Cryptanalysis of TWIS Block Cipher

Onur KOÇAK, Neşe ÖZTOP

Institute of Applied Mathematics
Middle East Technical University, Turkey

SKEW 2011
February 17, 2011

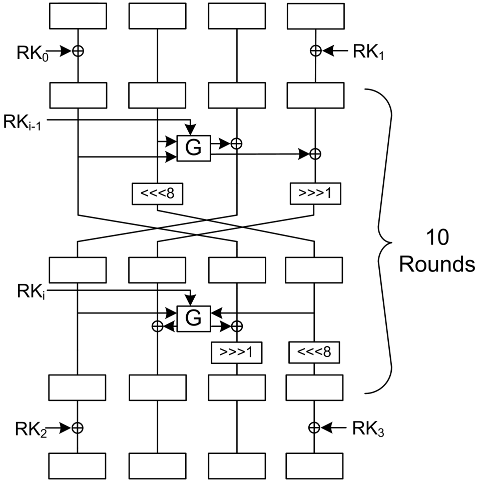
Outline

Outline

TWIS Block Cipher

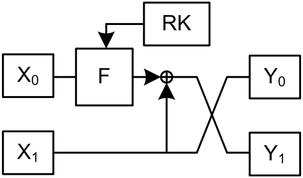
- A lightweight block cipher
- Key Size/Block Size: 128 bits
- 2-Branch Generalized Feistel Network
- 10 Rounds

TWIS Algorithm



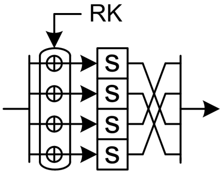
G-Function

- *G*-Function is the round function of the algorithm



F-Function

- *F*-Function is the core of the *G*-function
- Consists of S-Box and a permutation



S-Box

- 6x8 S-Box
- 8-bit input $I \rightarrow I \wedge 0x3f \rightarrow$ 6-bit

Table: S-Box

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	90	49	d1	c6	2f	33	74	fb	95	6d	82	ea	0e	b0	a8	1c
1	28	d0	4b	92	5c	ee	85	b1	c4	0a	76	3d	63	f9	17	af
2	bf	bf	19	65	f7	7a	32	20	16	ce	e4	83	9d	5b	4c	d8
3	ee	99	2e	f8	d4	9b	0f	13	29	89	67	cd	71	dd	b6	f4

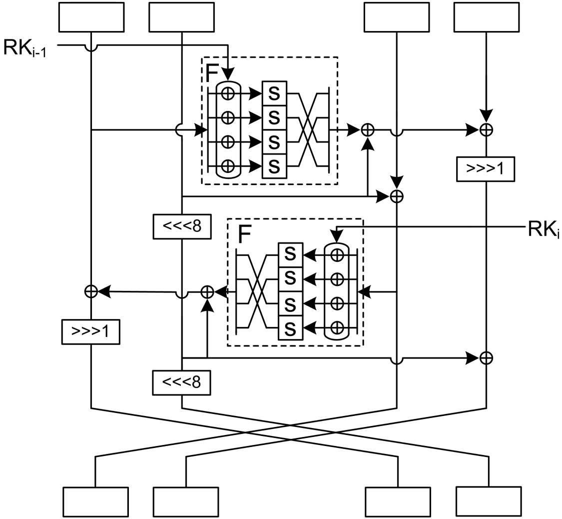
S-Box

- 6x8 S-Box
- 8-bit input $I \rightarrow I \wedge 0x3f \rightarrow$ 6-bit

Table: S-Box

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	90	49	d1	c6	2f	33	74	fb	95	6d	82	ea	0e	b0	a8	1c
1	28	d0	4b	92	5c	ee	85	b1	c4	0a	76	3d	63	f9	17	af
2	bf	bf	19	65	f7	7a	32	20	16	ce	e4	83	9d	5b	4c	d8
3	ee	99	2e	f8	d4	9b	0f	13	29	89	67	cd	71	dd	b6	f4

