Side channel collision attack on TWINE-80 and DES with reduced masked rounds

Student Name: Neha Gupta

IIIT-D-MTech-CS-IS
June, 2015

Indraprastha Institute of Information Technology
New Delhi

Thesis Committee
Dr. Donghoon Chang (Chair)
Dr. Somitra Sanadhya
Dr. Sourav Mukhopadhyay

Submitted in partial fulfilment of the requirements
for the Degree of M.Tech. in Computer Science,
with specialization in Information Security

Keywords: TWINE, DES, Hamming weight, Collision attack, Masking, Differential characteristic, Side channel cryptanalysis

Certificate

This is to certify that the thesis titled **"Side channel collision attack on TWINE-80 and DES with reduced masked rounds"** submitted by **Neha Gupta** for the partial fulfilment of the requirements for the degree of *Master of Technology* in *Computer Science & Engineering* is a record of the bonafide work carried out by her under our guidance and supervision in the Security and Privacy group at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Dr. Donghoon Chang
Indraprastha Institute of Information Technology, New Delhi

Abstract

In this work, we present the first side channel collision based key recovery attack on TWINE block cipher with 80-bit secret key and also present the improved version of the work done by Jongsung et al. in [14] on DES. We focus on TWINE-80 security when the first 7, 8 and 9-rounds of the cipher are masked. Our 7-masked round attack requires the lowest measurements ($2^{22.58}$) and can recover 12-bits of the secret key. In our 8-masked round attack, we can find 24-bits of the secret key with $2^{32.58}$ measurements whereas in our 9-masked round attack, we are able to find 40-bits of the secret key information with $2^{46.17}$ measurements. The fact that encryption and decryption functions of TWINE-80 are similar can be utilized to launch the above attacks when the last 7, 8 and 9 rounds of the cipher are masked. Thus, we show that atleast 20 rounds of TWINE-80 need to be masked to ensure security against side channel leakage. The differential characteristics constructed to demonstrate our attacks are new and hitherto not been reported before for TWINE-80.

In our work on DES, we improved the 7-round masked attack in [14] using one more charachteristic mentioned in [15] and we recover full round 48-bit subkey of the first round. The data complexity of our attack is $2^{36.99}$. The time complexity is $2^{36.99}$ measurements and $2^{35.99}$curve comparisons.

Acknowledgments

I would like to express my deepest gratitude to my advisor Dr. Donghoon chang for his guidance and support. The quality of this work would not have been nearly as high without his well-appreciated advice. I would also like to thank Mohona and Monika for devoting her time in discussing her idea with me and giving her invaluable feedback.

I would like to dedicate this thesis to my loving and supportive parents who have always been with me, no matter where I am.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Side channel attacks (SCA), proposed by Kocher [16,17], exploit weaknesses in the physical implementation of block ciphers to recover the secret key information. These attacks treat ciphers as grey box and capitalize on the side channel leaks such as *timing information, power consumption, electromagnetic leaks* etc. to correlate them with the internal states of the processing device which are dependent on the secret key. Several flavors of side channel attacks have been proposed depending upon the type of leakage investigated. Among them, power attacks are the most popular and extensively studied class of attacks. Power attacks analyze the power consumption of a cryptographic hardware device to recover the secret information. Several types of power attacks exist, e.g., simple power analysis (SPA) [17] [21], differential power analysis (DPA) [17] [18], higher order DPA attacks [11] [1] etc. In SPA attacks, the attacker exploits the relation between the operations executed and the corresponding power consumed. He visually analyzes the power traces over a period of time and then tries to map the variations in power consumption to specific operations, e.g., multiplication operations, xoring operation with key bytes etc. In DPA attacks, the attacker studies the correlation between the intermediate processed data and the side-channel output. In it power traces are collected and statistically correlated to make guesses at the secret key to derive the correct one. In this work, we study one such variant of side channel power attack termed as - *side channel collision attack.*

1.1 Notations

We use the following notations in this chapter:

ΔX	:	byte input difference
s_i	:	denotes i^{th} sbox
SK_l	:	l^{th} round subkey
WK_m	:	m^{th} whitening subkey

1.2 Background and Related work

In side channel collision (SCC) attacks, the aim of the attacker is to detect collision between two intermediate values by comparing their power consumption traces and recover information about the secret key. It is assumed that in case of a collision, since identical instructions have been executed, power consumed would be same and hence power traces would be similar [5]. SCC attack for block ciphers was first proposed in [23] by Schramm et al. when they applied the technique on DES to recover 10.2 bits of the secret key. Their attack on DES was later improved by Ledig et al. in [19]. This attack was also investigated against AES in [23]. Bogdanov et al. further improved the collision attack on AES in terms of reduced measurements by proposing variants of SCC attack such as linear collisions [6], algebraic collisions [9] and collisions based on multiple differentials [7]. In [3], SCC attack on AES based MAC construction was presented. Following these, security of block ciphers against SCC attacks was evaluated under circumstances when countermeasures such as masking were available. Biryukov et al. in [5] showed that full 10 rounds of AES-128, 10 out of 12 rounds of AES-192 and 10 out of 14 rounds of AES-256 need to be masked to guarantee resistance against side channel attacks. New variants of collision attack, termed impossible collision attack and multiset collision attack, were proposed and utilized in this work.

In [14], Kim et al. showed SCC attack on DES with first 5, 6 and 7 rounds masked and suggested full round masking to prevent these attacks. Let us discuss 7-masked rounds attack presented by Jongsung et al. in [14] which suggests full round masking to prevent DES from side channel attacks. In this attack, authors considered a 2-round iterative differential characteristic mentioned in [2] of probability $\approx 2^{-7.87}$. They construct 7-round differential characteristic from this 2-round iterative charcteristic with probability $2^{-31.5}$ using the characteristic three and half times having input output difference both 00 00 00 00 19 60 00 00(shown in Fig. 3.1).

The basic idea of the attack is that if a plaintext pair satisfy input output difference both of the differential characteristic, then that pair is considered as right pair and we use that pair for key filtering in first round. For each right pair we get 32-bit collision after F function in 8th round which means wrong pairs can be fitered out with probability 2^{-32}. To increase the wrong pair filtering rate, we can filter out in 9th round also after F function. Since the difference for the differential characteristic is 19 60 00 00 in 9th round, so we can get collision at 22 bits after E expansion box with probability 2^{-22}, therefore we can filter out all wrong pairs at filtering rate 2^{-54}. The attack procedure is as follows:

1. Choose $2^{34.5}$plaintext pairs with satisfying input difference 19 60 00 00 and encrypt then with DES first 7 and last 7 masked rounds.

2. For each plaintext pair,collect it's power traces and check for collision at input positions of $s_1, s_2, s_3, s_4, s_5, s_6, s_7$ and s_8 s-boxes in 8th round and if collision doesn't occur discard that pair.

3. For remaining pairs check for collision at input positions of s_4, s_5, s_6, s_7 and s_8 s-boxes in

9th round and if collision doesn't occur discard that pair.

4. Now using those pairs which satisfy above test, analyse s_1, s_2 and s_3 s-boxes in the first round using difference distribution table which suggest key candidates because s_1, s_2 and s_3 are active in the first round, active means s-box having nonzero input difference.

5. output key which give maximum hits.

Right key will give 8 hits always as 8 plaintext pairs expected to be right pairs, since the differential characteristic holds with probability $2^{31.5}$ and we choose $2^{34.5}$ plaintext pairs and the filtering rate is 2^{-54}, so there will be no wrong pair out of $2^{34.5}$ chosen plaintexts. Now, we have to calculate that wrong key will give how many hits.

In case of each active s-boxes s_1, s_2 and s_3 respectively, there will be 2, 2 and 2 equivalent keys. Suppose, for s_1 we have a set S of 14 plaintext pairs, then equivalent means $\{a \cdot b | b \in S\}$, thus $S = 03_x \cdot S$. So $k, k \oplus 03_x$ are equivalent where k is any key candidate suggested through difference distribution table of s_1. For, any right pair following the differential characteristic, 20 keys will never be suggested, so 20 wrong keys will give 0 hits and since we are remaining with 44 keys out of which one is correct and 2 will be equivalent to it, so 42 will be wrong keys. Therefore wrong key hits $= (14t - 2t)/42$ hits, where t is number of right pairs i.e 8. So, wrong key will give 2.29 hits.
In case of s_2, 14 keys will be suggested only and out of which 12 will be wrong keys as two are equivalent keys i.e $k, k \oplus 32_x$ and 8 plaintext pairs satisfies the input output difference of s_2, therefore wrong key will give $= (8t - 2t)/12$ i.e 4 hits.
In case of s_3, 22 keys will be suggested only and out of which 20 will be wrong keys as two are equivalent keys i.e $k, k \oplus 2c_x$ and 10 plaintext pairs satisfies the input output difference of s_3, therefore wrong key will give $= (10t - 2t)/20$ i.e 3.2 hits.
So, finally through this differential characteristic, they were able to recover 15(5+5+5) bit key information using s_1, s_2 and s_3.
Therefore the data complexity of attack is $2^{35.5}$ chosen plaintexts. The time complexity is calculated as : $2^{35.5}$ measurements and $2^{34.5}$ curve comparisons, so the total time complexity is $2^{35.5}$. Therefore, they suggested full round masking of DES to prevent from side channel attacks.

In [5], Biryukov et al. showed side channel collision attack on AES with 2-rounds masked. They performed this attack using 2-round differential characteristic of probability 2^{-6}(shown in Fig. 1.1).

The attack procedure is as follows:

1. Choose 4-active bytes in the first round as shown in Fig. 1.1.

2. Then, 4-active bytes in the first round reduce to 3-active bytes in the second round with probability $2^{-8} \times 4 = 2^{-6}$.

3. After that, 3-active bytes are expanded into 12-active bytes in the third round with probability $p = 1$.

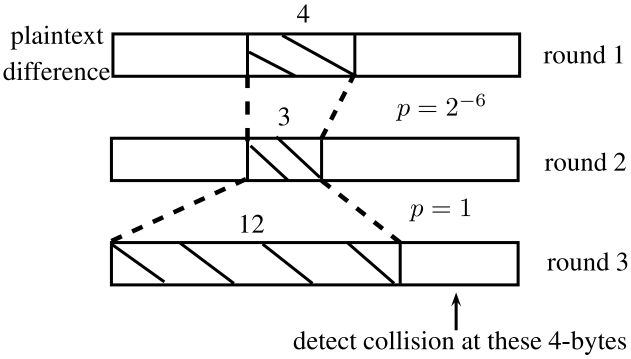


Figure 1.1: 2-round diffrential characteristic

4. Now, the aim is to find collision in the rest 4-passive bytes in the third round as shown in Fig. 1.1.

5. The plaintext pair which satisfies the input difference and gives collision at the 4-passive bytes in the third round with probability 2^{-6} is called *right pair.*

6. They used structures to reduce data complexity i.e take 24 texts having same random constants at the place of passive bytes in the first round. Then generate 2^8 plaintext pairs using 24 texts.

