PRESENT Cipher

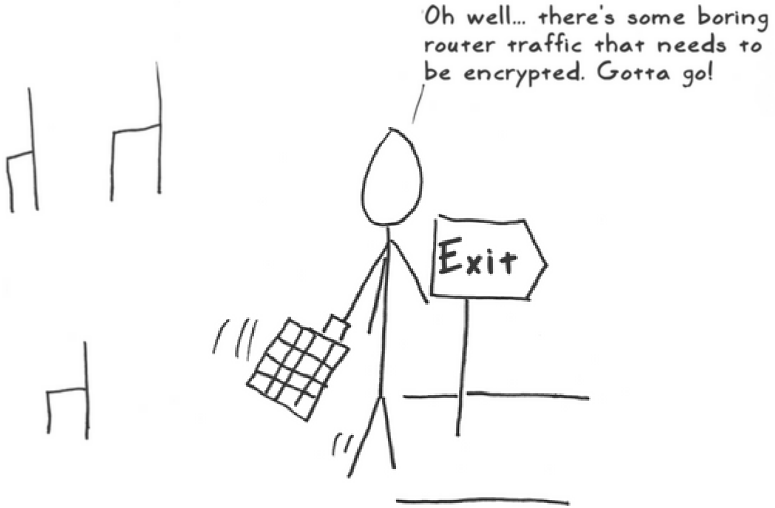
Random_LavaRands

Department of Computer Science
Indian Institute of Technology Bhilai

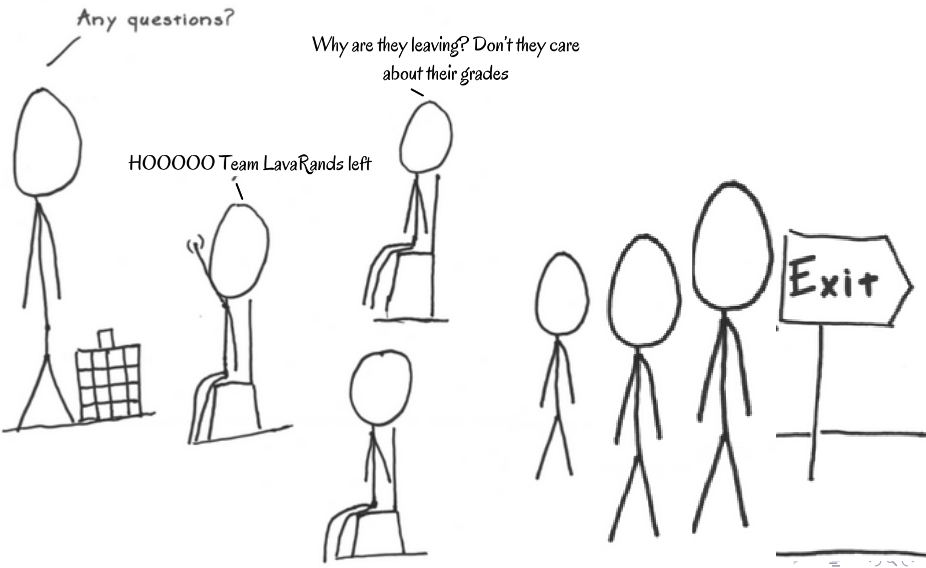
December 7,2024

Outline

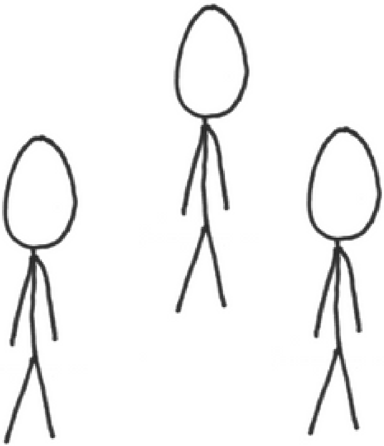
Problem

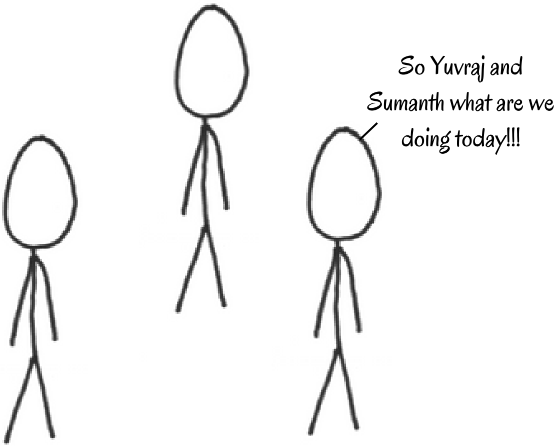

Oh well... there's some boring router traffic that needs to be encrypted. Gotta go!
Exit