Alternative Round Function

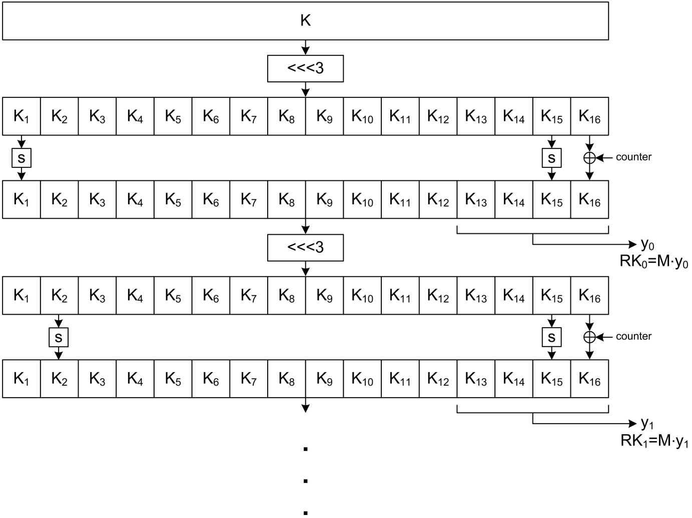


Key Schedule

- Key schedule generates 11 subkeys
- NFSR which uses an S-Box and a diffusion matrix

$$M = \begin{pmatrix} 0x01 & 0x02 & 0x04 & 0x06 \\ 0x02 & 0x01 & 0x06 & 0x04 \\ 0x04 & 0x06 & 0x01 & 0x02 \\ 0x06 & 0x04 & 0x02 & 0x01 \end{pmatrix}$$

Key Schedule



Outline

Overview of the Differential Attack

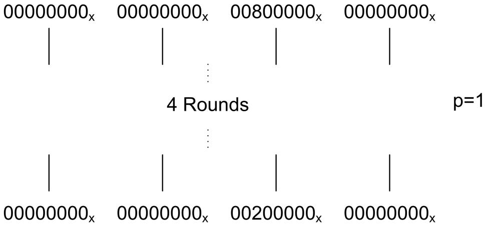
- Attack on 10-Round TWIS
- Exclude final key whitening
- 9.5-Round Characteristic
- Recover 12 bits of 32-bit round subkey

Properties

Property 1:
The first two bits of the S-Box input is ignored: $O = S(I \wedge 0x3f)$.
Property 2:
Input differences $0x01$ and $0x25$ produce zero output differences with probability 2^{-5}.

9.5-round Differential Characteristic

- First find a 4-round characteristic of probability 1 using *Property 1*.



9.5-round Differential Characteristic

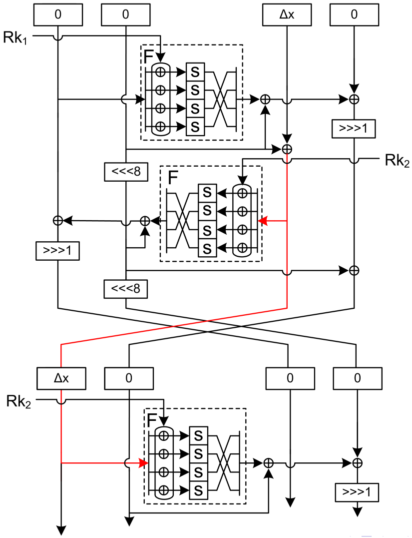
- Then, extend the characteristic by appending rounds to the beginning and the end
- Use *Property 2* in order to decrease the number of active S-Boxes

