Crypto Report - 1

Team Anonymous:
CS14B023, Rahul Kejriwal
CS14B039, Bikash Gogoi

Introduction

One of the drawbacks of AES and DES are their static structures which enable easy application of attacks like Linear Cryptanalysis and Differential Cryptanalysis which reduce the attack complexity from 2^{56} for DES and from 2^{128} for AES.

We propose a sort of dynamic design of the cipher system where the type of each round is also a function of the key. We use a hybrid design with rounds of both SPN and Fiestel cipher systems. The input key decides which round will be of which type. Thus, for a n-round design of our system, we effectively have 2^n different structures of the cipher system. Any attack would have to at least analyze all the 2^n structures for linear/differential trails to apply linear/differential cryptanalysis. Even so, the attacker is left with 2^n different possibilities after attacking all of the possible structures (although the complexity of each attack on a possible cipher structure decreases since only a subset of the keyspace has to be searched for any given possible structure). The design intends to disrupt simple application of Linear/Differential cryptanalysis and tries to maintain the security at 2^k where k is the number of key bits.

Report Questions

Q1. What is your cipher called?

A1. Our cipher is called DHE - 1 short for Dynamic Hybrid Encryption - Mach 1 (dynamic since the cipher structure changes w.r.t. key).

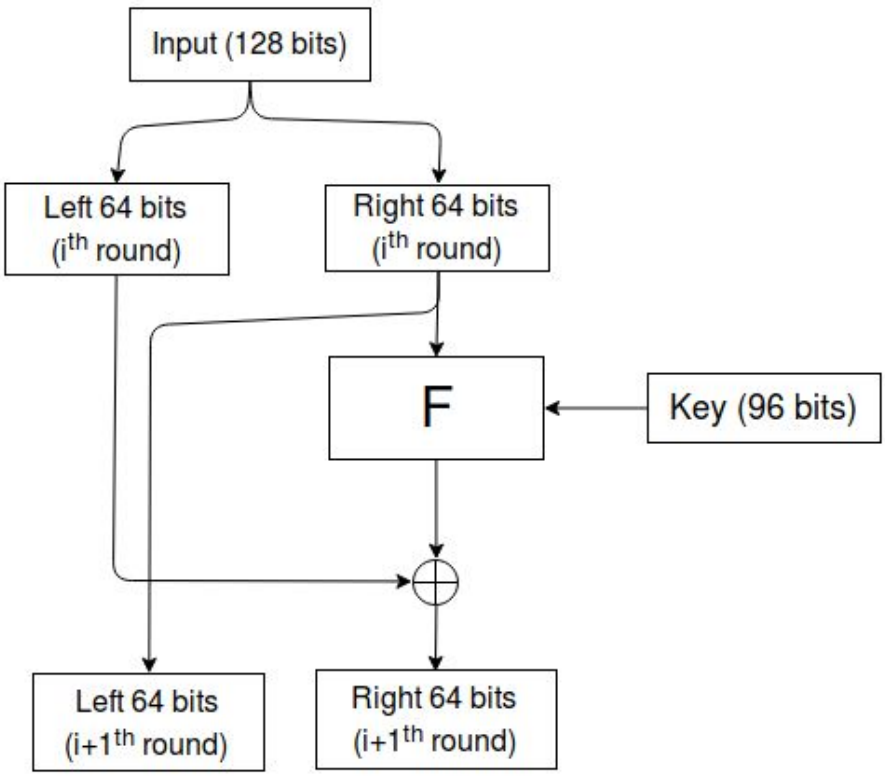
Q2. How many rounds does your cipher have?