Once upon a time clearing in a magical forest

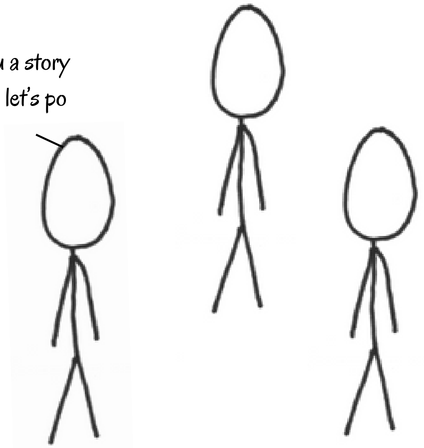

Any questions?
Why are they leaving? Don't they care about their grades
HOOOOO Team LavaRands left
Exit

A lot of boring AES now you studied so let's begin and let it encrypt boring network traffic only




So Yuvraj and
Sumanth what are we
doing today!!!

So let me tell u a story
so hop on and let's po




Dil chahta hai
Team Lavarands fatfatiya

Hola! I am PRESENT, an ultra lightweight cipher.
Let me tell you about me.

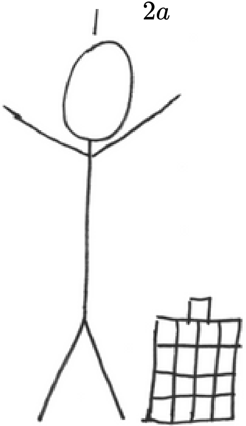
WHY ME ??

$$x^2 - 1 = 0$$

$$x = \frac{[-b \pm \sqrt{(b2 - 4ac)}]}{2a}$$

come on bro, it's just

$$x = \pm 1$$



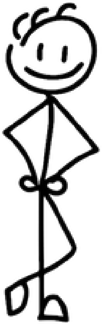
AES

Simply why using high computation power, memory for a short task

PRESENT

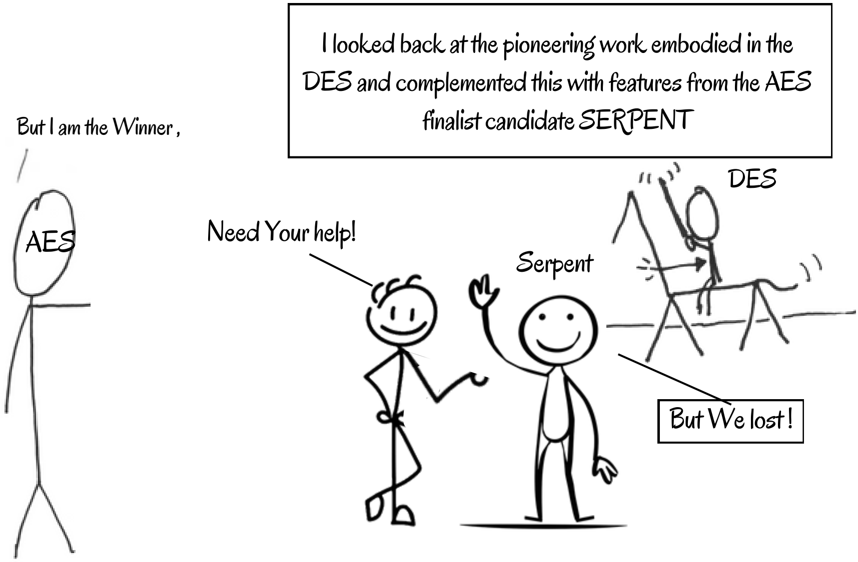
PRESENT: Secure solutions for small devices, where AES falls short!

