Analysis of KLEIN cipher

iterators

Department of EECS
Indian Institute of Technology Bhilai

November 28, 2020

Outline

1. Introduction
2. Cipher Specifications
3. Observations
4. Brownie Point Nominations
5. Conclusion

Introduction

1. As the development and usage of wireless computing and embedded systems is increasing we are being increasingly dependent on Ubiquitous computing examples are sensors, RFID tags etc.
2. On these limited resource systems the selection of the security algorithms must be done carefully by taking the implementation costs along with the level of security provided into consideration.
3. Many algorithms with various design strategies were proposed.Few of them were skipjack, KATAN, KTANTAN, PRESENT etc.
4. Before wide implementation of a security algorithm, it should be thoroughly analysed.

Introduction

5. As the result of these analysis an attack on 31 out of 32 rounds of skipjack based on impossible differential is discovered. Also there are weak key attacks and linear attacks on PRESENT.
6. KLEIN is a lightweight block cipher which is mainly invented for devices like sensors which have very less resources.
7. KLEIN is based on Substitution Permutation Networks.

Outline

1. Introduction
2. Cipher Specifications
3. Observations
4. Brownie Point Nominations
5. Conclusion

Design rationale

1. KLEIN is collection of ciphers of different key length(64, 80, and 96) and fixed 64 bit block length.We shall denote these ciphers as KLEIN-64/80/96 based on their key lengths.
2. We know that block cipher's security and implementation cost mainly depends on the key size and block size. Keeping in mind that lightweight ciphers are used in resource constrained machines like sensors, RFIDs, the block length is decided to be 64 bits as high-throughput is not expected in these devices as large block lengths and larger keys are unnecessary.
3. As 64 bit key length might be little vulnerable if we consider attacks based on pre-computation and large storage capacities, it is suggested to use KLEIN-64 for message authentication codes and hash functions.

Optimal Platform

1. Generally light weight ciphers are optimized for hardware implementation as they are used in RFID tags and smart cards.
2. But if a system can support the computation and memory requirements of the software implementation, the costs of manufacturing and maintenance will reduce drastically as we can simply update the implementation of the cipher by simply installing a software update.
3. Software implementation is more flexible.
4. Both the software and hardware implementations are lightweight.

Structure of KLEIN

1. KLEIN is made of Substitution-Permutation-Network(SPN) which is also used in popular ciphers like AES and PRESENT.
2. By taking into the consideration of security margin and the asymmetric iteration we chose 12/16/20 rounds for KLIEN-64/80/96 respectively.

Algorithm 1: KLEIN CIPHER

```
$sk^1 \leftarrow KEY$;
STATE ← PLAINTEXT;
for $i = 1$ to $N_R$ do
    $AddRoundKey(\text{STATE}, ski)$;
    $SubNibbles(\text{STATE})$;
    $RotateNibbles(\text{STATE})$;
    $MixNibbles(\text{STATE})$;
    $sk_{i+1} = KeySchedule(sk_i, i)$;
end
CIPHERTEXT ← $AddRoundKey(\text{STATE}, sk^{N_R+1})$;
```

Round Transformation

SubNibbles

Before this step, the corresponding round key will be xor-ed with the input . The obtained resultant state is passed to subnibbles where the state is divided into 16 4-bit nibbles and given as input to the 4 x 4 Involutive permutation.

Involutiveness of S-box is helpful to decrease the implementation costs of calculating its inverse and also the need of protect only one sbox instead of two(original and inverse) in other ciphers from side channel attacks.

Input	0	1	2	3	4	5	6	7	8	9	*A*	*B*	*C*	*D*	*E*	*F*
Output	7	4	*A*	9	1	*F*	*B*	0	*C*	3	2	6	8	*E*	*D*	5

It also satisfies following properties:

1. It should not contain any fixed points i.e S(x)≠x, where $x \in \mathbb{F}_2^4$.
2. For any non-zero input diff(Δ_I) and output diff(Δ_O) that belong to $\mathbb{F}_2^4$, It should follow:

$$\#\{x \in \mathbb{F}_2^4 | S(x) \oplus S(x \oplus \Delta_I) = \Delta_O\} \leq 4$$

. Furthermore, if $wt(\Delta_I) = wt(\Delta O) = 1$, where $wt(X) = \sum_i X_i$ (X_i is ith bit of X) , we have

$$\#\{x \in \mathbb{F}_2^4 | S(x) \oplus S(x \oplus \Delta_I) = \Delta_O\} \leq 2$$

.