7. They got $2^8 \times 2^{-6} = 4$ *right pairs.*

8. One right pair is responsible for reducing the key space by 8-bits [23], so 4 *right pairs* will reduce the key space by 32-bits.

In this attack, overall measurements calculated as follows: *online measurements=72*(as 24×3 texts are required to get 96-bits of key information) and *offline measurements* $\approx 2^{32}$(as rest 32-bits of key are found through exhaustive search).

Biryukov et al. also showed impossible collision attack on AES with 3-rounds masked using 3-round differntial characteristic of probability 2^-22(shown in Fig. 1.2).

The attack procedure is described as follows:

1. Choose 4-active bytes in the first round as shown in Fig. 1.2.

2. Then, 4-active bytes in the first round reduce to 1-active bytes in the second round with probability 2^{-22}.

3. After that, 1-active bytes are expanded into 4-active bytes in the third round with probability $p = 1$.

4. Then 4-active bytes are expanded into 16-active bytes after third round with probability $p = 1$.

5. Now, the idea is to detect impossible collision at these 16-active bytes as shown in Fig. 1.2.

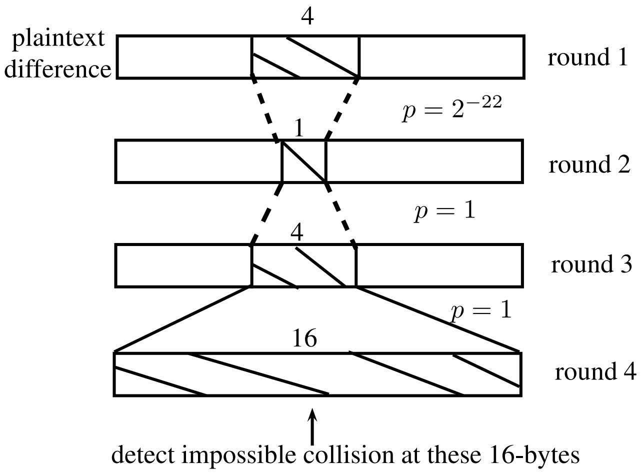


Figure 1.2: 3-round diffrential characteristic

The probability that atleast one byte out of 16 bytes is same $= 1-(1-2^{-8})^{16} \approx \frac{1}{16}$. Therefore, the wrong pair survival rate is $\frac{15}{16}$ which is very high and they need to filter out these wrong pairs.

For each tested pair, they constructed its variation by varying only the passive bytes and keeping active bytes same, so total 16 such variations were possible which significantly reduced the wrong pair survival rate to $(1-2^{-8})^{16 \cdot 16} \approx 0.36$. Then repeat this for all tested remaining pairs which pass first filtering again and again untill get a *right pair* which reduces the key space by 8-bits [23].

For efficient measurements, they used structures of 2^{12} texts which generate 2^{23} plaintext pairs. Since the filtering rate is low, therefore they considered $2^7 - 2^8$ such structures as $\frac{15}{16}^{256} \approx 2^{-23.8}$ which means it will filter out $2^{23} - 2$ wrong pairs. Therefore overall measurements required $2^{19} - 2^{20}$ and 16×2^{23} curve comparisons(as 16 trials are needed for each 2^{23} pairs).

In [5], authors also discussed another type of side channel attack on AES called multiset collision attack using 3 and 4-round distinguisher.

Lee et al. applied impossible collision attack on HIGHT in [20] to recover secret key bits when the first 11, 12 & 13 rounds are masked. Let us discuss impossible collision attack when the first 11 rounds are masked.

They constructed a 10-round truncated differential characteristic of probability $p = 1$(as shown in Fig. 1.3 which was never used before and used it to present 11, 12 and 13 round attack on HIGHT.

The attack procedure is as follows:

1. Attach one round at the front of 10-round distinguisher(shown in Fig. 1.4) and masked 10-rounds from 2^{nd} to 11^{th} round and rest rounds were unmasked.

2. Guess the subkey of the attached round and collect t plaintext pairs which satisfy the

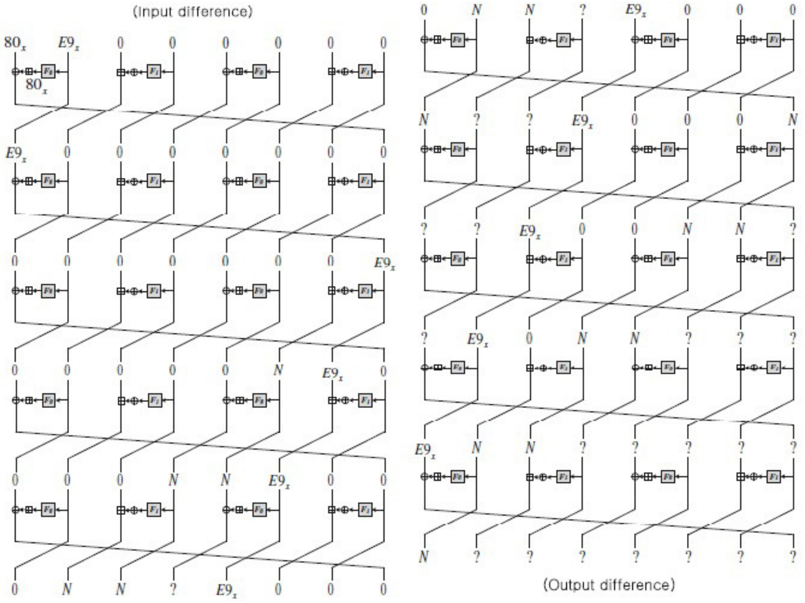


Figure 1.3: Truncated differential of 10-round

input difference of the second round(as 2^{nd} round input differnce depends only on WK$_2$, WK$_3$, SK$_2$ and SK$_3$, so we need to guess 2^{32} keys(shown in Fig. 1.4)).

3. Then test each t pair that we get nonzero after 11^{th} round at 1^{st} byte i.e $\Delta X_{11,7} \neq 0$, if this condition satisfies then the guessed key is correct otherwise discard that key.

Since the probability of getting impossible collision at a byte is $1 - 2^{-8}$, if we take $t = 2^{13}$, then probability that atleast one pair out of 2^{13} gives collision is $1 - (1 - 2^{-8})^{2^{13}}$(probability of filtering out wrong key).

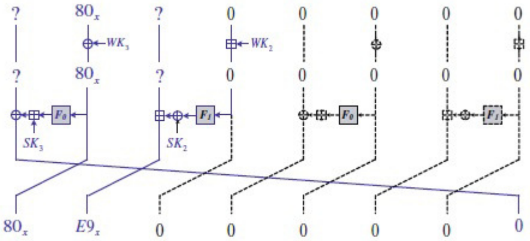


Figure 1.4: Extended one-round backward

The probability of filtering out all $2^{32} - 1$ wrong keys is, $(1 - (1 - 2^{-8})^{2^{13}})^{2^{32}-1} \approx 0.99$. Thus, the data complexity is $t \cdot 2^{32} = 2^{45}$, so to reduce the data complexity, they constructed two structures of 16 plaintexts each as:

$S = S(i, j)$ and $S' = S(i', j')$

$S = i\|x_0\|\|j\|x_1\|x_2\|x_3\|x_4\|x_5$
$S' = i\|x_0 \oplus 08_x\|\|\|j\|x_1\|x_2\|x_3\|x_4\|x_5$ where (i,j)ϵ $\{00_x....FF_x\}$. The attack procedure is same as described above except that we select plaintext pair from S and S'. The data complexity is

reduced to $2 \cdot 2^{16} = 2^{17}$ and curve comparisons is 2^13 and time measurements is $2^{13} \cdot 2^{32} = 2^{45}$.

To reduce computations, authors improved 11-round attack as follows:

Instead of guessing all 2^{32} keys, they calculated two values α and β where

$\alpha = F_1(x_1 + WK_2) \oplus SK_2$ and $\beta = (F_0(x_0 + WK_3) \oplus SK_3) \oplus (F_0((x_0 \oplus 08_x) + WK_3) \oplus SK_3)$, they found that for many guessed keys, α and β were same, so they called that keys to be equivalent keys and finally they found 128 distinct α and 24 distinct β. Then, they calculated two structures $S = S(i, j)$ and $S' = S(k, l)$ as:

$S = i \| x_0 \| \| j \| x_1 \| x_2 \| x_3 \| x_4 \| x_5$ and $S(k, l)$ corresponding to plaintext $S(i, j)$ is computed as follows:

$k = \beta \oplus i$ and $l = (E9_x \oplus (\alpha + j)) - \alpha$ and using this information the attack procedure is as follows:

1. Select a plaintext pair from above defined S and S'.
2. Guess α and β from 128 and 24 possible values respectively.
3. Then test each selected plaintext pair that we get nonzero after 11^{th} round at 1^{st} byte i.e $\Delta X_{11,7} \neq 0$, if this condition satisfies then the guessed key is correct otherwise discard that key.

If we take $t = 2^{12}$, then probability that atleast one pair out of 2^{12} gives collision is $1 - (1 - 2^{-8})^{2^{12}}$(probability of filtering out wrong key).

As the guessed values are $2^{11.6}$(128 α and 24 β), the probability of success i.e recovering all 11.6 secret bit information is $(1 - (1 - 2^{-8})^{2^{12}})^{2^{11.6}-1} \approx 0.99$.

So, we need 2^{12} curve comparisons and $2^{11.6} \cdot 2^{12} = 2^{23.6}$ time complexity .

Similarly, by extending 10-round truncated differential(shown in Fig. 1.3) by 2 and 3-round backward, they performed 12 and 13-round attack respectively with the data complexity 2^{32} and 2^{40} chosen plaintexts respectively.

Chapter 2

Side channel collision attack on TWINE-80

The field of lightweight cryptography encompasses the current state-of-the-art cryptographic algorithms designed for implementation in constrained environments (e.g., RFID tags, sensors, smartcards etc.) and addresses the security concerns of low cost devices. With the growing interest of symmetric cryptographic community in this field, several lightweight variants of traditional cryptographic primitives such as *lightweight block ciphers* - PRESENT [8], HIGHT [13], LED [12], TWINE etc. have been proposed and studied in literature.