- .PRESENT is designed for minimal power consumption
- are unlikely to require the encryption of large amounts of data.
- that demand the most efficient use of space

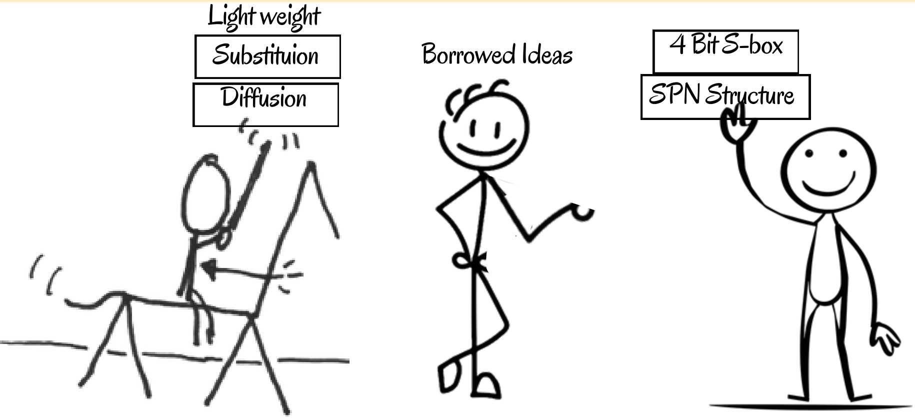


- Applications will only require moderate security levels
- crucial for battery-powered or energy-harvesting devices.

History


I looked back at the pioneering work embodied in the DES and complemented this with features from the AES finalist candidate SERPENT
But I am the Winner,
AES
Need Your help!
Serpent
DES
But We lost!

History



war is over, you can demount from horse

demonstrated excellent performance in hardware.

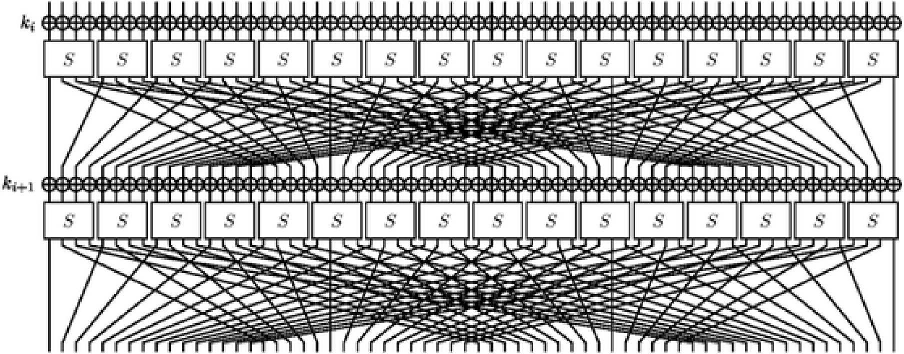
You made me cry

Outline

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

Substitution/permutation

This is how I look the old conufsion and diffusion story



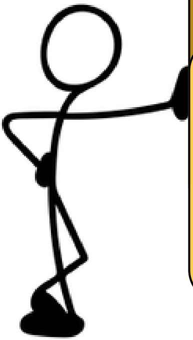
And this is my SBox.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S[x]$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Cipher Design

- PRESENT-80 is an example of SP-network.
- 4-bit S-Box is applied 16 times in parallel for the 64-bit input during each round.

Ya so this is how I work in short



High level psuedo-code of PRESENT algorithm

```
generateRoundKeys() ?
for i = 1 to 31 do
    addRoundKey(State,Ki )
    sBoxLayer(State)
    pLayer(State)
addRoundKey(State,K32)
```

Cipher Design

Ah I see so you want to know about my generateRound key function. I got you bro.

High level psuedo-code of PRESENT algorithm

1: generateRoundKeys()
2: for i = 1 to 31 do
3: addRoundKey(State,Ki)
4: sBoxLayer(State)
5: pLayer(State)
6: addRoundKey(State,K32)

Cipher Design

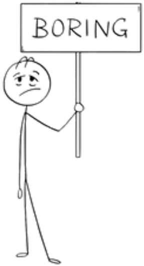
1. the key register is rotated by 61 bit positions to the left