9.5-round Differential Characteristic

Rounds	ΔI_0	ΔI_1	ΔI_2	ΔI_3	# Active S-boxes	I/O Diff. for S-box	Probability
1	02000000_x	00000000_x	00000000_x	$0000A600_x$	1	0x02 → 0xA6	2^{-4}
2	00000000_x	00000000_x	01000000_x	00000000_x	1	0x01 → 0x00	2^{-5}
3	01000000_x	00000000_x	00000000_x	00000000_x	1	0x01 → 0x00	2^{-5}
4	00000000_x	00000000_x	00800000_x	00000000_x	0	-	1
5	00800000_x	00000000_x	00000000_x	00000000_x	0	-	1
6	00000000_x	00000000_x	00400000_x	00000000_x	0	-	1
7	00400000_x	00000000_x	00000000_x	00000000_x	0	-	1
8	00000000_x	00000000_x	00200000_x	00000000_x	1	0x20 → 0x83	2^{-4}
9	00200000_x	00000000_x	80000041_x	00000000_x	2	0x20 → 0x83 0x01 → 0x00	$2^{-5} \cdot 2^{-4}$
9.5	80000041_x	80000041_x	00100000_x	00000000_x	1	0x01 → 0x00	2^{-5}
	80000041_x	00004180_x	80100041_x	$C0000020_x$	-	-	-

The total probability is 2^{-32}.

9.5-round Differential Characteristic

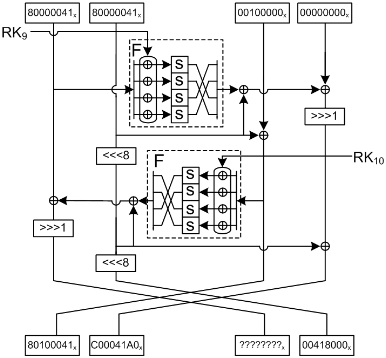


9.5-round Differential Characteristic

Rounds	ΔI_0	ΔI_1	ΔI_2	ΔI_3	# Active S-boxes	I/O Diff. for S-box	Probability
1	02000000_x	00000000_x	00000000_x	$0000A600_x$	1	0x02 → 0xA6	2^{-4}
2	00000000_x	00000000_x	01000000_x	00000000_x	1	0x01 → 0x00	2^{-5}
3	01000000_x	00000000_x	00000000_x	00000000_x	1	0x01 → 0x00	1*
4	00000000_x	00000000_x	00800000_x	00000000_x	0	-	1
5	00800000_x	00000000_x	00000000_x	00000000_x	0	-	1
6	00000000_x	00000000_x	00400000_x	00000000_x	0	-	1
7	00400000_x	00000000_x	00000000_x	00000000_x	0	-	1
8	00000000_x	00000000_x	00200000_x	00000000_x	1	0x20 → 0x83	2^{-4}
9	00200000_x	00000000_x	80000041_x	00000000_x	2	0x20 → 0x83 0x01 → 0x00	2^{-5}*
9.5	80000041_x	80000041_x	00100000_x	00000000_x	1	0x01 → 0x00	1*
	80000041_x	00004180_x	80100041_x	$C0000020_x$	-	-	-

The total probability is reduced to 2^{-18}.

Attack Procedure

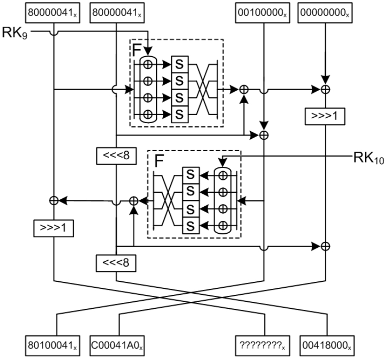


- Take $N = c.2^{18}$ plaintext pairs P^i, P^{i*} s.t.

  $$P^i \oplus P^{i*} = (02000000_x, 00000000_x, 00000000_x, 0000A600_x)$$

  and obtain their corresponding ciphertexts C^i, C^{i*}.
- Check the first 64-bit and the last 32-bit ciphertext difference and keep the text pairs satisfying correct differences.
- Keep a counter for each possible value of the 12 bits of the subkey RK_{10} corresponding to the second and the fourth bytes.