In this work, we focus on the side channel security of one such lightweight block cipher - TWINE which was proposed by Suzaki at al. in SAC 2012 [25]. TWINE is a 64-bit block cipher which supports two key sizes - 80-bit and 128-bit and consists of 36 rounds. It is a type-2 based Generalized Feistel Structure (GFS) [22] with 16 branches. The high branch number allows TWINE to have very small input-output branch size (4-bits), thus enabling it to be used in extremely small hardware settings. At the same time, such a design ideology also suffers from slow diffusion property since a k-branch GFS requires k-rounds to achieve full diffusion. This may provide vantage point to an attacker to attack large number of rounds. To mitigate this limitation, the designers of TWINE adopted an improved permutation configuration as suggested by Suzaki et. al in [24]. This improved permutation allowed TWINE to achieve full diffusion in just 8 rounds. These above mentioned properties enable TWINE to achieve reasonably good performance in both hardware and software implementations under lightweight settings [4,10,25]. Its each round consists of 8 F-functions, where one F-function (as shown in Fig. 2.1) comprises of subkey (SK) xoring and 4-bit S-box (S) operation. It is followed by a diffusion layer in which all the 16 branches are shuffled using the permutation shown in Table 2.2. The key schedule algorithm of TWINE-80 generates 36 32-bit round keys where each 32-bit round key is split into 8 4-bit subkeys which enter the corresponding 8 F-functions in each round. The first round key is directly derived from the 80-bit master key. For further details, one can refer [25].

The main contributions of this work are as follows:

Table 2.1: Summary of our results on TWINE-80

Rounds attacked	Key bits recovered	Time complexity	Data complexity	Reference
7	12	$2^{22.58}$	$2^{22.58}$	This work, § 2.3
8	12	$2^{26.58}$	$2^{26.58}$	This work, § 2.6
8	24	$2^{32.58}$	$2^{32.58}$	This work, § 2.4
9	12	$2^{32.58}$	$2^{32.58}$	This work, § 2.6
9	40	$2^{46.17}$	$2^{46.17}$	This work, § 2.5

- We present the first side channel analysis of TWINE-80 block cipher to recover the secret key.
- We present side-channel collision attack on TWINE-80 when the first 7, 8 and 9 rounds of the cipher are masked.
- Our 7, 8 and 9-masked round attacks recover 12, 24 and 40-bits of secret key respectively with $2^{22.58}$, $2^{32.58}$ and $2^{46.17}$ measurements respectively.
- We also report 8 and 9-masked round attacks with lower data and time complexities but at the expense of lesser key bits recovery. In our 8-masked round attack, we can find 12-key bits of the secret key with $2^{26.58}$ measurements whereas in our 9-masked round attack, we can recover 12 secret key bits with $2^{32.58}$ measurements. This reduction depends upon the type of differential characteristic chosen for the attack.
- We construct new (previously unreported) differential characteristics for TWINE-80 to carry out all our attacks.
- In our above attack results, first $r = 7, 8, 9$-rounds are masked. The same attack can be applied in the last r-masked rounds with similar complexity as well. Therefore, our results show that a powerful attacker possesses the capability to recover secret key bits even when the first 9 and the last 9 rounds of TWINE-80 are masked and hence, atleast 20 rounds of TWINE-80 should be masked to guarantee resistance against side channel leakage.

Our summarized results are shown in Table 2.1.

Table 2.2: Permutation table of TWINE-80

j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\pi(j)$	5	0	1	4	7	12	3	8	13	6	9	2	15	10	11	14
$\pi^{-1}(j)$	1	2	11	6	3	0	9	4	7	10	13	14	5	8	15	12

In this chapter, first we will discuss about the basic layout of our attack on TWINE-80 which is necessary to understand the attack procedure. Then later, we will present attack on TWINE-80

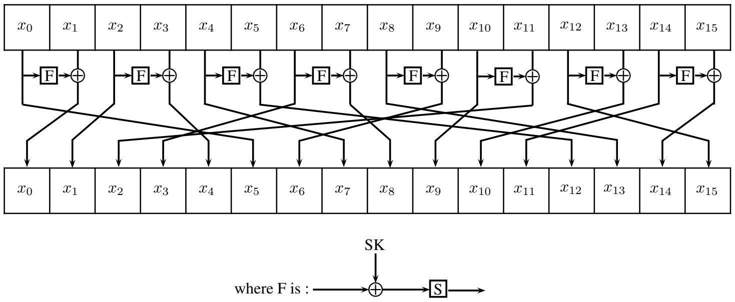

x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12} x_{13} x_{14} x_{15}
F
x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12} x_{13} x_{14} x_{15}
SK
where F is :
S

Figure 2.1: One round of TWINE

with first 7, 8 and 9 masked rounds where we can extract maximum number of key bits. In our attack, we assume that the attacker can detect collisions in the intermediate state positions by analyzing the power traces [19].

2.1 Notations

We use the following notations in this chapter:

$\#R$	:	Round number ($1 \leq R \leq 36$)
$\#R^i$	:	i^{th} input branch of round R ($1 \leq i \leq 16$)
Δ	:	branch input difference where $\|\Delta\| = 4$ bits
F^l	:	l^{th} F-function in a round ($1 \leq l \leq 8$)
Δ_{in}^F	:	F-function input difference where $\mid \Delta_{in}^F \mid = 4$-bits
Δ_{out}^F	:	F-function output difference where $\mid \Delta_{out}^F \mid = 4$-bits
Δ_{in}	:	input difference of differential characteristic where $\mid \Delta_{in} \mid= 64$-bits
Δ_{out}	:	output difference of differential characteristic where $\mid \Delta_{out} \mid= 64$-bits

2.2 High level layout of the attack

Our attack consists of two phases: *online phase and offline phase.*

Online phase : *Detection of right pairs*

In the online phase :

1. We first choose an appropriate differential characteristic over first r-rounds ($\Delta_{in} \xrightarrow{r} \Delta_{out}$) such that collisions occur in some nibble positions in the output after r-masked rounds, i.e., $\Delta = 0$ at these positions. Such a differential characteristic holds with some probability 2^{-p} over r-rounds.

2. We then collect 2^t plaintext pairs satisfying the input difference (Δ_{in}) of the differential characteristic (where $t > p$) such that we get 2^{t-p} right pairs which follow the differential trail over r-rounds. We call other pairs to be wrong pairs, i.e., pairs for which ($\Delta_{in} \not\rightarrow \Delta_{out}$) over r rounds.

3. To detect the right pairs, we collect the power traces of all 2^t plaintext pairs (2^t measurements) and perform curve comparisons to check whether collisions at pre-determined positions happen or not. If collisions do not happen, we discard that pair.
4. The pairs which pass the above step are analyzed in the offline phase.

Offline phase : *Key detection phase*

In the offline phase:

1. We use the right pairs obtained in the online phase to examine the first round of the cipher.
2. In the first round, we select F-functions in which underlying Sboxes are active (i.e., $\Delta_{in}^{F} \neq 0$). We call such F-functions as active F-functions.
3. For each active F-function, depending on the right pair chosen and the difference distribution table of the Sbox, certain 4-bit subkey candidates will be suggested (out of the 16 possible subkey values). For example, for a given right pair (P, P') and F-function F (as shown in Fig. 2.2) for which Δ_{in}^{F} and Δ_{out}^{F} are known to the attacker, those subkeys which satisfy the relation $S(P \oplus SK) \oplus S(P \oplus \Delta_{in}^{f} \oplus SK) = \Delta_{out}^{f}$ will be suggested as candidate subkeys.
4. For each right pair, a different set of subkey candidates will be suggested for each active F-function.
5. Once all the right pairs are tested, it is expected that among the set of subkeys suggested for one active F-function, only a single subkey will be common in all the 2^{t-p} sets. That subkey will be the actual subkey for the corresponding F-function. The same will hold true for other active F-functions as well.

If more than one potentially correct key candidates are suggested. In the offline phase, it may happen that more than one correct subkey candidates are suggested for few active F-functions (i.e., more than one subkey candidates will be common in all the 2^{t-p} sets). Such subkeys will be called *equivalent subkeys.* In order to identify the actual subkey among the equivalent subkeys, we then choose a second differential characteristic that should differ from the first characteristic at positions where the corresponding active F-functions suggested equivalent subkeys. We repeat the whole attack procedure again with this second differential characteristic. This will provide us a new set of subkey candidates with respect to those active F-functions. The key candidate which is common in both the older and the newer subkey sets will be our actual secret subkey.

Attack on the last r-masked rounds. The above described attack is a chosen plaintext attack in which the first r-rounds are masked. Since TWINE-80 decryption function is exactly the same as the encryption function except inverse block shuffle being used during decryption,

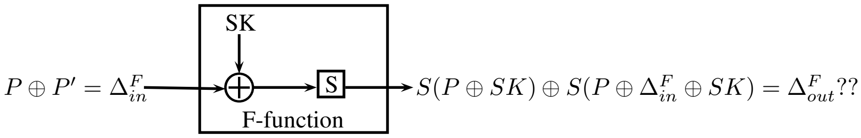

SK
S
F-function
$P \oplus P' = \Delta^F_{in}$
$S(P \oplus SK) \oplus S(P \oplus \Delta^F_{in} \oplus SK) = \Delta^F_{out}??$

Figure 2.2: Suggested subkey candidates

the same attack procedure can be applied in the last r-masked rounds as well with similar attack complexity but with chosen ciphertexts. Thus, a powerful attacker possesses the capability to recover secret key bits even when the first r and the last r rounds of TWINE-80 are masked.

2.3 Detailed attack on TWINE-80 with $r = 7$

In this section, we discuss the detailed attack on TWINE-80 with 7-masked rounds in which we can recover 12-secret key bits. The differential characteristics used in this attack are new and have not been reported in any of the previously existing TWINE-80 attacks.