Rotate Nibbles

1. Assume at the ith round state is $b_0 b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10} b_{11} b_{12} b_{13} b_{14} b_{15}$ where b_i is a nibble. The rotate nibble step involves left circular shift of two bytes of this state i.e after the rotate nibbles step our state becomes $b_4 b_5 b_6 b_7 b_8 b_9 b_{10} b_{11} b_{12} b_{13} b_{14} b_{15} b_0 b_1 b_2 b_3$

Mix nibbles

1. Let the current state be
   $c_0 c_1 c_2 c_3 c_4 c_5 c_6 c_7 c_8 c_9 c_{10} c_{11} c_{12} c_{13} c_{14} c_{15}$
2. This state is divided into two parts
   $c_0 c_1 c_2 c_3 c_4 c_5 c_6 c_7$ and $c_8 c_9 c_{10} c_{11} c_{12} c_{13} c_{14} c_{15}$
3. Each two nibble pair acts as a byte and each of the above part form a column. Then the multiplication process is same as that of one column of AES.

$$\begin{bmatrix} s_0^{i+1} | s_1^{i+1} \\ s_2^{i+1} | s_3^{i+1} \\ s_4^{i+1} | s_5^{i+1} \\ s_6^{i+1} | s_7^{i+1} \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} c_0 | c_1 \\ c_2 | c_3 \\ c_4 | c_5 \\ c_6 | c_7 \end{bmatrix} ,$$

$$\begin{bmatrix} s_8^{i+1} | s_9^{i+1} \\ s_{10}^{i+1} | s_{11}^{i+1} \\ s_{12}^{i+1} | s_{13}^{i+1} \\ s_{14}^{i+1} | s_{15}^{i+1} \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} c_8 | c_9 \\ c_{10} | c_{11} \\ c_{12} | c_{13} \\ c_{14} | c_{15} \end{bmatrix}$$

Thus we obtain the next state

$s_0^{i+1} s_1^{i+1} s_2^{i+1} s_3^{i+1} s_4^{i+1} s_5^{i+1} s_6^{i+1} s_7^{i+1} s_8^{i+1} s_9^{i+1} s_{10}^{i+1} s_{11}^{i+1} s_{12}^{i+1} s_{13}^{i+1} s_{14}^{i+1} s_{15}^{i+1}$

Key Schedule

- The first subkey sk_0 is same as the master key $sk_0{=}mk$.Each of the subsequent sk_{i+1} will be derived from sk_i as follows -
- Denote sk_i as a tuple of bytes - (x0 x1 x2 x3 x4 x5 x6 x7) Divide the tuple into two equal parts and call them a and b
  a - $(x_0\ x_1\ x_2\ x_3)$
  b - $(x_4\ x_5\ x_6\ x_7)$
- Now Perform one byte left circular shift to both a and b
  $a' = (x_1\ x_2\ x_3\ x_0)$
  $b' = (x_7\ x_4\ x_5\ x_6)$

- Swap a' and b' i.e $a'' = b'$ and b"=a'
- Now let $a'' = (y_0\ y_1\ y_2\ y_3)$
  and $b'' = (z_0\ z_1\ z_2\ z_3)$
  We will Xor the round counter i with 3rd byte of a'' and pass 2nd and 3rd byte of b'' through the KLIEN S-BOX and then $a''||b''$ will become the next subkey
  $sk_{i+1} =$ (y0 y1 $y2 \oplus R_i$ y3 z0 Sbox(z1) Sbox(z2) z4)