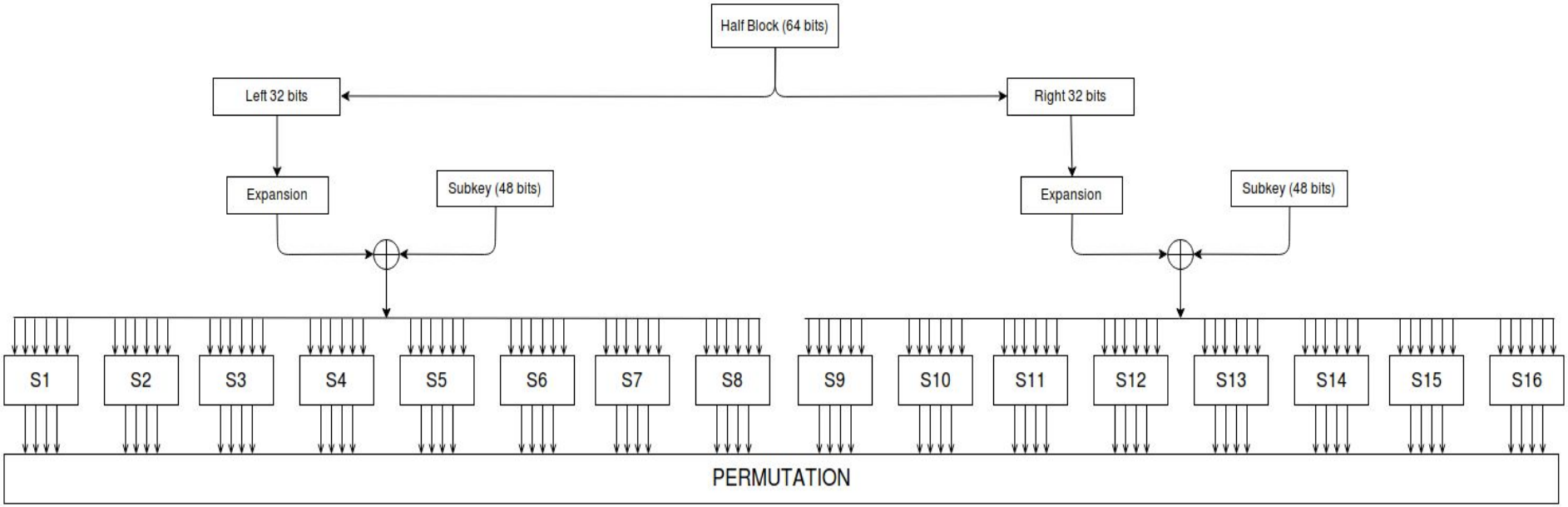
A2. Our cipher has 8 rounds.

Q3. Explain one round of your cipher. You can assume that key expansion is done by a separate process. Don't need to explain that. You can assume that your encryption function takes an array of N keys, where N is the number of rounds in your cipher.

A3. Our cipher has a (sort of) dynamic structure. It consists of 2 types of rounds - one is an SPN round and the other is a Fiestel round. Key is first hashed using MD5. Then the hashed value is divided into 16 groups with 8 bits in each group. Calculate parity of each group. If the parity of i^{th} group is odd then the round is Fiestel network, else SPN network.

