Introduction
ooo

Cipher Specifications
oooooooooooooo

Brownie Point Nominations
oo

Observations
oooooooo

Conclusion
ooo

Rectangle Cipher

<Ciphermakers>

Department of <CS553-Cryptography>
Indian Institute of Technology Bhilai

December 6, 2021

Outline

1. **Introduction**

2. Cipher Specifications

3. Brownie Point Nominations

4. Observations

5. Conclusion

Introduction

History and Evolution

- PRESENT was introducd at CHES'2007 with a low-cost hardware performance.
- Further, LED was introduced at CHES'2011 adding a reasonable software performance.
- Then the Serpent Block Cipher was introduced with the bit-slice technique.
- Finally, Rectangle Cipher was introduced with all better improvements from the previous ciphers.

Introduction

Rectangle Cipher is

- Lightweight Block Cipher
- Based on SP-Network
- 16 4x4 S-boxes in parallel in S-Layer
- 3 rotations composed in the P-layer
- Both Hardware and Software Friendly

Outline

1. Introduction

2. Cipher Specifications

3. Brownie Point Nominations

4. Observations

5. Conclusion

Cipher Specifications

- Lightweight Block Cipher
- Bit-Slice Style
- Competetive Software Performance
- Hardware Friendly
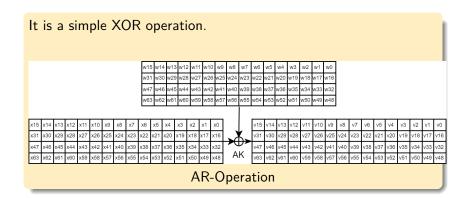- Very Strong Security

The Round Transformation

- AddRoundkey (ARK)
- SubColumn (SC)
- ShiftRow (SR)
- Key Schedule (KS)

Pseudo-Code-

GenerateRoundKeys(state):
.   **for i = 0 to 24 do:**
.   **ARK(state, Ki)**
.   **SC(state)**
.   **SR(state)**
.   **ARK(state, K25)**

AddRoundkey (AR)

It is a simple XOR operation.

AR-Operation

SubColumn (SC)

S-Box

The S-Box $S : F_2^4 \to F_2^4$.

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

Rectangle S-Box

SubColumn

It is the parallel application of S-boxes oo the 4 bits in the same column. Input- Col(j) = v3,j || v2,j || v1,j || v0,j for $0 \leq j \leq 15$
Output- S(Col(j)) = x3,j || x2,j || x1,j || x0,j.

SC-Operation

Introduction
ooo

Cipher Specifications
ooooo●oooooo

Brownie Point Nominations
oo

Observations
ooooooooo

Conclusion
ooo

Differential Distribution Table (DDT)

```
[16   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
[ 0   0   0   2   0   0   4   2   0   0   0   2   0   0   4   2]
[ 0   0   0   0   0   0   2   2   2   0   2   0   2   4   0   2]
[ 0   0   0   2   0   0   2   0   2   4   2   2   2   0   0   0]
[ 0   0   0   4   0   0   0   4   0   0   0   4   0   0   0   4]
[ 0   2   0   0   4   2   0   0   4   2   0   0   0   2   0   0]
[ 0   2   4   0   2   0   0   0   0   0   0   2   2   2   0   2]
[ 0   0   4   0   2   2   0   0   0   2   0   2   2   0   0   2]
[ 0   2   0   2   0   2   0   2   0   2   0   2   0   2   0   2]
[ 0   2   0   0   0   2   4   0   0   2   0   0   0   2   4   0]
[ 0   0   0   0   0   4   2   2   2   0   2   0   2   0   0   2]
[ 0   4   0   2   0   0   2   0   2   0   2   2   2   0   0   0]
[ 0   0   0   0   4   0   0   0   4   0   4   0   0   0   4   0]
[ 0   2   0   0   0   2   0   0   0   2   4   0   0   2   4   0]
[ 0   0   4   2   2   2   0   2   0   2   0   0   2   0   0   0]
[ 0   2   4   2   2   0   0   2   0   0   0   0   2   2   0   0]
```