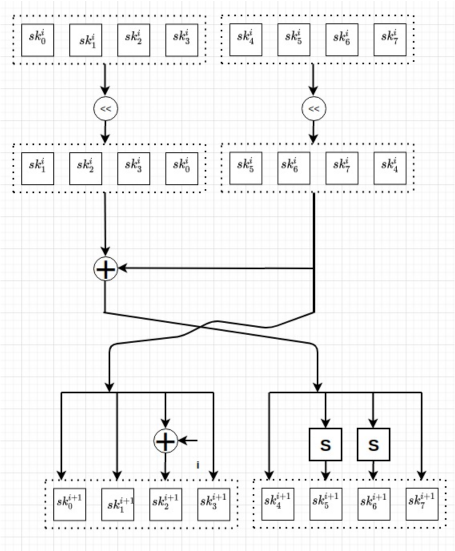


Figure: Key Schedule

Overview of KLEIN Cipher

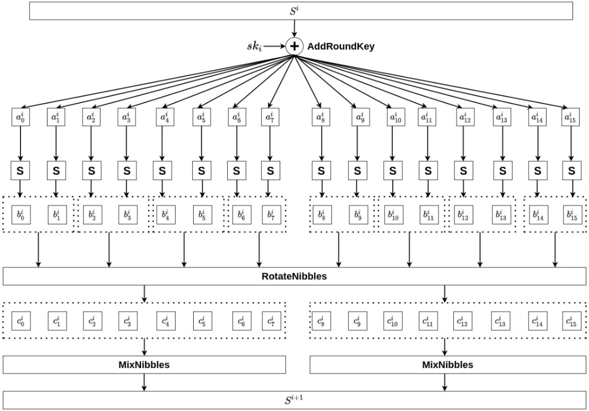

S^i
sk_i
AddRoundKey
a_0^i a_1^i a_2^i a_3^i a_4^i a_5^i a_6^i a_7^i a_8^i a_9^i a_{10}^i a_{11}^i a_{12}^i a_{13}^i a_{14}^i a_{15}^i
S
b_0^i b_1^i b_2^i b_3^i b_4^i b_5^i b_6^i b_7^i b_8^i b_9^i b_{10}^i b_{11}^i b_{12}^i b_{13}^i b_{14}^i b_{15}^i
RotateNibbles
c_0^i c_1^i c_2^i c_3^i c_4^i c_5^i c_6^i c_7^i c_8^i c_9^i c_{10}^i c_{11}^i c_{12}^i c_{13}^i c_{14}^i c_{15}^i
MixNibbles
MixNibbles
S^{i+1}

Figure: KLEIN cipher

Outline

1. Introduction
2. Cipher Specifications
3. Observations
4. Brownie Point Nominations
5. Conclusion

The DDT(Differential Distribution Table) of SBOX is written below:

$\Delta_{In} \vert \Delta_{Out}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	4	2	0	2	0	2	0	0	2	0	0	2	2
2	0	0	0	0	0	4	0	0	0	0	2	2	0	4	2	2
3	0	4	0	2	2	0	0	0	0	0	2	0	0	2	4	0
4	0	2	0	2	2	0	2	0	0	2	0	2	0	2	0	2
5	0	0	4	0	0	2	0	2	2	0	2	4	0	0	0	0
6	0	2	0	0	2	0	4	0	2	0	2	2	2	0	0	0
7	0	0	0	0	0	2	0	2	2	2	0	0	2	0	4	2
8	0	2	0	0	0	2	2	2	2	2	0	2	0	0	0	2
9	0	0	0	0	2	0	0	2	2	0	0	2	2	2	2	2
10	0	0	2	2	0	2	2	0	0	0	4	0	2	0	2	0
11	0	2	2	0	2	4	2	0	2	2	0	0	0	0	0	0
12	0	0	0	0	0	0	2	2	0	2	2	0	4	2	0	2
13	0	0	4	2	2	0	0	0	0	2	0	0	2	2	0	2
14	0	2	2	4	0	0	0	4	0	2	2	0	0	0	0	0
15	0	2	2	0	2	0	0	2	2	2	0	0	2	2	0	0

The maximum differential probability is KLEIN's SBOX is 4/16 i.e 1/4 and some of the transitions leading to it are $(\Delta_{In}, \Delta_{Out})=$ (1,4),(3,1) etc.