1. Fiestel Type Round:

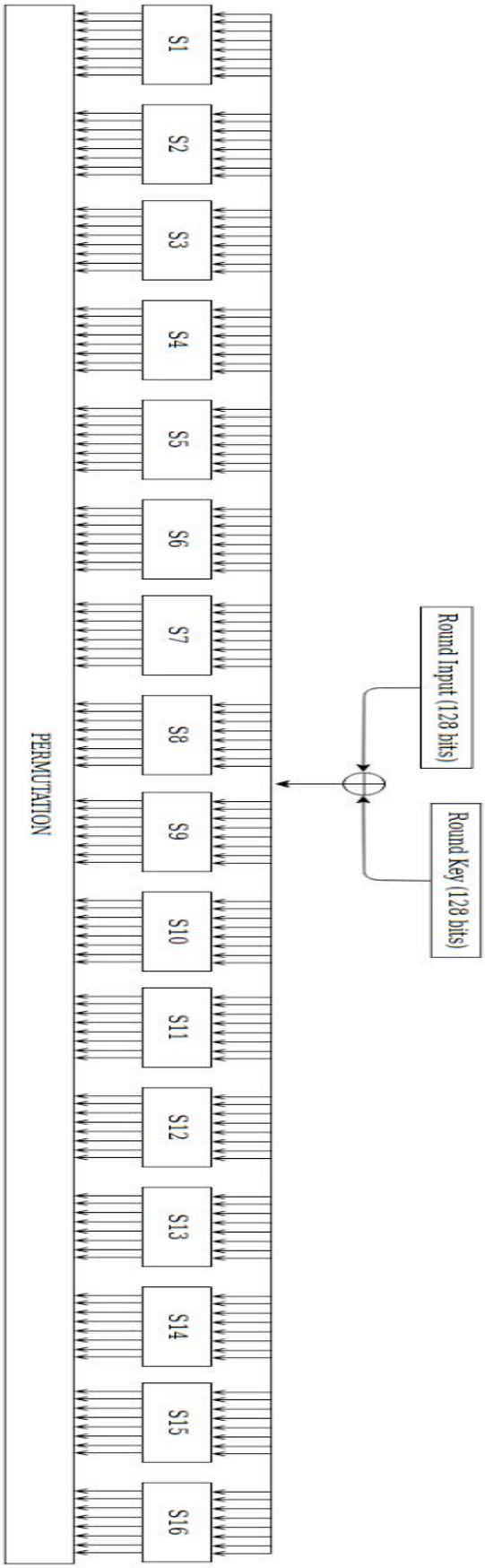



Half Block (64 bits)
Left 32 bits
Right 32 bits
Expansion
Subkey (48 bits)
Expansion
Subkey (48 bits)
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
PERMUTATION

Expansion Function

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

2. SPN Round:


S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
PERMUTATION
Round Input (128 bits)
Round Key (128 bits)

Q4. How did you arrive at the number of rounds for your cipher?

A4. We calculate minimum number of rounds required to reduce worst case bias at end of a linear trail to below 2^{-128} .

For 6x4 S-boxes: (complete diffusion in 3 rounds)

	S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**
Max Bias	0.21875	0.18750	0.21875	0.18750	0.21875	0.18750	0.21875	0.21875	0.21875	0.21875	0.21875	0.21875	0.18750	0.21875	0.21875	0.25000

Max bias across all 6x4 s-boxes = 0.25
Avg bias across all 6x4 s-boxes = 0.21289

For 8x8 S-boxes: (complete diffusion in 2 rounds)

	S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**
Max Bias	0.140625	0.140625	0.140625	0.140625	0.125000	0.132813	0.140625	0.132813	0.148438	0.132813	0.132813	0.140625	0.140625	0.140625	0.140625	0.132813

Max bias across all 8x8 s-boxes = 0.1484375
Avg bias across all 8x8 s-boxes = 0.13769531

Due to the hybrid structure, in the worst case we achieve complete diffusion in 3 rounds (see last answer).

$$min.\ no\ of\ activated\ sboxes = log(2^{-128}) / log(0.25) = 64$$

So, $$(min.\ no\ of\ rounds - 3) * 16 + 3 = 64$$

Since in the first 3 rounds, we consider (conservatively) only one active s-box, and in remaining all s-boxes are activated.
Thus,

$$min.\ no.\ of\ rounds = 6.8125$$

Which rounds off to 7 rounds minimum. We add an extra round for safety making a total of 8 rounds.

Q5. How did you choose the s-boxes size and type?

A5. We use a 8x8 invertible straight s-box for the SPN round and 6x4 non-invertible compression s-box for the Fiestel rounds. The dimensions of the s-box were so chosen to enable discovery of s-box using Genetic Optimization Algorithms within reasonable time.

For input sizes higher than 8, we were getting very slow iterations of Genetic Optimization. Also, with 8 input s-box for SPN rounds, we can effectively use 16 such boxes for each round. They also partly draw inspiration from s-box sizes of AES and DES.

Q6. How did you go about choosing the s-box(es) mappings?

A6. The S-Box mappings were discovered using Genetic Optimization Algorithms.

We enforced balancedness (strictly) on all the output bits of the s-box and used non-linearity of the output bits as a measure of fitness of the individual s-box. We gave higher weightage to lower nonlinearity output bits of s-box in order to ensure high minimum nonlinearity of outputs.

We used a variant of ordered crossover and mutation as swaps between images of two input values.

We ran the optimization for around 100 iterations where we practically reached saturation.