Let us first discuss some key points which will facilitate better understanding of the attack subsequently:

- We choose the initial plaintext difference Δ_{in} =0 0 0 0 0 E 0 1 4 C E C E 0 0_x. This differential characteristic will give an output difference of Δ_{out} = 0 0 0 0 1 0 E 0 0 0 0 0 0 C 0 0_x with a probability of 2^{-18} after first 7-masked rounds (as shown in Fig. 2.3).[1]
- From Fig. 2.3, we can see that collisions occur at $\#8^1$, $\#8^2$, $\#8^3$, $\#8^4$, $\#8^6$, $\#8^8$, $\#8^9$, $\#8^{10}$, $\#8^{11}$, $\#8^{12}$, $\#8^{13}$, $\#8^{15}$, $\#8^{16}$, $\#9^1$, $\#9^2$, $\#9^3$, $\#9^5$, $\#9^6$, $\#9^7$, $\#9^{10}$, $\#9^{12}$, $\#9^{14}$, $\#9^{15}$, $\#9^{16}$. We call such collisions - *0-collisions.*
- To detect these collisions, we use the Hamming weight model. We assume that if the Hamming weights of the two intermediate values at the input and output positions of the F-function are same, then collision happens and can be detected. We call such a collision - *Hamming Weight (HW)-collision.* It can be easily verified that *0-collision* $\Longrightarrow$ *HW-collision.* As mentioned earlier, we assume that such a collision can be detected by an attacker through curve comparisons. With this assumption, we thus check Hamming weight collisions at $\#8^1$, $\#8^3$, $\#8^9$, $\#8^{11}$, $\#8^{13}$, $\#8^{15}$, $\#9^1$, $\#9^3$, $\#9^5$, $\#9^7$, $\#9^{15}$ (shown in squared boxes in Fig. 2.3), i.e., total 11 positions.
- The probability that for any random plaintext pair which satisfies Δ_{in}, *HW-collision* occurs at all the 11 positions is $2^{-35.97}$. The details of this calculation are shown in § .2.
- Since, we are able to detect only *HW-collisions*, cases may arise where they do not correspond to actual *0-collisions*(as shown in Fig. 2.4). However, we later show that in our

[1] Readers can verify the probability calculation from the difference distribution table (DDT) of TWINE Sbox given in the § .1.

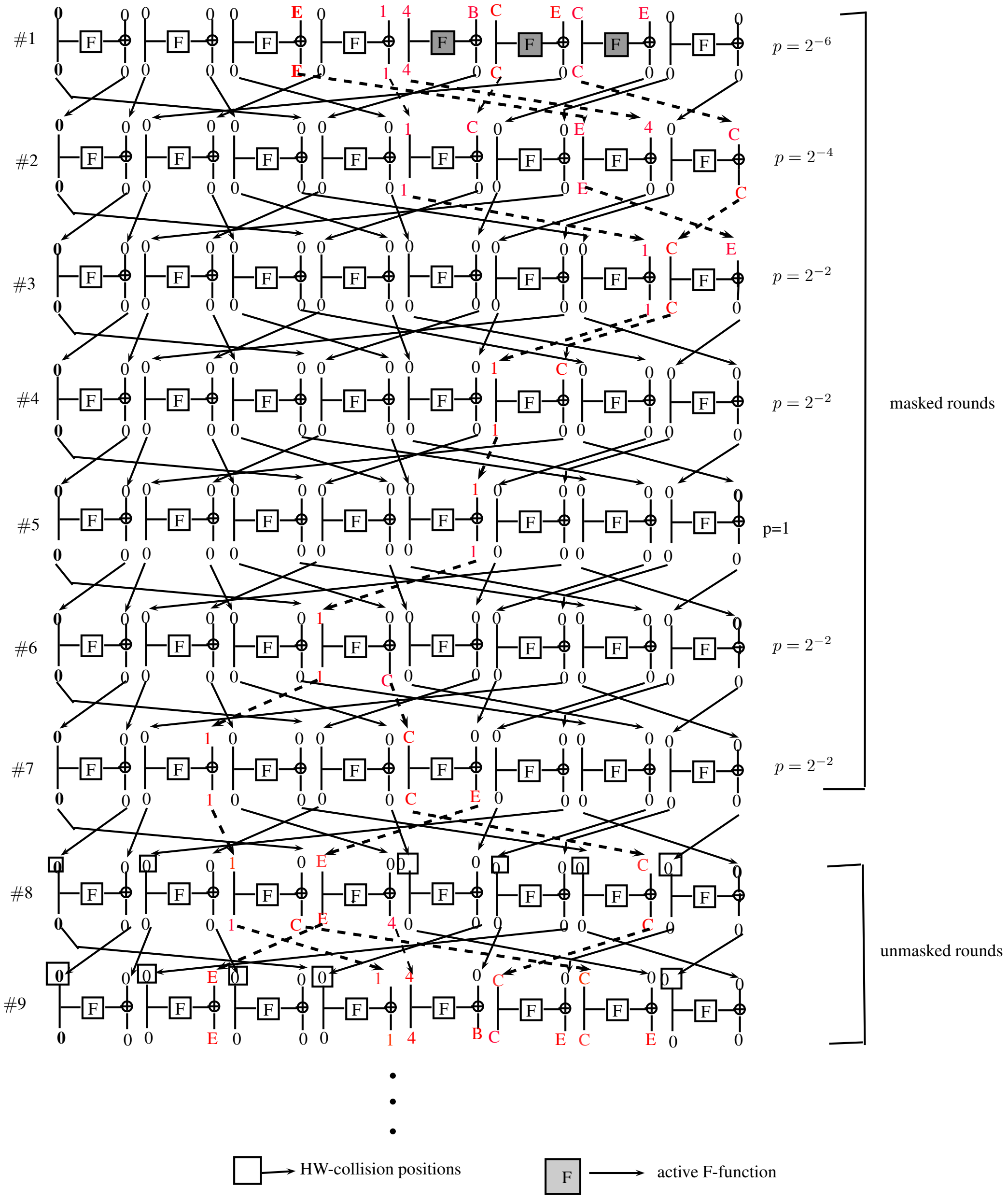
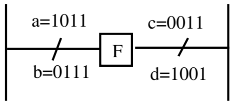


Figure 2.3: First 7-round differential characteristic for TWINE-80

attack, depending upon our choice of differential characteristic, we can safely assume that *HW-collision* so detected corresponds to actual *0-collisions* with very high probability.



$$HW(a) = HW(b) = 3 \textit{ but } a \oplus b = 1100 \Rightarrow \Delta^F_{in} \neq 0000$$
$$HW(c) = HW(d) = 2 \textit{ but } c \oplus d = 1010 \Rightarrow \Delta^F_{out} \neq 0000$$

Figure 2.4: A case where *HW-collision* $\not\Rightarrow$ *0-collision*

- As shown in Fig. 2.3, the active F-functions in the first round are F^5, F^6, and F^7 (i.e., F-functions with $\Delta^F_{in} \neq 0$). Let us consider F^5 where $\Delta^F_{in} = 4_x$ and $\Delta^F_{out} = B_x$ (as shown in Fig. 2.5). If we consider the difference distribution table of the Sbox (given in § .1), we see that 4 values exist which satisfy this input-output difference of the Sbox. These values suggest 4 subkey candidates for this Sbox. Similarly, we found that for all the other active F-functions in the first round, 4 subkey candidates are possible.

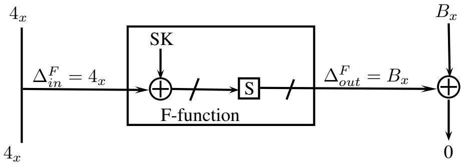


Figure 2.5: Input - Output difference of F^5

Utilizing this information we have, the attack steps are as follows:

1. We choose 2^{20} plaintext pairs having Δ_{in} =0 0 0 0 0 E 0 1 4 C E C E 0 0_x, and encrypt them with Twine-80 with the first seven rounds masked.
2. For each pair, we collect its power traces during encryption process and check for HW collision at $\#8^1$, $\#8^3$, $\#8^9$, $\#8^{11}$, $\#8^{13}$, $\#8^{15}$, $\#9^1$, $\#9^3$, $\#9^5$, $\#9^7$, $\#9^{15}$ positions (shown in Fig. 2.3 in squared boxes).
   - Since this differential characteristic holds with probability 2^{-18}, 4 right pairs ($2^{20} \times 2^{-18}$) exist. This means that 4 right pairs yield *0-collisions* and give *HW-collision* with probability 1.
   - Since, for any random pair satisfying Δ_{in}, collisions are detected with a probability of $2^{-35.97}$, thus, out of 2^{20} pairs, the number of pairs generating *HW-collision* are:

$$\begin{aligned}&= (\text{number of right pair}) \times (\text{probability of } \textit{HW-collision}) + \\ &+ (\text{number of wrong pairs}) \times (\text{probability of } \textit{HW-collision}) \\ &= (4 \times 1) + (2^{20} - 4) \times 2^{-35.97} \approx (4 \times 1) + 2^{-15.97} = 4\end{aligned}$$

   Thus, we can safely assume that the pairs detected in our attack as right pairs through *HW-collision* are pairs which correspond to actual *0-collisions*.

3. Once we get 4 right pairs, we move to the offline stage.

4. In the offline stage, as discussed in § 2.2, we analyze the 3 active F-functions (F^5, F^6 and F^7) in round 1 with each of the four right pairs. Let us consider the first active F-function F^5. As seen in Fig. 2.5, the input-output difference for this F-function are 4_x and B_x respectively. Based on the DDT table of the Sbox, the four input pairs satisfying the given input-output difference over the Sbox are - (5_x, 1_x), (1_x, 5_x), (A_x, E_x), (E_x, A_x). This gives us a set of 4 subkey candidates. We then analyzed F^5 with the second right pair to obtain another set of 4 subkeys. However, we found that the sets of 4 candidate subkeys obtained from both the first and second right pair respectively were exactly the same. This was true for each of the remaining two other right pairs as well. Occurrence of this peculiar behaviour can be explained as follows. Let us suppose that the four right pairs are labelled as - (P_1, P_1'), (P_2, P_2'), (P_3, P_3') & (P_4, P_4'). If SK denotes the correct subkey for F^5, then

$$S(P_1 \oplus SK) \oplus S(P_1 \oplus 4_x \oplus SK) = B_x \quad (2.1)$$
$$S(P_2 \oplus SK) \oplus S(P_2 \oplus 4_x \oplus SK) = B_x \quad (2.2)$$
$$S(P_3 \oplus SK) \oplus S(P_3 \oplus 4_x \oplus SK) = B_x \quad (2.3)$$
$$S(P_4 \oplus SK) \oplus S(P_4 \oplus 4_x \oplus SK) = B_x \quad (2.4)$$

Let us suppose, $P_1 \oplus SK = 5_x$, $P_2 \oplus SK = 1_x$, $P_3 \oplus SK = A_x$ and $P_4 \oplus SK = E_x$. Then,

$$S(5_x) \oplus S(1_x) = B_x \quad (2.5)$$
$$S(1_x) \oplus S(5_x) = B_x \quad (2.6)$$
$$S(A_x) \oplus S(E_x) = B_x \quad (2.7)$$
$$S(E_x) \oplus S(A_x) = B_x \quad (2.8)$$

Now, if we replace SK by $SK \oplus B_x$ in Eqs. 1-4, we see that Eqs. 1-4 again reduce to Eqs. 6-8 (though in different order). This holds true even when we replace SK by $SK \oplus 4_x$ or $SK \oplus F_x$. Hence, we call SK, $SK \oplus 4_x$, $SK \oplus B_x$ and $SK \oplus F_x$ as "equivalent keys", as all of them produce the same set of input values to the Sbox. Since, in our attack, all the four subkey sets (corresponding to each right pair) will have the correct key and its three other equivalent keys as well, we get the same set of four subkeys for each pair.