Attacks on KLEIN cipher

Round Reduced Attack

The weakness present in the Rotate Nibbles and Mix columns step is exploited here in this attack.

Firstly, a 6 round truncated differential distinguisher with 2^{-29} is made.Using this as base, an 8 Round distinguisher is constructed.One of the assumption is they will be having access to round wise outputs.

Lets have look at few terminologies used in this attack:

1. X_i : The input of the i-th round.
2. ΔX_i : The input difference of the i-th round.
3. Y_i : The input of SubNibbles in the i-th round .
4. ΔY_i : The input difference of SubNibbles in the i-th round .
5. $X_{i,j}$: The j-th nibble of the X_i , where j = 0, 1, ...15.
6. sk_i : The subkey of the i-th round.
7. $X||Y$: The the concatenation of X and Y.

The state of encryption:

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

The number in box represents the Nibble numbers of the 64 bit data block. **Some properties and observations of KLEIN cipher**

Lemma1. If a byte is of the form $0z$, where z is a 4-bit string with MSB bit as 0, then $0z$ multiply by x is equal to $0z^{'}$, where $z^{'}$ is a 4-bit string.

The following oservations are derived based on above lemma.

Observation 1. $\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \text{x} \begin{bmatrix} 0z \\ 00 \\ 00 \\ 00 \end{bmatrix} = \begin{bmatrix} 0z'_1 \\ 0z'_2 \\ 0z'_3 \\ 0z'_4 \end{bmatrix}$ if only if MSB z is 0.

Observation 2. $\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \text{x} \begin{bmatrix} 00 \\ 00 \\ 0z_1 \\ 0z_2 \end{bmatrix} = \begin{bmatrix} 0z'_1 \\ 0z'_2 \\ 0z'_3 \\ 0z'_4 \end{bmatrix}$ if only if MSB z_1 and z_2 is 0.

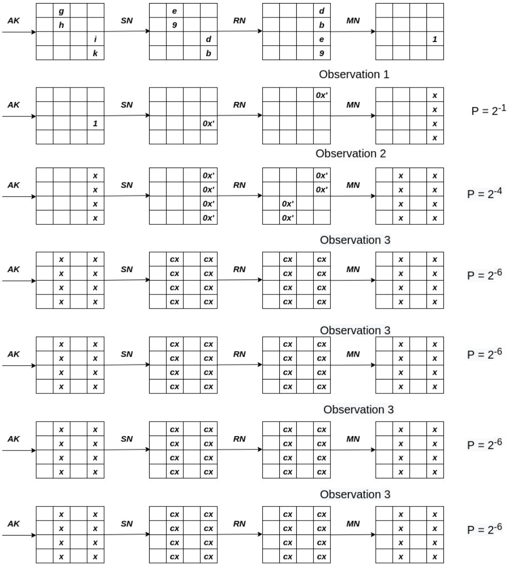
Observation 3 $\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \text{x} \begin{bmatrix} 0z_1 \\ 0z_2 \\ 0z_3 \\ 0z_4 \end{bmatrix} = \begin{bmatrix} 0z'_1 \\ 0z'_2 \\ 0z'_3 \\ 0z'_4 \end{bmatrix}$ if only if MSB's of $z_1, z_2, z_3\ and\ z_4$ are 0.

Truncated Six Round Differential Distinguisher

Based on the above observations, we show that if the input difference of 6-round KLEIN are all zero except the 13-th nibble, after encryption, the first and the third column of state matrix will stay 0 with the probability of 2^{-29}.
This happens because the difference in column will not transfer to other column due to the RotateBits algorithm and above observations.
This is the reason for high probability 6-round differential distinguisher.