Attack Procedure

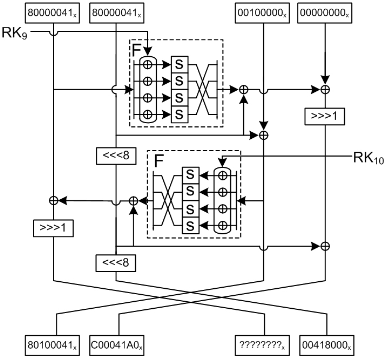


- Take $N = c.2^{18}$ plaintext pairs P^i, P^{i*} s.t.

  $$P^i \oplus P^{i*} = (02000000_x, 00000000_x, 00000000_x, 0000A600_x)$$

  and obtain their corresponding ciphertexts C^i, C^{i*}.
- Check the first 64-bit and the last 32-bit ciphertext difference and keep the text pairs satisfying correct differences.
- Keep a counter for each possible value of the 12 bits of the subkey RK_{10} corresponding to the second and the fourth bytes.

Attack Procedure

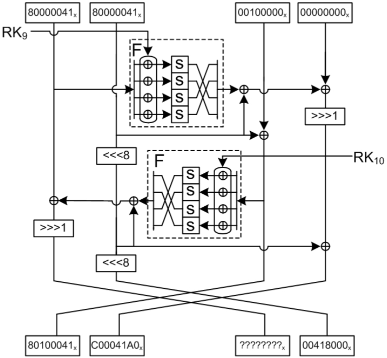


- Take $N = c.2^{18}$ plaintext pairs P^i, P^{i*} s.t.

$$P^i \oplus P^{i*} = (02000000_x, 00000000_x, 00000000_x, 0000A600_x)$$

  and obtain their corresponding ciphertexts C^i, C^{i*}.
- Check the first 64-bit and the last 32-bit ciphertext difference and keep the text pairs satisfying correct differences.
- Keep a counter for each possible value of the 12 bits of the subkey RK_{10} corresponding to the second and the fourth bytes.

Attack Procedure

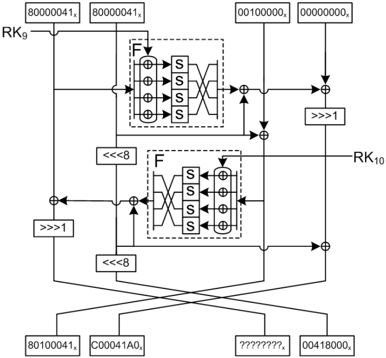


- For each pair of plaintexts and their corresponding ciphertexts (C^i, C^{i*}), increment the counter for the corresponding candidate subkey RK_{10} when the following equations holds:

$$F(C_0^i, RK_{10}) \oplus F(C_0^{i*}, RK_{10}) \oplus 00004180_x = 80000041_x \oplus (\Delta C_2^i <<< 1).$$

- Adopt the key with the highest counter as the right key.

Attack Procedure



- For each pair of plaintexts and their corresponding ciphertexts (C^i, C^{i*}), increment the counter for the corresponding candidate subkey RK_{10} when the following equations holds:

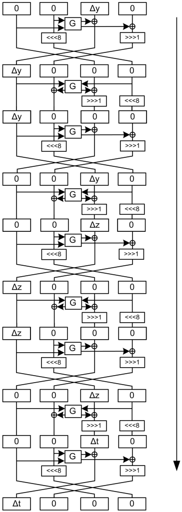
$$F(C_0^i, RK_{10}) \oplus F(C_0^{i*}, RK_{10}) \oplus 00004180_x = 80000041_x \oplus (\Delta C_2^i <<< 1).$$

- Adopt the key with the highest counter as the right key.