Thus, at this stage we get 4 "equivalent keys" for F^5 and cannot determine which among them is the actual key. The same property is observed in other active F-functions as well, i.e., 4 equivalent keys are obtained for each of the three active F-functions in #1. Hence, as discussed in § 2.2, we now choose a second differential characteristic and repeat the whole attack again.

2.3.1 Recovering the actual key from the set of equivalent keys

We choose the second differential characteristic as Δ_{in} =0 0 0 0 0 B 0 C D 1 E 4 4 B 0 0_x. This gives us Δ_{out} =0 0 0 0 C 0 4 0 0 0 0 0 0 E 0 0_x after 7 rounds with a probability of 2^{-19} (as shown in Fig. 2.6). Our choice of second differential characteristic is driven by two factors:

1. It should differ from the first characteristic at positions where F-function in round 1 are active.

2. For each active F-function, the new set of subkeys so obtained should suggest only one subkey (and not more than one) that is common with the corresponding older set. We then claim that key as the correct subkey.

Let us discuss how we can ensure this second condition before carrying out the actual attack. If we repeat the whole 7-round masked attack with the second differential characteristic in a manner similar to that discussed in § 2.3 and focus on F-function F^5 in #1, the new set of 4 keys so obtained will be of the form: SK', $SK' \oplus 8_x$, $SK' \oplus D_x$ and $SK' \oplus 5_x$.

We now have 2 sets of 4 subkeys for F^5 in #1 as shown in Table 2.3.

Table 2.3: Set of subkey candidates obtained for F^5 from two differential characteristics.

Set1 (from characteristic 1)	Set2 (from characteristic 2)
SK	SK'
$SK \oplus 4_x$	$SK' \oplus 8_x$
$SK \oplus B_x$	$SK' \oplus D_x$
$SK \oplus F_x$	$SK' \oplus 5_x$

Since, we do not know the actual values of these 8 subkeys at this stage, hence we cannot do an intersection operation on both sets and find out the common subkey. Instead, we xor each element in Set1 with each element in Set2 and propose the following Lemma:

Lemma 1. *Let A and B be two sets of subkey candidates obtained from differential characteristics 1 and 2 respectively. Let P be a set as defined below,*

$$P = \{x | y \oplus z = x, where, y \in A \text{ and } z \in B\}$$

Here $|P| = 16$, i.e., P will have 16 elements. If all the 16 elements in P are unique, then only one single key will be common in both the sets - A and B.

The proof of this lemma is given in § .3

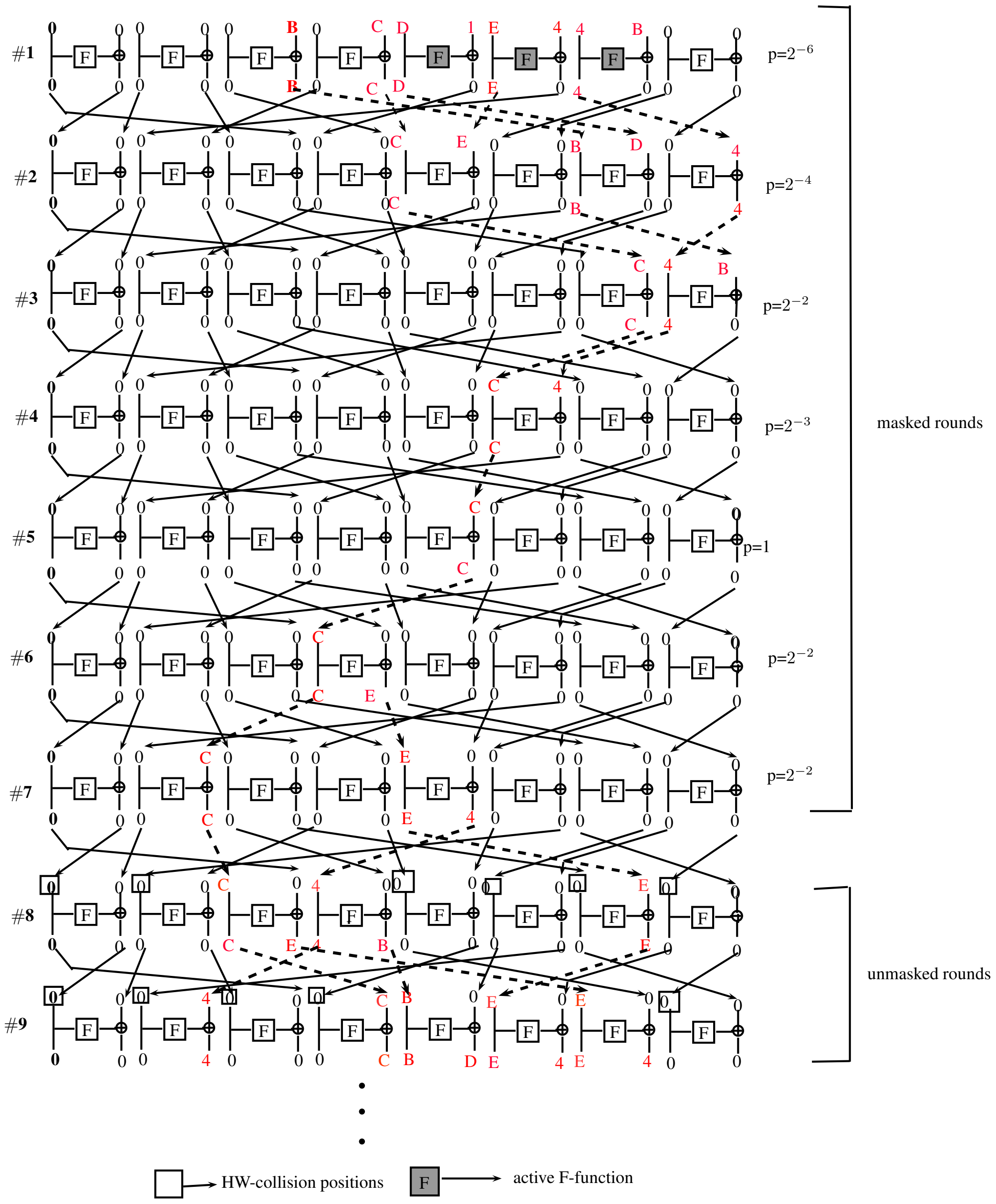


Figure 2.6: Second 7-round differential characteristic for TWINE 80

It is easy to check that in our attack, the set P constructed for the two sets - Set1 and Set2 have 16 distinct elements. Hence, only a single subkey will be obtained for F-function F^5 which will be our correct subkey. We can similarly find the actual secret subkey for each of the other active F-functions, i.e., F^6 and F^7.

Thus, in the attack, we collect 2^{21} plaintext pairs satisfying Δ_{in} and run the attack as discussed in § 2.3. We will get 12-bits of secret key information. Therefore, total key bits recovered from the first round are 12-bits.

Attack Complexity. In the attack using the first and second differential trails, we need 2^{21} and 2^{22} chosen plaintexts respectively. Therefore the total data complexity is - 2^{21}+ 2^{22}= $2^{22.58}$ chosen plaintexts. In the online stage, we trace 2^{21}+ 2^{22} power curves (measurements) and do 2^{20}+ 2^{21}= $2^{21.58}$ curve comparisons. Thus, the total time complexity is approximately $2^{22.58}$.

As mentioned in § 2.2, since TWINE-80 encryption and decryption functions are similar [25], the same attack procedure can be launched in chosen ciphertext model when the last 7 rounds are masked. This shows that an attacker can attack both top 7 and bottom 7 rounds of TWINE-80 and hence atleast 16 rounds (top 8 + last 8) of TWINE-80 should be masked.

2.4 Detailed attack on TWINE-80 with $r = 8$

In this section, we discuss the detailed attack on TWINE-80 with 8-masked rounds. First let us discuss some key points.

- We choose initial plaintext difference Δ_{in} =E 4 0 0 1 0 0 4 E 4 B D 0 C E 4_x. This differential characteristic will give an output difference of Δ_{out} = 0 0 0 0 1 0 E 0 0 0 0 0 0 C 0 0_x with the probability of 2^{-28} after first 8-masked rounds (as shown in Fig. 2.7).[2]
- From Fig. 2.7, we can see that with this trail collisions happen at $\#9^1$, $\#9^2$, $\#9^3$, $\#9^4$, $\#9^6$, $\#9^8$, $\#9^9$, $\#9^{10}$, $\#9^{11}$, $\#9^{12}$, $\#9^{13}$, $\#9^{14}$, $\#9^{15}$, $\#10^1$, $\#10^2$, $\#10^3$, $\#10^5$, $\#10^6$, $\#10^7$, $\#10^{10}$, $\#10^{12}$, $\#10^{14}$, $\#10^{15}$, $\#10^{16}$, $\#11^1$, $\#11^2$, $\#11^4$, $\#11^6$, $\#11^8$, $\#11^{12}$, $\#11^{13}$, $\#11^{15}$. We call such collisions - *0-collisions.*
- To detect these collisions, we again use the hamming weight model. With the same assumption made in § 2.3, we check hamming weight collision at $\#9^1$, $\#9^3$, $\#9^9$, $\#9^{11}$, $\#9^{13}$, $\#9^{15}$, $\#10^1$, $\#10^3$, $\#10^5$, $\#10^7$, $\#10^{15}$, $\#11^1$, $\#11^{13}$, $\#11^{15}$ (shown in squared boxes in Fig. 2.7) i.e., total 14 positions.
- The probability that for any random plaintext pair which satisfies Δ_{in}, *HW-collision* occurs, can be calculated as discussed in § .2. Since we check *HW-collision* at all the above 14 positions in our attack, so the total probability of getting *HW-collisions* at all these 14 positions is $2^{-45.78}$.

[2] The reader can verify the computation of probability by the difference distribution table of TWINE sbox given in the § .1.