Q7. For your s-box(es), explain / demonstrate how you satisfied the following properties:

1. **The balancedness property**
2. **SAC**
3. **Non-linearity**
4. **Algebraic degree (optional)**

A7. Since there is a tradeoff among the first 3 properties, we decided to optimize our s-boxes based on balancedness and non-linearity.

1. We strictly enforce all s-box output bits to be balanced.
2. We did not consider SAC and favored non-linearity as the s-box design criteria for optimization.
3. We optimize on non-linearity using genetic optimization with higher weightage on the least non-linear bits of s-box to ensure high minimum non-linearity of the least non-linear bit.

Q8. Draw the linear approximation table for your s-box(es).

A8. PFA files containing information regarding s-boxes on mapping, LAT tables and DDT tables.

Q9. Draw the differential distribution table for your s-box(es).

A9. PFA files containing information regarding s-boxes on mapping, LAT tables and DDT tables.

Q10. How did you choose the diffusion layer for your cipher?

A10.

Permutation for spn round:

The values in the table contains the mapping from output bit of n^{th} round to input bit of $n+1^{th}$ round.

	S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**
bit 1	1	114	99	84	69	54	39	24	9	122	107	92	77	62	47	31
bit 2	17	2	115	100	85	70	55	40	25	10	123	108	93	78	63	48
bit 3	33	18	3	116	101	86	71	56	41	26	11	124	109	94	79	64
bit 4	49	34	19	4	117	102	87	72	57	42	27	12	125	110	95	80
bit 5	65	50	35	20	5	118	103	88	73	58	43	28	13	126	111	96
bit 6	81	66	51	36	21	6	119	104	89	74	59	44	29	14	127	112
bit 7	97	82	67	52	37	22	7	120	105	90	75	60	45	30	15	128
bit 8	113	98	83	68	53	38	23	8	121	106	91	76	61	46	31	16

Permutation for fiestel round:

The values in the table contains the mapping from output bit of n^{th} round s-boxes to output bit of F function of n^{th} round.

	S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**
bit 1	1	50	35	20	5	54	39	24	9	58	43	28	13	62	47	32
bit 2	17	2	51	36	21	6	55	40	25	10	59	44	29	14	63	48
bit 3	33	18	3	52	37	22	7	56	41	26	11	60	45	30	15	64
bit 4	49	34	19	4	53	38	23	8	57	42	27	12	61	46	31	16

The mappings chosen for the diffusion layer are circular.
Let say i^{th} output bit of sbox S_k is mapped to j^{th} input bit of sbox S_l of next round. Then $i+1^{th}$ bit of sbox S_k will be mapped to j^{th} output bit of S_{l+1}. If j^{th} output bit is already mapped, then it is mapped to $j+1^{th}$ bit of S_{l+1}. So each output bits is mapped to different but consecutive sboxes.
If a input bit of sbox is changed, it's effect will be seen in the maximum possible sboxes in next round. Hence, this mapping forces one bit changes to diffuse quickly.

Q11. How many rounds would it take to obtain complete diffusion? Show minimum rounds needed for diffuse completely when the i-th bit in the plaintext is toggled, where i can vary from 0 to 127.

A11.

SPN rounds:

Each change in input bit of a sbox gets diffused to 8 output bits in the first round. In second round those 8 bits will get fed into 8 consecutive sboxes. So at the end of second round, 64 bits of the 8 consecutive sboxes will get affected. The permutation layer used in this cipher has a property that output of any two consecutive sboxes feds all the 16 sbox. So at the end of third round, all the output bits will get affected.

Fiestel rounds:

In first round, 1 input bit change in a sbox gets diffused to 4 output bits. This 4 output bits will get fed into four consecutive sboxes in second round. At the end of second round, 16 bits of 4 consecutive sboxes gets affected. The permutations used here has a property that output of four consecutive sboxes feeds 16 different sbox. Since we have total of 16 sboxes, so at the end of third round all the output bits of the 16 sboxes will get affected.

In both Fiestel and SPN network, 3 rounds are required for complete diffusion. So in this hybrid cipher, max rounds for complete diffusion is 3.