DDT for Rectangle cipher S-Box

Linear Approximation Table (LAT)

```
[ 8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  4  0 -4  0  0  2 -2 -2 -2 -2 -2  2 -2]
[ 0  0  0  0  0  0  4  4  0  0  4 -4  0  0  0  0]
[ 0  0  0 -4  4  0  0  0 -2  2 -2 -2 -2 -2  2 -2]
[ 0  0  0  0  0  0 -4  4  0  0  0  0  0  0  4  4]
[ 0  0 -4  0  0 -4  0  0 -2  2 -2 -2  2  2 -2  2]
[ 0  0  0  0  0  0  0  0  4  4  0  0 -4  4  0  0]
[ 0  0 -4  0 -4  0  0  0 -2  2  2  2 -2 -2  2 -2]
[ 0  0  0 -4 -2 -2  2 -2  0 -4  0  0 -2  2  2  2]
[ 0  0  0  0 -2  2  2 -2  2  2 -2 -2  4  0  4  0]
[ 0  0  0 -4 -2 -2 -2  2  4  0  0  0  2 -2 -2 -2]
[ 0  0  0  0  2 -2  2 -2  2  2  2  2  0 -4  0  4]
[ 0  4  0  0 -2  2 -2 -2  0  0  0 -4 -2 -2 -2  2]
[ 0  4  4  0 -2 -2  2  2 -2  2 -2  2  0  0  0  0]
[ 0 -4  0  0 -2  2  2  2  0  0 -4  0 -2 -2 -2  2]
[ 0  4 -4  0  2  2  2  2  2 -2 -2  2  0  0  0  0]
```

LAT for Rectangle cipher S-Box

ShiftRow (SR)

It is just a left rotation to each row.

SR-Operation

Key Schedule (KS)

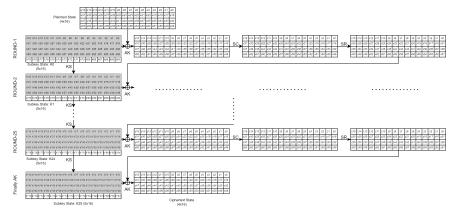### 80-bit key

- SC to the bits at the 4 uppermost rows and the 4 rightmost columns.
- Using a 1-round generalized Feistel transformation-
  Row'0 := (Row0 $<<$ 8) $\oplus$ Row1, Row'1 := Row2, Row'2 := Row3, Row'3 := (Row3 $<<$ 12) $\oplus$ Row4, Row'4 := Row0
- A 5-bit round constant RC[i] is XORed with the 5-bit key state for i $\in$ (1,2,..,24).

Block Diagram

80-bit key block diagram (80.png)

Security Analysis

Integral Cryptanalysis

- We implemented the Square attack which used a 4-round integral distinguisher
- Encryption : After 4-rounds, the XOR sum in any 4 bit positions equals to 0, i.e. (Balanced property)
  $\oplus S_4[0] = \oplus S_4[17] = \oplus S_4[43] = \oplus S_4[60] = 0$

P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15
P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	P31
P32	P33	P34	P35	P36	P37	P38	P39	P40	P41	P42	P43	P44	P45	P46	P47
P48	P49	P50	P51	P52	P53	P54	P55	P56	P57	P58	P59	P60	P61	P62	P63

- Decryption : We choose 2^{48} plaintexts s.t. cols - 0, 13, 14, 15 maintain CONSTANT property and other 12 cols maintain the ALL property
- 2^{48} Intermediate values $\Rightarrow 2^{47}$ subsets $\Rightarrow 2$ values.
- $4 \rightarrow 7 \rightarrow 25$ rounds with same integral distinguisher.