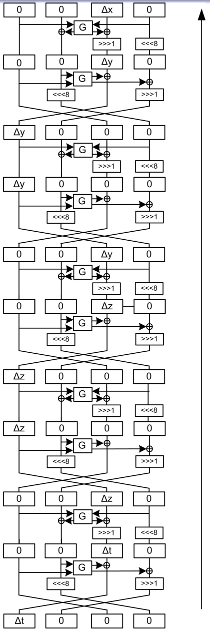
Outline

1. Description of TWIS
2. Differential Cryptanalysis
3. **Impossible Differential Analysis**
4. Observations
5. Conclusion

Impossible Differential Characteristic

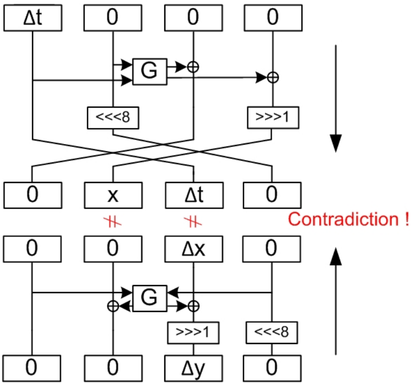


- Start with the difference $(0, 0, \Delta y, 0)$, $\Delta y = 00800000_x$
- Propagate this difference for 4.5 rounds
- Obtain the difference $(\Delta t, 0, 0, 0)$, $\Delta t = 00200000_x$
- 4.5-round differential characteristic with probability 1



- Start with the difference $(\Delta t, 0, 0, 0)$, $\Delta t = 00200000_x$
- Propagate backwards for 5 rounds
- Obtain the difference $(0, 0, \Delta x, 0)$, $\Delta x = 01000000_x$
- 5-round differential characteristic with probability 1

Impossible!



$$\Delta t = 00200000_x \neq 01000000_x = \Delta x$$

Possible Attack

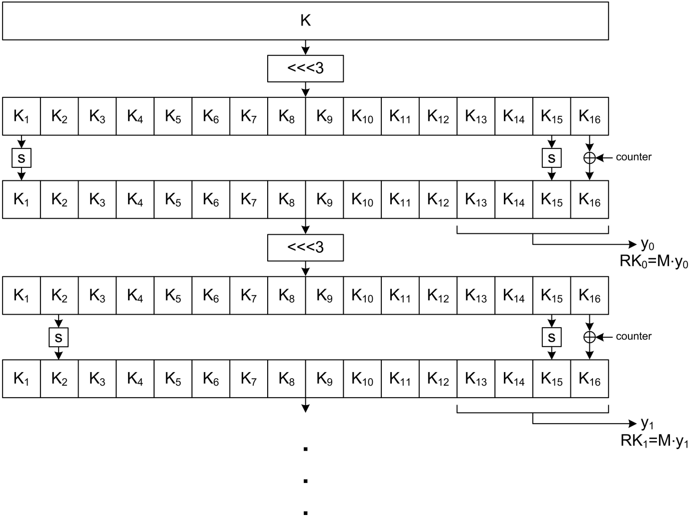
- Add half round to this characteristic
- Guess the corresponding subkeys
- Eliminate the wrong key values

Outline

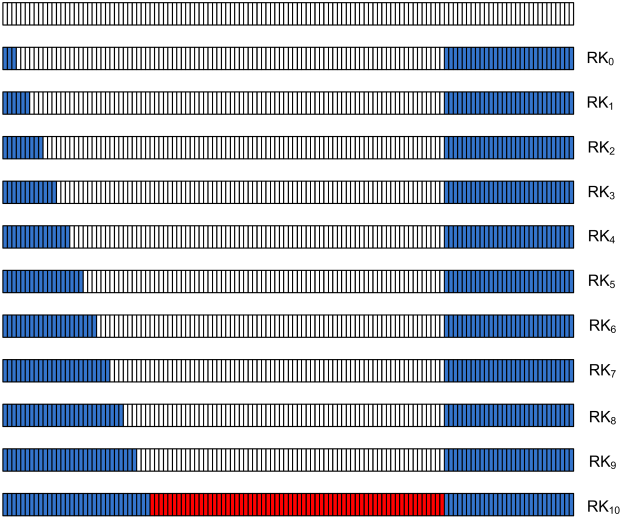
Actual Key Size

- The key size of TWIS is 128 bits.
- However, not all the bits are used to generate subkeys:
  - First subkey is generated using the first 3 and last 29 bits
  - Remaining 10 subkey is generated by 3 left rotation

