212	19321	11464	19265
192 bit key	n/a	n/a	n/a	19760	26192	16922	11474	19296
256 bit key	n/a	n/a	n/a	19737	26209	14957	11471	19275

Decryption Speed (kbit/s)	DES (56 bit)	Triple DES (168 bit)	IDEA	MARS	RC6	Rijndael	Serpent	Twofish
128 bit key	10519	4173	13018	19443	24338	18968	11519	18841
192 bit key	n/a	n/a	n/a	19670	24382	16484	11514	18841
256 bit key	n/a	n/a	n/a	19489	24279	14468	11533	18806

Encryption Key Setup (keys/s)	DES (56 bit)	Triple DES (168 bit)	IDEA	MARS	RC6	Rijndael	Serpent	Twofish
128 bit key	18128	5150	90571	28680	45603	96234	34729	13469
192 bit key	n/a	n/a	n/a	27928	40625	86773	33516	10556
256 bit key	n/a	n/a	n/a	26683	29069	70494	31973	8500

Decryption Key Setup (keys/s)	DES (56 bit)	Triple DES (168 bit)	IDEA	MARS	RC6	Rijndael	Serpent	Twofish
128 bit key	18039	5136	20737	28743	45709	56017	34687	13469
192 bit key	n/a	n/a	n/a	27917	40625	48324	33560	10550
256 bit key	n/a	n/a	n/a	26731	39028	39963	31973	8531

©ICU Kwangjo Kim

36

## Comparison of AES2 algorithms(III)

□ Smart Card Implementation by F. Sano (Toshiba)

Cipher	RAM (bytes)		ROM (bytes)		Time (clock)						
					Encrypt	Schedule	Encrypt + Schedule				
MARS	572	5	5,468	2	45,588	4	21,742	2	67,330	3	*
RC6	156	3	1,060	2	34,736	3	138,851	4	173,587	4	only encryption
Rijndael	66	1	980	1	25,494	1	10,318	1	35,812	1	
Serpent	164	4	3,937	4	71,924	5	147,972	5	219,896	5	
Twofish	90	2	2,808	3	31,877	2	28,512	3	60,389	2	
DES	17		772						25,398		
Triple DES	17		849						72,341		
MISTY	44		1,598						25,486		

\* : omit to check "weak" in the key schedule

©ICU Kwangjo Kim

37

## Comparison of AES2 algorithms(IV)

□ CMOS ASIC Implementation by Ichikawa (Mitsubishi)

**Table 4.1 Hardware evaluation results**

Algorithm name	area [Gate]			Key setup time[ns]	Critical-path[ns]	Throughput [Mbps]
	Encryption & Decryption	Key Schedule	Total			
DES	42,204	12,201	54,405	-	55.11	1161.31
Triple-DES	124,888	23,207	148,147	-	157.09	407.4
MARS	690,654	2,245,096	2,935,754	1740.99	567.49	225.55
RC6	741,641	901,382	1,643,037	2112.26	627.57	203.96
Rijndael	518,508	93,708	612,214	57.39	65.64	1950.03
Serpent	298,533	205,096	503,770	114.07	137.4	931.58
Twofish	200,165	231,682	431,857	16.38	324.8	394.08

©ICU Kwangjo Kim

38