Security Analysis

Differential Cryptanalysis

- Differential Cryptanalys is strongest techniques for the cryptanalysis of block ciphers.
- Using the algorithm based on the branch and bound method, the best differential trails from round-1 to round-15 were found-

♯ R	Cor. Pot.	♯ R	Cor. Pot.	♯ R	Cor. Pot.
1	2^{-2}	6	2^{-20}	11	2^{-50}
2	2^{-4}	7	2^{-26}	12	2^{-56}
3	2^{-8}	8	2^{-32}	13	2^{-62}
4	2^{-12}	9	2^{-38}	14	2^{-68}
5	2^{-16}	10	2^{-44}	15	2^{-74}

- Using the 14-round differential propagation, we can mount an attack on 18-round Rectangle cipher
- 25-round Rectangle is enough to behold out against this differential cryptanalysis attack.

Outline

Brownie Point Nominations

- Rectangle is based on SP-Network.
- It is slightly similar to AES.
- Out of 25, the maximum of 18-rounds can be attacked.
- The remaining 7-rounds are for security purposes.
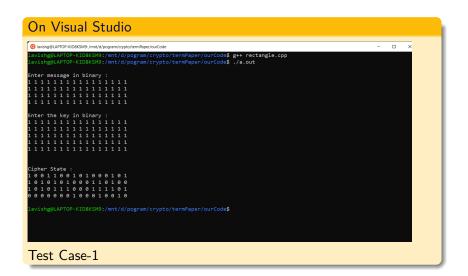- It attains a very fast software as well as hardware performance.

Outline

1. Introduction

2. Cipher Specifications

3. Brownie Point Nominations

4. Observations

5. Conclusion

Observations

Rectangle Cipher Implementation

- Rectangle is an iterated block cipher with block length 64-bits and key length as 80 or 128bits.

- The S-box of rectangle can be implemented using a sequence of 12 basic logical instructions.

- The P-Layer of rectangle is composed of 3 rotations.

Introduction
○○○

Cipher Specifications
○○○○○○○○○○○○○

Brownie Point Nominations
○○

Observations
○○●○○○○○○

Conclusion
○○○

Code Snippet

On Visual Studio

Test Case-1

Code Snippet

On Visual Studio

Test Case-2
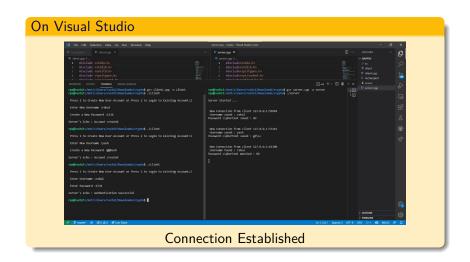
Software Implementation

User-Login Application

- User i.e client can choose between Sign-up or Sign-in option to the server.
- For Sign -up, user gives the username and password.
- For Sign-in, user have to give the same username and password created during sign-up.
- If the encrypted password matches the password stored in the server, then authentication passes.
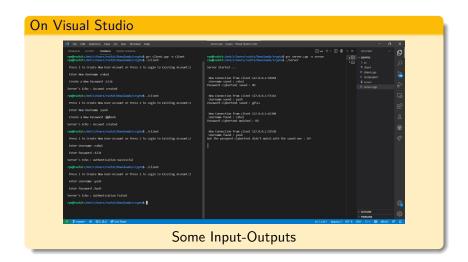- Otherwise, authentication fails.

Demo

On Visual Studio

- Client.cpp
- Server.cpp
- Rectangle Cipher used

Code Snippet

On Visual Studio

Connection Established

Code Snippet

On Visual Studio

Some Input-Outputs

Outline

Introduction
ooo

Cipher Specifications
ooooooooooooo

Brownie Point Nominations
oo

Observations
oooooooo

Conclusion
oooo

Conclusion

Rectangle Cipher

- It is a bit-slice styled lightweight block cipher.
- It provides applications enough flexibility.
- It has the ability to trigger various new cryptographic problems.
- Its security is encouraged.

Thanks

Team Members

- Ruchit Prakash Saxena (11941040)
- Anubh Sanoj Gupta (11940150)
- Lavish (11940640)

Implementation Info

- Github Repository:
  https://github.com/ruchitsaxena/Rectangle-cipher