Truncated Differential Analysis of 8-Round KLEIN-64

The 8 round distinguisher is constructed by adding an extra layer at the top and bottom of the 6 round distinguisher.
As this is CPA(chosen plain text) attack, we choose plain text pairs such a way that input difference to the second round is all zero except the 13-th nibble. We actually obtain the required input pairs of first round by reverse tracing the single nibble that should be active at the end of the round.This process is also diplayed in Figure 7



Observation 1

P = 2^{-1}

Observation 2

P = 2^{-4}

Observation 3

P = 2^{-6}

Observation 3

P = 2^{-6}

Observation 3

P = 2^{-6}

Observation 3

P = 2^{-6}

x is any difference in {0,....,15} and x' is any difference in {0,....,7}

c is either 1 or 0

Figure: 7 round characteristic of 8 Round Distinguisher

The Steps and Analysis Procedure

Step1. We will choose the input plaintexts in such a way that, all nibbles have some fixed values except four nibbles $X_{1,1}, X_{1,3}, X_{1,13}, X_{1,15}$. If we fix the values and change the values only in these 4 nibbles, that is called one structure. There are 2^{16} possible plain texts in one structure.

We can form nC2 i.e $(2^{16}\text{x}(2^{16}-1))/2 = 2^{31}$ plain text pairs from those 2^{16} plain texts.

If we took m structures then we will have $2^{16}m$ plain texts and $2^{31}m$ pairs.

Step 2. By Guessing the values(trying all possible values) of the subkey nibbles $sk_{1,1}, sk_{1,3}, sk_{1,13}, sk_{1,15}$ we should make sure that $\Delta SubNibbles(X_{1,1} \oplus sk_{1,1}) = 0e, \Delta SubNibbles(X_{1,3} \oplus sk_{1,3}) = 09, \Delta SubNibbles(X_{1,13} \oplus sk_{1,13}) = 0d\ and\ \Delta S(X_{1,15} \oplus sk_{1,15}) = 0b.$ The probability of this is 2^{-16} as we are fixing values of 4 particular nibbles, so the expected number of confirming pairs is 2^{31} x m x $2^{-16} = 2^{15}m$.

Step 3. Now take those remaining $2^{15}m$ pairs and encrypt them upto 8 rounds. Then verify the third and first columns output difference of MC^{-1} to zero. If not, discard the key guess. The probability of this event to happen is $(2^{-16})^2$ so the number of confirming pairs would be $2^{15} \ * \ m \ * \ (2^{-16})^2 = 2^{-17}m$. Now we should use meet in the middle technique.

Step 4. For the obtained confirming pairs from previous step, guess the value of the subkey $sk_{9,j}$, j = 0, 1, . . . , 7 to inverse the SubNibbles step and find input difference of 8-th round, i.e $\Delta X_{8,j}$, j = 0, 1, . . . , 7.

Step 5. Now, reverse the MixColumns step i.e MC^{-1} on ΔX_8 and verify whether the first column difference is zero and also the MSB of second column should be all 1 or all 0. If it failed to satisfy the above condition then discard the key guess.The expected confirming pairs is $2^{17} \ * \ m \ * \ 2^{-7} = 2^{-24}m$ as the probability of the above event is 2^{-7}.

Step 6. Guess the remaining 16 bits key in the similar way by verifying third and fourth columns.

Time and Space Complexity

Step 2 takes $2^{16} * 2^{16}/8$ one round 64 bit encryptions. In Step 3 it takes $2^{15} * 2^{16} * 2^{16} * 7/8$ encryptions whereas in step 4 it takes $2^{-17} * 2^{16} * 2^{16} * 2^{32}/8$ encryptions are required and step 6 takes about 2^{16} encryptions.So overall time complexity $2^{46.8}$ one round 64 bit encryptions.The data complexity is 2^{32} plaintexts. Memory complexity is 2^{32} 64bit states.