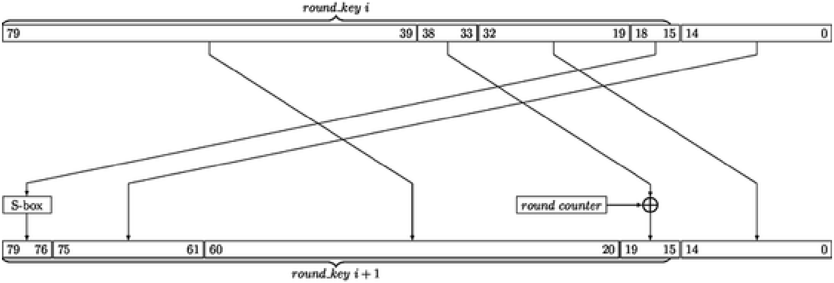
1. $[k_{79}k_{78}\dots k_1k_0] = [k_{18}k_{17}\dots k_{20}k_{19}]$
2. $[k_{79}k_{78}k_{77}k_{76}] = S[k_{79}k_{78}k_{77}k_{76}]$
3. $[k_{19}k_{18}k_{17}k_{16}k_{15}] = [k_{19}k_{18}k_{17}k_{16}k_{15}]$

Key Schedule

Pseudo Code

```
keyRegister = K
For round = 1 to R do:
   RoundKeys.append(keyRegister[0:64])
   keyRegister = keyRegister << 61 | keyRegister >> (keyLength - 61)
   keyRegister[0:4] = Sbox[keyRegister[0:4]]
   if keyLength == 80:
       keyRegister[15:20] ^= round
   elif keyLength == 128:
       keyRegister[62:67] ^= round
```





Outline

Observations

Key Schedule Observations

- Non-Linearity and Diffusion: Utilizes an S-box and bitwise rotation to introduce non-linearity and diffusion, disrupting statistical relationships between master and round keys.
- Resistance to Related-Key Attacks: Small changes in the master key produce significantly different round keys, though related-key vulnerabilities exist in reduced-round versions.
- Limited Key Size Security: Supports 80-bit and 128-bit keys; 80-bit keys are less secure in modern contexts, while 128-bit keys improve resistance without altering fundamental vulnerabilities.

The nerdy stuff again

S Box and Diffusion Observations

Any five-round differential characteristic of present has a minimum of 10 active S-boxes.

Let ϵ4R be the maximal bias of a linear approximation of four rounds of PRESENT.

$$Then \epsilon_{4R} \leq \frac{1}{2^7}$$

Differential CryptAnlysis

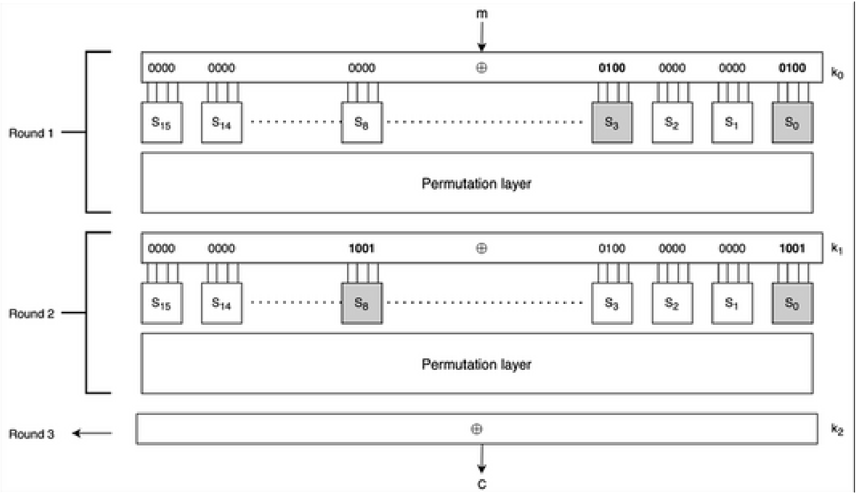
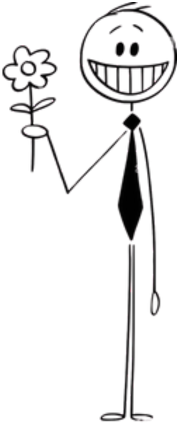