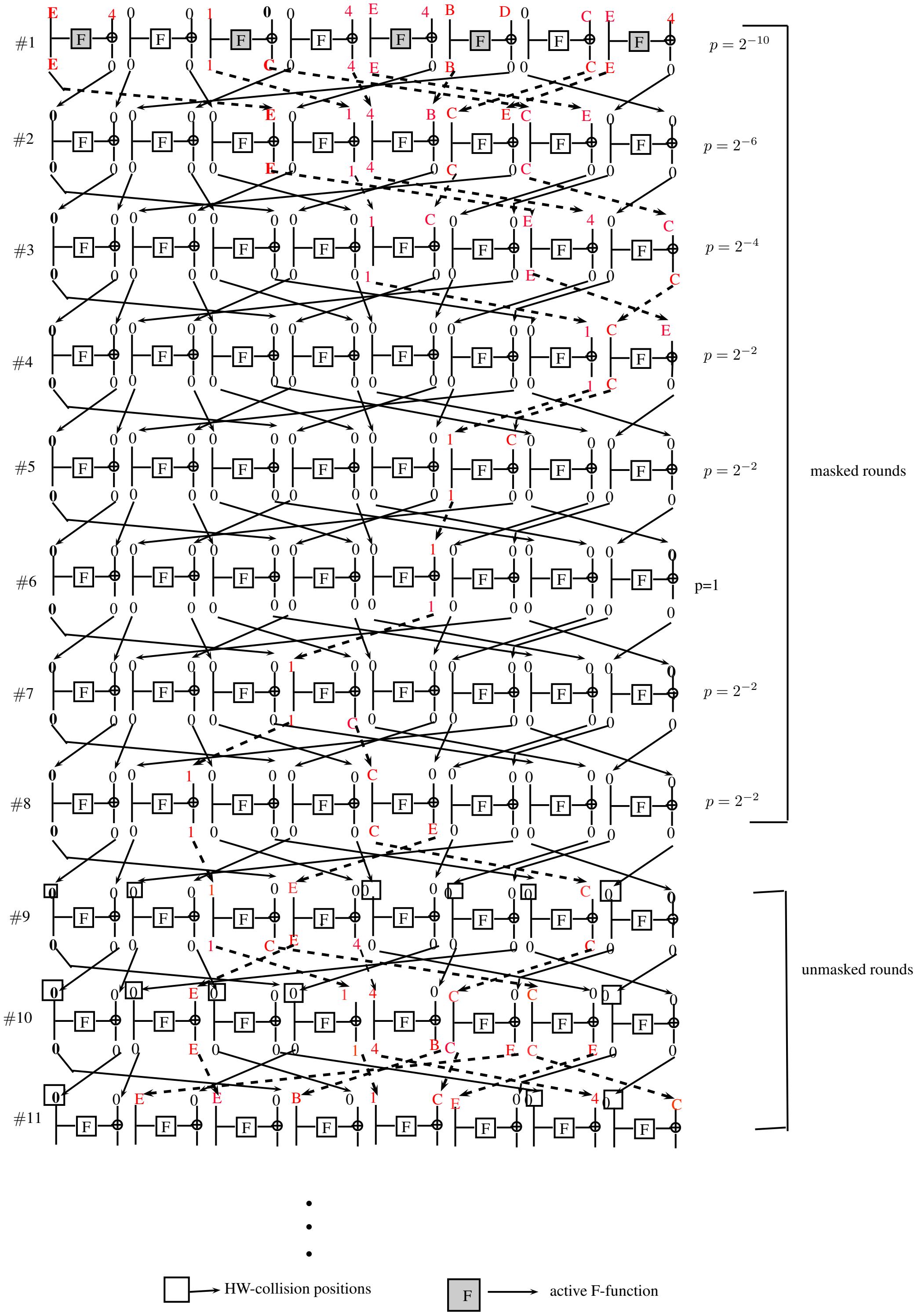


Figure 2.7: First 8-round differential characteristic

- Since, we are able to detect only *HW-collisions*, cases may arise where they do not correspond to actual *0-collisions*(as shown in Fig. 2.4). However, we later show that in our attack, depending upon our differential characteristic so chosen, we can safely assume that *HW-collision* so detected corresponds to actual *0-collisions* with very high probability.
- As shown in Fig. 2.7, the active F-functions in the first round are F^1,F^3,F^5,F^6 and F^8(i.e., F-functions with $\Delta^f_{in} \neq 0$).
- By applying the same procedure as discussed in § 2.3, we found that for all active F-function in the first round, 4 key candidates will be suggested.

  Now, using this information we have, the attack steps are as follows:

1. We choose 2^{30} plaintext pairs having Δ_{in} =E 4 0 0 1 0 0 4 E 4 B D 0 C E 4_x, and encrypt them with TWINE-80 with first eight masked rounds.
2. For each pair, we collect its power traces during encryption process and check for HW collision at $\#9^1, \#9^3, \#9^9$, $\#9^{11}, \#9^{13}, \#9^{15}$, $\#10^1, \#10^3, \#10^5, \#10^7$ $, \#10^{15}, \#11^1, \#11^{13}, \#11^{15}$ positions.
   - Since this differential characteristic holds with probability 2^{-28}, 4 right pairs($2^{30} \times 2^{-28}$) exist. This means that 4 right pairs yield *0-collisions* and give *HW-collision* with probability 1.
   - Since, for any random pair satisfying Δ_{in}, collisions are detected with a probability of $2^{-45.78}$. Thus, out of 2^{30} pairs, the number of pairs generating *HW-collision* are:

   $$\begin{aligned} &= \text{(number of right pair)} \times \text{(probability of } \textit{HW-collision}) + \\ &+ \text{(number of wrong pairs)} \times \text{(probability of } \textit{HW-collision}) \\ &= (4 \times 1) + (2^{30} - 4) \times 2^{-45.78} \approx (4 \times 1) + 2^{-15.78} \\ &= 4 \end{aligned}$$

     Thus, we can safely assume that the pairs detected in our attack as right pairs through *HW-collision* are pairs which correspond to actual *0-collisions.*
3. Once we get 4 right pairs, we move to the offline stage.
4. In the offline stage, as discussed earlier, for each right pair, we analyzed each of the 5 active F-functions in #1. After examining each active F-function, 4 equivalent keys were obtained for each of the five active F-function in #1.

2.4.1 Recovering the actual key from the set of equivalent keys.

To recover the actual key for each of the active F-function, we have to choose another differential characteristic over 8 rounds as discussed in § 2.3.1. We choose second differential characteristic with Δ_{in} =B D 0 0 C A 0 D B D 1 C 0 E 4 B_x. This characteristic will give Δ_{out} =0 0 0 0 C 0 4 0 0 0 0 0 0 E 0 0_x output difference after 8-masked rounds with a probability of 2^{-29} as shown in Fig. 2.8. We collect such 2^{31} plaintext pairs satisfying Δ_{in}. We then repeat the attack steps as discussed in § 2.4. We found another set of 4 equivalent keys for each active F-function in #1. Finally, we have got two sets of 4 equivalent keys for each active F-function and now our aim is to find correct subkey among 8 keys for each active F-function. We find the correct subkey for all the active F-functions i.e., F^1, F^3, F^5, F^6 and F^8 using the Lemma 1. Thus 20-bit key information is recovered.

Recovering key bits in the second round . Once subkey bits in the first round are obtained, we can use that key information to extract some key bits in the second round. From the subkey information obtained for F^3, we know the exact intermediate values for $\#2^{13}$ and from $\#1^9$ we know $\#2^{14}$. This is true for both the differential characteristics. Therefore, by applying the same procedure as discussed above, we can also find the subkey corresponding to F^7 in the second round. Finally, in our attack, we have recovered total 20(from first round)+4(from second round)=24-bits key information.

Attack Complexity. In the attack using the first and second differential trails, we need 2^{31} and 2^{32} chosen plaintexts respectively. Therefore the total data complexity is - 2^{31}+ 2^{32}= $2^{32.58}$ chosen plaintexts. In the online stage, we trace 2^{31}+ 2^{32}= $2^{32.58}$ power curves (measurements) and do 2^{30}+ 2^{31}= $2^{31.58}$ curve comparisons. Thus, the total time complexity is approximately $2^{32.58}$.

As mentioned in § 2.2, since TWINE-80 encryption and decryption functions are similar [25], the same attack procedure can be launched in chosen ciphertext model when the last 8 rounds are masked. This shows that an attacker can attack both top 8 and bottom 8 rounds of TWINE-80 and hence atleast 18 rounds (top 9 + last 9) of TWINE-80 should be masked.

2.5 Detailed attack on TWINE-80 with $r = 9$

In this section, we show the attack on TWINE-80 with 9-masked rounds. First let us discuss some key points.

- We choose initial plaintext difference Δ_{in} =0 E 4 A 4 B 0 E C E 4 B 4 0 D F_x as shown in Fig. 2.9. We get an output difference of Δ_{out} =0 0 0 0 1 0 E 0 0 0 0 0 0 C 0 0_x with the probability of 2^{-40}.[3]
- From Fig. 2.9, we can see that with this differential trail actual *0-collisions* happen at $\#10^1$, $\#10^2$, $\#10^3$, $\#10^4$, $\#10^6$, $\#10^8$, $\#10^9$, $\#10^{11}$, $\#10^{12}$, $\#10^{13}$, $\#10^{15}$, $\#10^{16}$, $\#11^1$,

[3] The reader can verify the computation of probability by the difference distribution table of TWINE sbox given in the § .1.