Integral Analysis

Observations

1. If we give 2^{32} different input values to the Rotate Nibble then after the Rotate Nibble and sub nibble operation and 3 rounds of KLIEN all the output nibbles are balanced
2. If in the input state i_{th} nibble is active where i=0,1,2,3,12,13,14,15 then after 1 round of KLIEN and one add key and one sub nibble all j_{th} nibbles are active where j=8,9,10,11,12,13,14,15

A	A	C	A	C	C	C	C
C	C	C	C	A	A	A	A

$\xrightarrow{1.5 round}$

X	X	X	X	X	X	X	X
A	A	A	A	A	A	A	A

Combining Obs. to Get 5 round Distinguisher

A	A	C	A	C	C	C	C
C	C	C	C	A	A	A	A

$\downarrow 5 round$

B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B

7 round integral attack

- We can take our 5 round distinguisher one step further as Round key addition does not change the Balance property. Let Y_j be the input to 6th round Sbox of the plaintext where j denotes the jth nibble. Then the xor sum of Y_j over all plaintext is 0
  Let y_j be the jth nibble obtained after Reversing the sub bytes. Clearly $y_j = S^{-1}$ $(X_j \oplus sk_{7,j})$
- After Mix columns of 6th round we get the following relation $Y_4 || Y_5 = S^{-1}(R^{-1}(e.(y_1||y_2) \oplus b.(y_2||y_3) \oplus d.(y_4||y_5) \oplus 9.(y_6||y_7)) \oplus sk_{6,4}||sk_{6,5})$

Analysis Procedure

- We first get **5** sets of 2^{32} plaintexts. For each of these sets we do the following-
- Guess $sk_{7,0}$,$sk_{7,1}$,$sk_{7,2}$,$sk_{7,3}$ for each plaintext and obtain $e.(y_1||y_2) \oplus b.(y_2||y_3)$
  Let $u1{=}e.(y_1||y_2) \oplus b.(y_2||y_3)$
  . Now we get 2^{24} different values of
  $(u1 \oplus d.(y_4||y_5) \oplus 9.(y_6||y_7))$ as each of the term is 8 bytes.
- Guess $sk_{7,4}$,$sk_{7,5}$ and obtain $d.(y_4||y_5)$
  Let $u2{=}u1 \oplus (y_4||y_5)$
  Now are left with 2^{16} values of $(u2 \oplus 9.(y_6||y_7))$

- Guess $sk_{7,6}$,$sk_{7,7}$ and obtain $d.(y_6||y_7)$
  Let $u3{=}u2 \oplus (y_6||y_7)$
  Now are left with 2^8 values of $(u3)$
- Now our equation becomes
  $Y_4||Y_5 = S^{-1}(R^{-1}(u3) \oplus sk_{6,4}||sk_{6,5})$
- We now guess $sk_{6,4}$ and $sk_{6,5}$ and then obtain the Sum of $Y_4||Y_5$. If its not 0 we discard our guess. Wrong key can also give 0 with 1/128 probability and therefore we used 5 sets

Practical Attack on 8 Rounds of KLEIN

Observations

- **Observation 1.** If the difference entering MixColumn is of the form 0000000X where X represents a non-zero difference in $\{1, \ldots, 7\}$ then the output difference is of the form 0Y0Y0Y0Y, where the wildcard Y represents a non-zero difference. That is, higher nibbles remain free of difference.
- **Observation 2.** If the difference entering MixColumn is of the form 0X0X0X0X where the wildcard X represents a difference in $\{0, \ldots, 7\}$, then the output difference is of the form 0Y0Y0Y0Y,where Y represents a possibly null difference. Furthermore, the average number of non-zero Y's is 3.75, as one can experimentally verify. For example, the input difference 04020405 leads to the output difference 0f090100.

Observations Cont'd

- **Observation 3.** If the difference entering MixColumn is of the form 0X0X0X0X where the wildcard X represents a difference in 8, . . . , f, then the output difference is of the form 0Y0Y0Y0Y, where Y represents a (possibly zero) difference. Furthermore, the average number of non-zero Y's is 3.75. Note that, unlike Observation 2, an X cannot be zero. For example, the input difference 0c0a080f leads to the output difference 010f0708.
- **Observation 4.** Given a random difference, KLEIN's Sbox returns a difference in 1, . . . , 7 with probability 7/15 approximates to $2^{1.1}$, for a random input. If the difference is b or e, the probability is 3/4 approximates to $2^{0.42}$.