Key Schedule



Actual Key Size



Actual Key Size

- The key size of TWIS is 128 bits.
- However, not all the bits are used to generate subkeys:
  - First subkey is generated using the first 3 and last 29 bits
  - Remaining 10 subkey is generated by 3 right rotation
  - So, $3 + 29 + 3 \cdot 10 = 62$ bits of the master key is used
- Therefore, the security is 62 bits.

Actual Key Size

- The key size of TWIS is 128 bits.
- However, not all the bits are used to generate subkeys:
  - First subkey is generated using the first 3 and last 29 bits
  - Remaining 10 subkey is generated by 3 right rotation
  - So, $3 + 29 + 3 \cdot 10 = 62$ bits of the master key is used
- Therefore, the security is 62 bits.
  - The key scheduling uses the same S-Box with data processing.
  - Considering the eliminated bits by the S-Boxes, the security reduces to 54 bits.

Actual Subkey Size

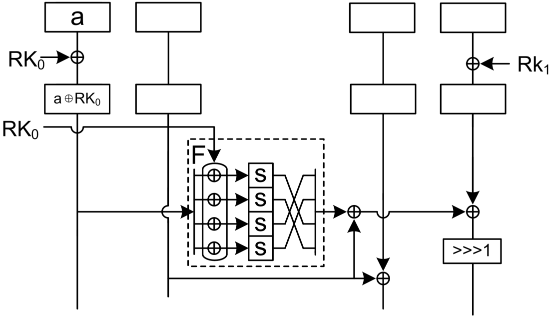
- Also, the S-Box in the F-function eliminates the first two bits of the subkey.
- Therefore, the actual subkey size is 24 bits.

Key Whitening

The key whitening, which is introduced to increase security, is used in an in aproppraте manner:

- RK_0 is XORed to the first 32-bit word.
- Then, this word is input to the F-function immediately where RK_0 is XORed again.

Key Whitening



Key Whitening

The key whitening, which is introduced to increase security, is used in an inappropriate manner:

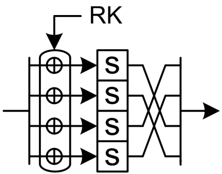
- RK_0 is XORed to the first 32-bit word.
- Then, this word is input to the F-function immediately where RK_0 is XORed again.
- Therefore, key has no effect in the first G-function: one can proceed without knowing the key.

Key Whitening

- Moreover, as the key whitening, RK_2 is XORed to the 32-bit word that is affected by RK_{10}.
- If one can find both RK_2 and RK_{10}, he can get information about the subkeys inbetween by going forwards and backwards from RK_2 and RK_{10} respectively.

Weak Diffusion

- The diffusion of the keys among S-Boxes is very weak.
- One can analyze the 32-bit subkey as 4 independent 8-bit subkeys.
- The complexity of an ordinary exhaustive exhaustive search will be 2^{24}.
- If, the search is on 4 8-bit subkeys, the complexity will be $4 \cdot 2^6 = 2^8$.



Outline

Conclusion

- A differential attack on full-round TWIS
- Recover 12 bits of the 32-bit final subkey with 2^{21} complexity
- 9.5-round impossible distinguisher
- At most 54-bit security
- Weaknesses due to the use of subkeys during the encryption and the choice of whitening subkeys

Thank you for your attention!

Questions?