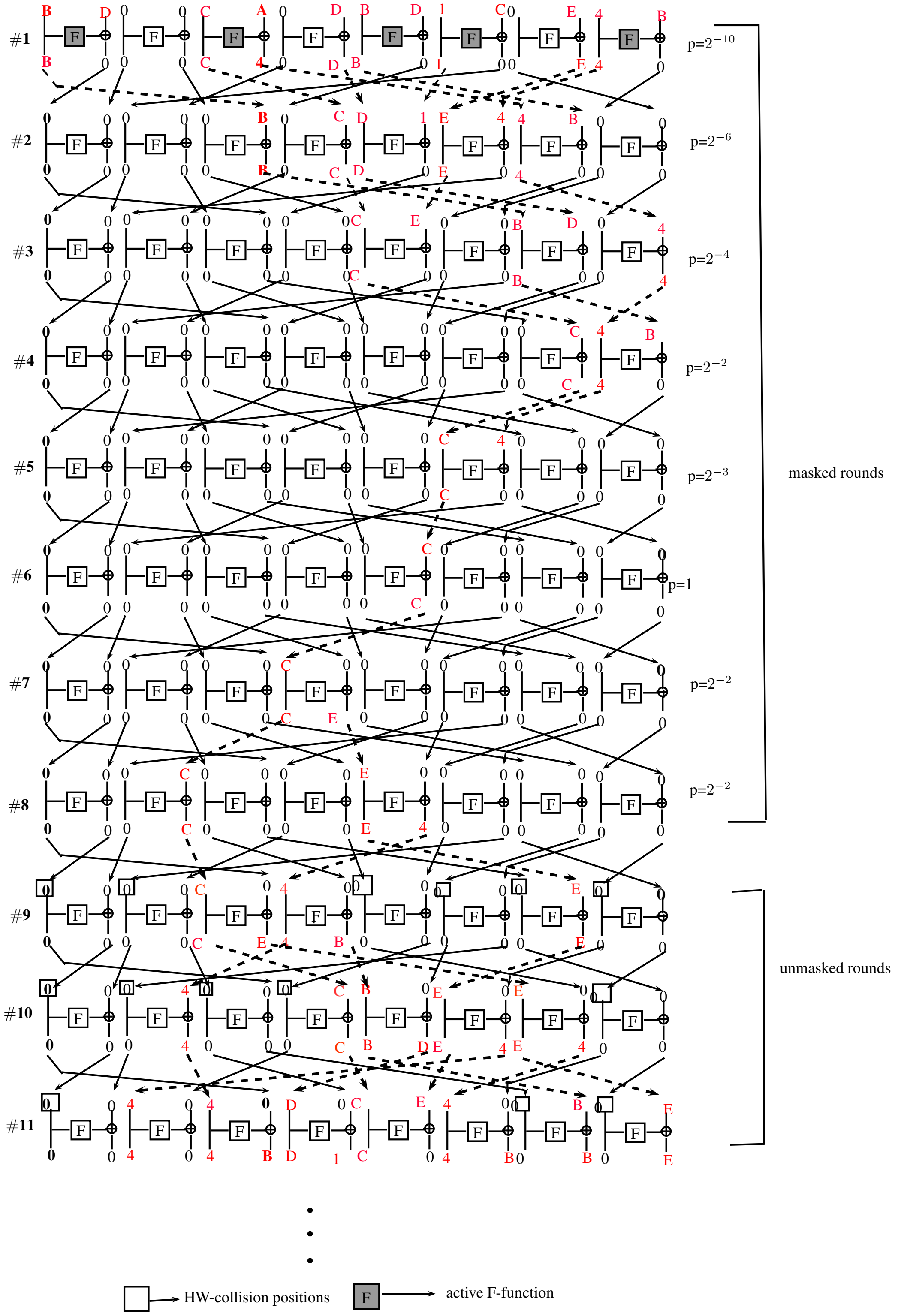


Figure 2.8: Second 8-round differential characteristic

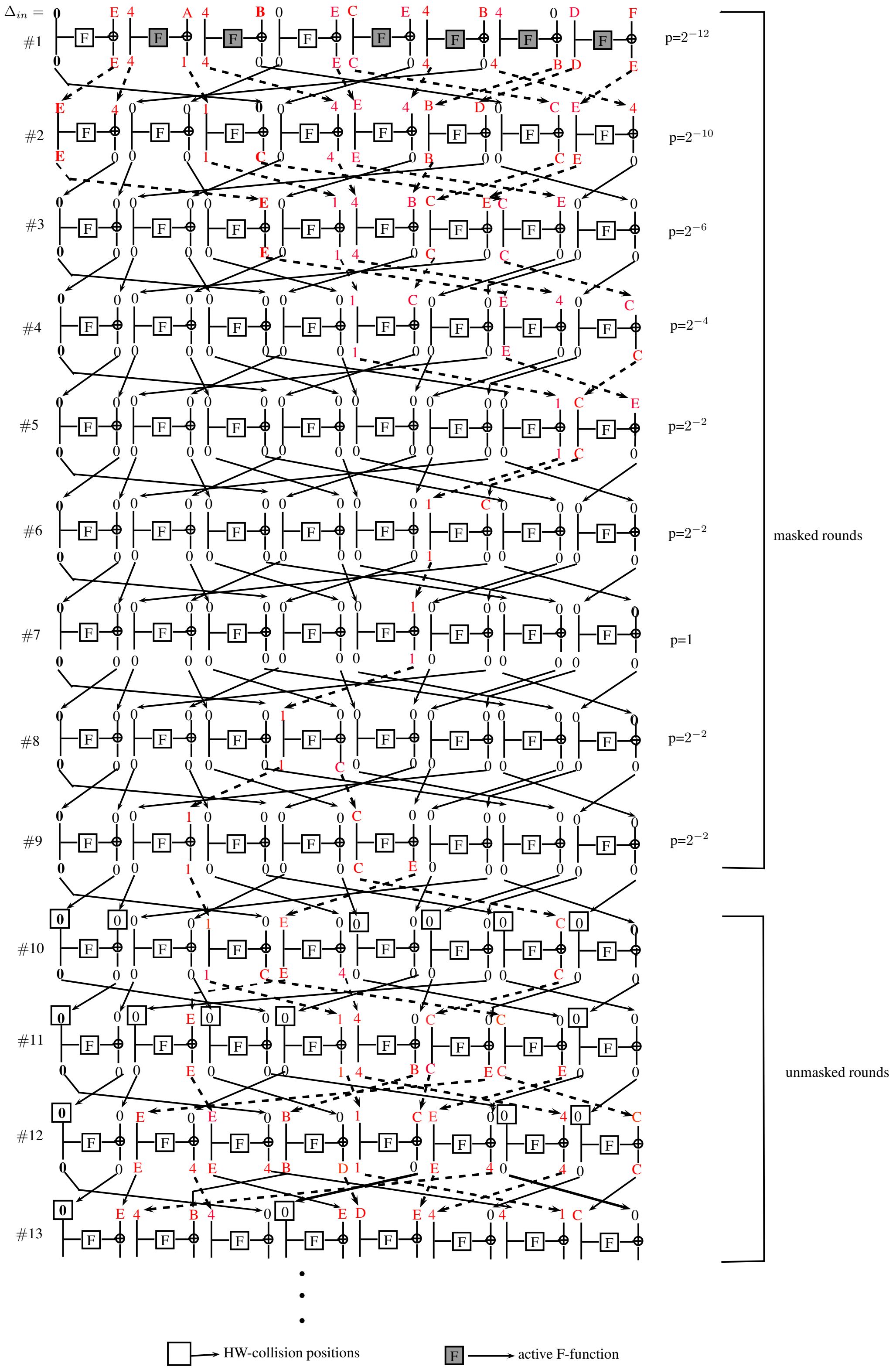


Figure 2.9: First 9-round differential characteristic for TWINE-80

$\#11^2$, $\#11^3$, $\#11^5$, $\#11^6$, $\#11^7$, $\#11^{10}$, $\#11^{12}$, $\#11^{14}$, $\#11^{15}$, $\#11^{16}$, $\#12^1$, $\#12^2$, $\#12^4$, $\#12^6$, $\#12^8$, $\#12^{12}$, $\#12^{13}$, $\#12^{15}$, $\#13^1$, $\#13^6$, $\#13^7$, $\#13^{12}$, $\#13^{16}$.

- To detect these collisions, we again use the hamming weight model as discussed in § 2.4. Based on the same assumption made in § 2.4, we check *HW-collision* at $\#10^1$, $\#10^3$, $\#10^9$, $\#10^{11}$, $\#10^{13}$, $\#10^{15}$, $\#11^1$, $\#11^3$, $\#11^5$, $\#11^7$, $\#11^{15}$, $\#12^1$, $\#12^{13}$, $\#12^{15}$, $\#13^1$, $\#13^7$. Then, the probability of *HW-collision* at all these 16 positions is $2^{52.32}$ which is calculated in a similar way as discussed § .2.

- From Fig. 2.9, we see that the active F-functions in the first round are F^2,F^3,F^5,F^6,F^7 and F^8(i.e., F-functions with $\Delta_{in}^{f} \neq 0$).

- By applying the same procedure as discussed in 8-round attack, we found that for all active F-function in the first round, 4 key candidates will be suggested. Now, using this information we have, the attack steps are as follows :

1. We choose 2^{42} plaintext pairs having Δ_{in} =0 E 4 A 4 B 0 E C E 4 B 4 0 D F_x, and encrypt them with TWINE-80 with first nine masked rounds.

2. For each pair, we collect its power traces during encryption process and check for HW collision at $\#10^1$, $\#10^3$, $\#10^9$, $\#10^{11}$, $\#10^{13}$, $\#10^{15}$, $\#11^1$, $\#11^3$, $\#11^5$, $\#11^7$, $\#11^{15}$, $\#12^1$, $\#12^{13}$, $\#12^{15}$, $\#13^1$, $\#13^7$ positions (shown in Fig. 2.9 in squared boxes).

   - Since this differential characteristic holds with probability 2^{-40}, 4 right pairs($2^{42} \times 2^{-40}$) exist. This means that 4 right pairs yield *0-collisions* and give *HW-collision* with probability 1.
   - Since, for any random pair, collision are detected with a probability of $2^{-52.32}$.Thus, out of 2^{42} pairs, the number of pairs generating *HW-collision* are:

$$\begin{aligned}
&= (\text{number of right pair}) \times (\text{probability of } \textit{HW-collision}) + \\
&+ (\text{number of wrong pairs}) \times (\text{probability of } \textit{HW-collision}) \\
&= (4 \times 1) + (2^{42} - 4) \times 2^{-52.32} \\
&= 4
\end{aligned}$$

   Thus, the pairs actually detected in our attack through *HW-collision* are pairs which correspond to *0-collisions* as well.

3. Once we got 4 right pairs, we move to the offline stage.

4. In the offline stage as discussed in § 2.3, for each right pair, we analyzed all active F-functions and we obtained 4 equivalent keys for each active F-function in the first round and we cannot determine which amongst them is actual key.

2.5.1 Recovering the actual key from the set of equivalent keys.

The procedure of recovering the key is same as discussed earlier. The another differential characteristic that we choose is shown in Fig. 2.10 with Δ_{in} =A C D D D 1 0 B E 4 D 1 D C C A_x and Δ_{out} =0 0 0 0 C 0 4 0 0 0 0 0 0 E 0 0_x and it holds with probability 2^{-43}. We collect 2^{45} plaintext pairs satisfying Δ_{in}. We then repeat the whole attack procedure discussed above and find the another set of equivalent keys for each active F-function in the first round. We can see that F^1 is active in this trail but it was not active in the first trail, so by using the Lemma 1, we can obtain the correct key for F^2,F^3,F^5,F^6,F^7 and F^8 active F-functions except F^1, but we can extract 2-bits of information from 4 equivalent keys obtained from the attack through the second trail. Thus, we have recovered total 24+2 = 26-bits of information.