Finding More Right Pairs with Neutral Bits

- A bit is said to be neutral with respect to a given differential (characteristic) when flipping this bit in an input conforming to the differential (characteristic) leads to a new input also conforming to that differential.
- In KLEIN, one can observe that the first two and last two input bytes in a plaintext block are neutral with respect to the first two rounds' collection of characteristics.
- Therefore, for example, after a 2^{28} effort to find a pair satisfying the 6-round differential, one can derive 2^{32} pairs for which the full differential is followed with probability $2^{23.26}$.

Key Recovery Of 8 Rounds

- The attack exploits the invertibility of the final MixNibbles and RotateNibbles to determine the output differences of each nibble after the last SubNibbles.
- With approximately 2^{34} encryptions, one can identify a conforming pair with high probability.
- Using neutral bits, one expects to produce approximately 8 other conforming pairs after $2^3$2 trials. This is more than enough to identify with certainty 32 bits of the last subkey.
- Overall, the 64 bits of the last subkey (and thus of the original key) can be found with complexity below 2^{35} encryptions.

Expanding to 7 and 8 rounds

- We observe that for a pair conforming to the 6-round differential, the SubNibbles of round 7 has all higher nibbles inactive. Therefore a 7-round distinguisher can be built with the same 2^{28} observations data complexity.
- In the eight-round attack one first collects approximately $2^{33.90}$ pairs, and records the ones that conform to the output difference as per our collection of characteristics.
- One expects to record approximately 4 pairs satisfying the difference by chance, and one conforming to the collection of characteristics. The conforming pair can be identified using the neutral bits.

Expanding to 7 and 8 rounds

- We observe that for a pair conforming to the 6-round differential, the SubNibbles of round 7 has all higher nibbles inactive. Therefore a 7-round distinguisher can be built with the same 2^{28} observations data complexity.
- In the eight-round attack one first collects approximately $2^{33.90}$ pairs, and records the ones that conform to the output difference as per our collection of characteristics.
- One expects to record approximately 4 pairs satisfying the difference by chance, and one conforming to the collection of characteristics. The conforming pair can be identified using the neutral bits.

Resistance against Side Channel Attacks

- KLEIN posses a highly balanced key schedule wrt its opposition against Key-Related attacks and the dexterity of the keys, secret sharing method is used for resistance to side-channel attacks In which a CRT Algorithm is implemented.
- The masking based on secret sharing increases the hardware overhead,but still promising because it was theoretically proven to be secure against DPA attacks. Differential power analysis is a side-channel attack involving mathematically analyzing power consumption measurements from a crypto-system. It makes the use of varying power consumption of microprocessorsor other hardwares while performing operations using secret keys.

Performance Evaluation On AVR microcontroller

- **With Respect To Energy Consumption**
- **With Respect To Memory Efficiency**
- **With Respect To Security**

With Respect To Energy Consumption

- To measure energy consumption, it is assumed that the energy per CPU cycle is fixed.This energy consumption includes the key scheduling and encryption.KLEIN is the best algorithm in comparison to other ciphers in aspect of energy consumption.
- We learnt many important factors that effect the energy consumption such as the performance of code depends on several specifications of a cipher such as the Type of instructions, Mode of operation, Structure, Number of loops,Number of Rounds etc.
- Another important factor is which kind of instruction.it is important factor to use correct instruction to write a code.