Figure: Attack Model

DDT

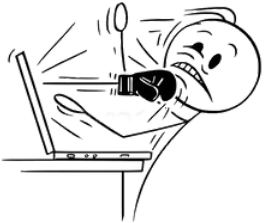


	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	4	0	0	0	4	0	4	0	0	0	4	0	0
2	0	0	0	2	0	4	2	0	0	0	2	0	2	2	2	0
3	0	2	0	2	2	0	4	2	0	0	2	2	0	0	0	0
4	0	0	0	0	0	4	2	2	0	2	2	0	2	0	2	0
5	0	2	0	0	2	0	0	0	0	2	2	2	4	2	0	0
6	0	0	2	0	0	0	2	0	2	0	0	4	2	0	0	4
7	0	4	2	0	0	0	2	0	2	0	0	0	2	0	0	4
8	0	0	0	2	0	0	0	2	0	2	0	4	0	2	0	4
9	0	0	2	0	4	0	2	0	2	0	0	0	2	0	4	0
A	0	0	2	2	0	4	0	0	2	0	2	0	0	2	2	0
B	0	2	0	0	2	0	0	0	4	2	2	2	0	2	0	0
C	0	0	2	0	0	4	0	2	2	2	2	0	0	0	2	0
D	0	2	4	2	2	0	0	2	0	0	2	2	0	0	0	0
E	0	0	2	2	0	0	2	2	2	2	0	0	2	2	0	0
F	0	4	0	0	4	0	0	0	0	0	0	0	0	0	4	4

Table: DDT of the S-box

Rounds		Diff.	Prob.
I		$x_0 = 4,\ x_4 = 4$	
R_1	k_0	$x_0 = 4,\ x_4 = 4$	1
R_1	S	$x_0 = 5,\ x_3 = 5$	2^{-4}
R_1	P	$x_0 = 9,\ x_8 = 9$	1
R_2	k_1	$x_0 = 9,\ x_8 = 9$	1

Table: Characteristics



Characteristic

$$(x_0 = 4, x_3 = 4) \xrightarrow{\mathsf{R}} (x_0 = 9, x_8 = 9)$$

Decrease Wrong pair $\dashrightarrow$ Idea of filtering

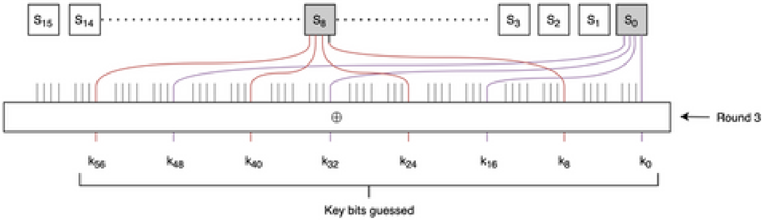
Observe from the DDT that transitions from 9 $\rightarrow$ {2,4,6,8,c,e}

Thus, after the effect of permutation layer of the second round, c1 $\oplus$ c2 must belong to the set given below :

$$\{x_4 = 1, x_6 = 1\}, \{x_6 = 1, x_8 = 1\}, \{x_4 = 1, x_6 = 1, x_8 = 1\}, \{x_6 = 1, x_{12} = 1\}, \{x_6 = 1, x_8 = 1, x_{12} = 1\}, \ldots$$

Characteristic

$$(x_0 = 4, x_3 = 4) \xrightarrow{\mathsf{R}} (x_0 = 9, x_8 = 9)$$



Guess 8 bits of the key k2 as shown in the figure.

The probability that the result of partial decryption probabilistically matches Δ out is <<1.

Thus, the right guess reaches Δ out more than any other wrong guess

Integral Attack

The propagation Of Most Important Bits

Choose a set of 2^n $(n = 4)$

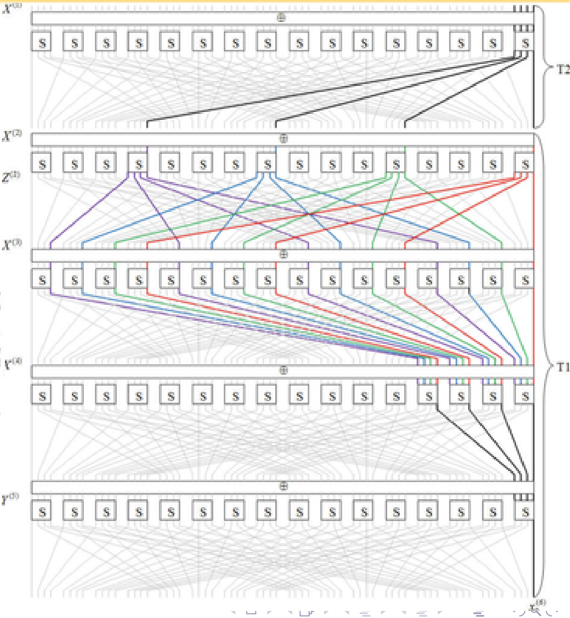
where the rightmost n bits take all possible valuesother bits are chosen to be arbitrary constants

For every guessing of the corresponding subkeys in the last $(r-m)$ rounds, decrypt the ciphertexts to obtain the one bit state $y_0^{(m+1)}$ after the m-th round, where m is the length of integral distinguishers.

Check whether $\bigoplus_\Lambda y_0^{(m+1)} (= \bigoplus_\Lambda x_0^{(m+1)})$ is zero, where Λ with $|\Lambda| = 2^n$ is the set of chosen plaintexts. If the equation is not satisfied, we know the guessed subkey is wrong. Then, we guess another subkey and repeat until the correct subkey is found.

Recover the remaining key bits in the master key by exhausting method.

uppose we need to guess k bit subkey in the last $(r-m)$ rounds, the com-



Summary

Rounds	Key Size	Data	Time
5	all	$N \cdot 2^{32}$	-
5	80	$2^{6.4}$	$2^{25.7}$
6	80	$2^{22.4}$	$2^{41.7}$
7	128	$2^{24.3}$	$2^{100.1}$
7	80	$2^{8.3}$	2^{60}
8	80	$2^{10.1}$	$2^{72.6}$
9	80	$2^{20.3}$	2^{60}
10	128	$2^{22.4}$	$2^{99.3}$

Attack Complexity Summary

Outline

Brownie Points

Implementation of DC of Reduced Round PRESENT Cipher

We could not find any implementation of Differential analysis on the round reduced version of PRESENT. So, using the idea of differential and filtering taught in the course, we have implemented a differential attack on 3 Rounds of PRESENT.

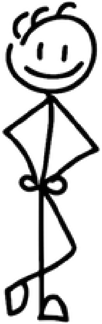
Brownie Points

Theoretical Linear CryptAnalysis

Theortical analysis of the linear crypt analysis based on the Linear Approximation Table. caluculated a bound inear approximation bias for 28 rounds of PRESENT

Outline

Slide One



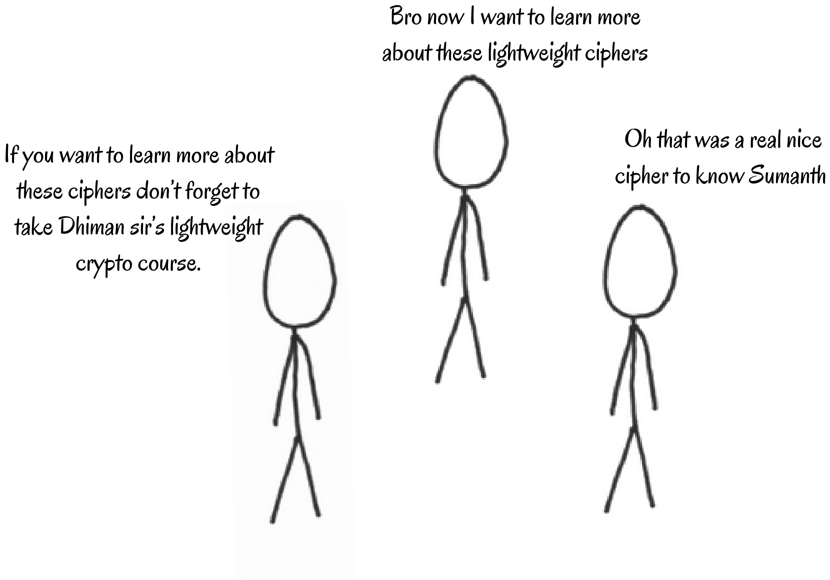
So you see how I work and you use me at many places unknowingly like:

- Internet of Things (IoT)
- Smart Cards
- Mobile and Wearable Devices

Slide Three

Gotta go and save some IoT devices. See ya

Slide Three


Bro now I want to learn more about these lightweight ciphers
If you want to learn more about these ciphers don't forget to take Dhiman sir's lightweight crypto course.
Oh that was a real nice cipher to know Sumanth

Thanks

Team Members

- Chetan
- Yuvraj
- Sumanth

Implementation Info

- Github Link: Link