With Respect To Energy Consumption Cont'd

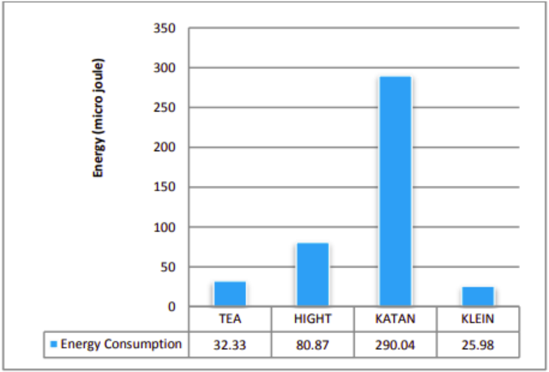
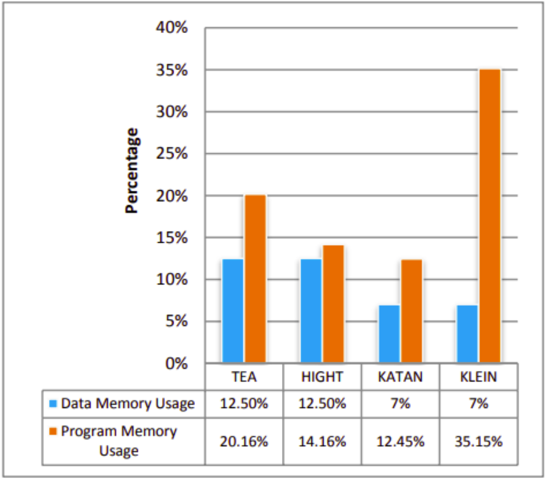


Figure: Energy consumption comparison of focused ciphers

With Respect To Memory Efficiency

- The memory usage of various lightweight algorithms is com-pared in Figure below which shows the percentage of memory used for each cipher. Analyzed results clearly state that KLEIN uses longer Flash memory space than the rest because the assembly code size of this algorithm is more than others Where as the percentage of SRAM usage for KLEIN cipher is not high as other ciphers. The Data Memory Usage for KATAN and KLEIN algorithm is equivalent.

With Respect To Memory Efficiency Cont’d



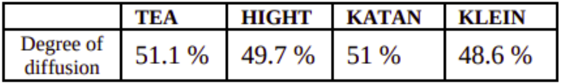
	TEA	HIGHT	KATAN	KLEIN
■ Data Memory Usage	12.50%	12.50%	7%	7%
■ Program Memory Usage	20.16%	14.16%	12.45%	35.15%

Figure: Memory Efficiency comparison of focused ciphers

With Respect To Security.

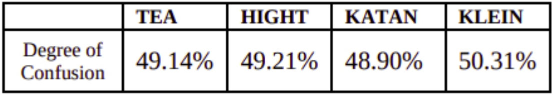
- Taking degree of confusion and diffusion as the security criteria. KLEIN has least de-gree of diffusion highest degree of confusion.This is related to the fact that of differentstructure.KLEIN is SPN and others Feistel.

With Respect To Security Cont'd



	TEA	HIGHT	KATAN	KLEIN
Degree of diffusion	51.1 %	49.7 %	51 %	48.6 %

Figure: THE ANALYSIS OF DIFFUSION



	TEA	HIGHT	KATAN	KLEIN
Degree of Confusion	49.14%	49.21%	48.90%	50.31%

Figure: THE ANALYSIS OF CONFUSION

Outline

1. Introduction
2. Cipher Specifications
3. Observations
4. **Brownie Point Nominations**
5. Conclusion

Brownie Point

1. All diagrams(except one).
2. Sbox analysis.
3. Detailed explanation of cipher and it's attacks.

Outline

1. Introduction
2. Cipher Specifications
3. Observations
4. Brownie Point Nominations
5. Conclusion

Conclusion

In this Presentation ,We have described the Design Specifications of KLEIN Cipher, Different attacks on KLEIN Cipher,Performance of KLEIN is Compared with other lightweight Ciphers.The goal is to explain how KLEIN is the practical and secure cipher for low-resource applications.Hardware efficiency of KLEIN from its simple structure with an involutive S-box. If we further modify rotate nibbles algorithm and Increase diffusion we can prevent the attacks.Further we can also compare DATA, TIME AND MEMORY complexities if skipjack, present, and klein as future work.

Thanks

Team Members

- Siram Nikhil - 11841090
- Kolli Venkata Madhukar - 11840670
- Gagandeep Singh - 11840480

Implementation Info

- Github Link: https://github.com/gagan2005/termpaper