Recovering key bits in the second round . Once subkey bits at first round are obtained, we can use the key information to extract key bits in the second round. From the subkey information obtained in first round, we know the exact intermediate values for $\#2^5$, $\#2^6$, $\#2^{11}$, $\#2^{12}$,$\#2^{15}$ and $\#2^{16}$. This is true for both the differential characteristics. Therefore, by using the Lemma 1, we can also find the subkey corresponding to F^3, F^6 and F^8 in the second round. We also found that we know the exact values of $\#2^1$ and $\#2^2$ in second differential trail but we do not know these values in first trail, so we can find 4 equivalent keys for F^1 in #2 from attack using second trail from which we can get 2-bits of information. Finally, in our attack, we have recovered total 26(from first round)+14(from second round)=40-bits key information.

Attack Complexity. In the attack using the first and second differential trails, we need 2^{43} and 2^{46} chosen plaintexts respectively. Therefore the total data complexity is - 2^{43}+ 2^{46}= $2^{46.17}$ chosen plaintexts. In the online stage, we trace 2^{43}+ 2^{46}= $2^{46.17}$ power curves (measurements) and do 2^{42}+ 2^{45}= $2^{45.17}$ curve comparisons. Thus, the total time complexity is approximately $2^{46.17}$.

As mentioned in § 2.2, since TWINE-80 encryption and decryption functions are similar [25], the same attack procedure can be launched in chosen ciphertext model when the last 9 rounds are masked (as shown in Table 2.4). This shows that an attacker can attack both top 9 and bottom 9 rounds of TWINE-80 and hence atleast 20 rounds (top 10 + last 10) of TWINE-80 should be masked.

Table 2.4: Details of our 9-round chosen ciphertext attack. Here, in columns Δ_{in} and Δ_{out}, the first row denotes the left half and second row denotes the right half of ciphertext/ intermediate state difference respectively.

Keybits Recovered	Rounds	Trail	Δ_{in}	Δ_{out}	Trail Probability	HW-collision Probability	Data/Time Complexity
36	9	First	0 E 4 0 4 B 0 E D F 4 B 4 A C E	0 0 0 0 0 0 0 0 1 0 0 0 E 0 0 C	2^{-40}	$2^{-52.32}$	$2^{44.58}$
		Second	0 4 B 0 B D 0 4 1 8 B D B 1 E 4	0 0 0 0 0 0 0 0 C 0 0 0 B 0 0 4	2^{-41}	$2^{-52.32}$	

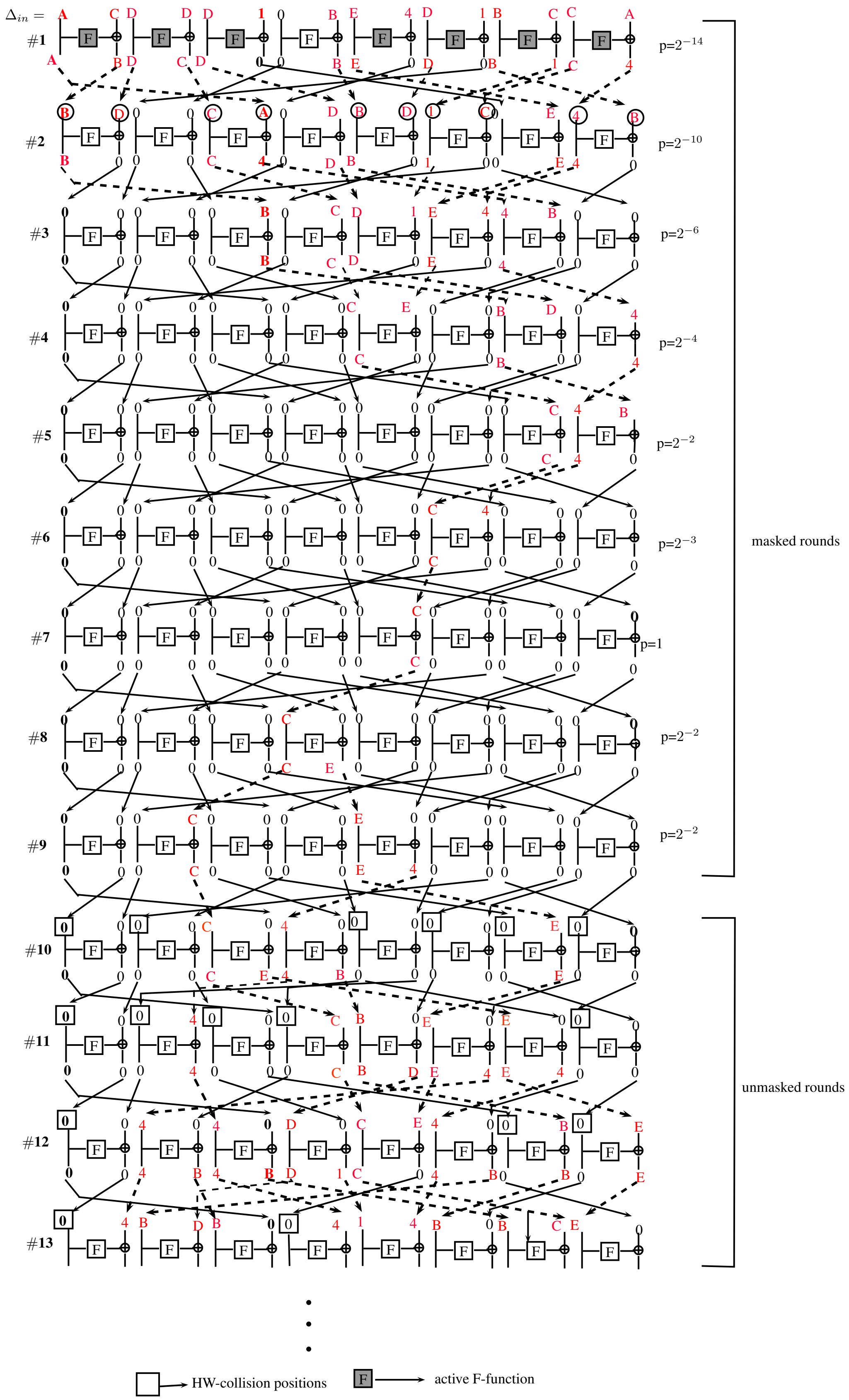


Figure 2.10: Second 9-round differential characteristic for TWINE 80

Table 2.5: Details of our other attacks. Here, in columns Δ_{in} and Δ_{out}, the first row denotes the left half and second row denotes the right half of plaintext/ intermediate state difference respectively.

S.No.	Rounds	Trail	Δ_{in}	Δ_{out}	Trail	HW-collision	Data
					Probability	Probability	Complexity
2	8	First	0 0 0 0 0 E 0 1 4 B C E C E 0 0	0 0 0 E 0 0 0 1 4 0 C 0 C 0 0 0	2^{-22}	$2^{-32.7}$	$2^{26.58}$
		Second	0 0 0 0 0 B 0 C D 1 E 4 4 B 0 0	0 0 0 4 0 0 0 C B 0 E 0 E 0 0 0	2^{-23}	$2^{-32.7}$	
4	9	First	0 0 0 0 0 E 0 1 4 B C E C E 0 0	0 0 E 0 E 0 B 0 1 C E 0 0 4 0 C	2^{-28}	$2^{-39.4}$	$2^{32.58}$
		Second	0 0 0 0 0 B 0 C D 1 E 4 4 B 0 0	0 0 4 0 4 0 D 0 C E 4 0 0 B 0 E	2^{-29}	$2^{-39.4}$	

2.6 Attacks with lower data and time complexities

Our above discussed 8 and 9-round attack requires large number of online measurements. Hence, in Table 2.5, we report another set of two 8 and 9-round differential characteristics with which our attack has lower time and data complexity. However, the number of key bits recovered are comparatively lower, i.e., only 12 key bits (as reported in Table 2.1) could be recovered. In Table 2.5, we also report the important details of our 8 and 9-round masked attack with lower data complexity. All the differential characteristics mentioned in Table 2.5 are new and hitherto not been reported before. Since, in our attacks, the time and data complexities are exactly the same, hence we report only the data complexity of these attacks. All these attacks can be launched similar to above attack discussed in this chapter.

Chapter 3

Side channel collision attack on DES

The Data Encryption standard(DES) designed by a team at IBM, is a 16-round Feistel network with 64-bit blocks and 56-bit keys. Each round key is of 48-bit computed from the master key through key schedule. In this chapter, we improved the Jongsung et al.'s work [14]. In that work, in case of 7-masked round attack they were able to get 15-bit key information only in the first round, but in our work we improved it using one more charachteristic mentioned in [15] and now we are recovering full 48-bit key information in the first round. The data complexity of our attack is $2^{36.99}$. The time complexity is $2^{36.99}$ measurements and $2^{35.99}$curve comparisons.

3.1 Basic layout of the attack

The basic layout of the attack is same as discussed in § 2.2 except the following in the *offline stage*,

1. In place of active F-functions, we analyze active sboxes here in the similar way as explained for active F-functions in§ 2.2.

2. Once we got key candidates for each active sbox, then we have to find the correct key among the suggested key candidates for each active sbox.

3. In [14], a concept of *hit* is given to distinguish the correct key and wrong key.

   As we find the key candidates by analysing each active sbox for each right pair found in the *online phase*, so the correct key will be suggested for each right pair corresponding to each active sbox and therefore we say that correct key will give 2^{t-p} hits(equal to the number of right pairs§ 2.2). The key which gives less than 2^{t-p} hits will be wrong key.

If all sboxes are not active in the first round. We choose another differential characteristic having active sboxes which are not active in the first diffrential trail so that we can find subkey information corresponding to these sboxes as well and finally we can find the full round subkey by finding the rest of the key bits through exhaustive search.

3.2 Detailed attack on DES with 7 masked rounds

We take two 2-round iterative characteristics, first is mentioned in [2] with probability $\approx 2^{-7.87}$ and other in [15] with probability $\approx 2^{-8.09}$. We construct 7-round differential characteristic from each 2-round iterative charcteristic with probability $2^{-31.5}$ and $2^{-32.37}$ respectively using each characteristic three and half times. The first characteristic having input output difference both 00 00 00 00 19 60 00 00 is shown in Fig. 3.1 and second characteristic having input output difference both 00 00 00 00 00 00 03 d4 is shown in Fig. 3.2.

In case of both differential characteristics, for each right pair we get 32-bit collision after F function in 8th round shown in Fig. 3.3 which means wrong pairs can be fitered out [14] with probability 2^{-32}. To increase the wrong pair filtering rate, we can filter out in 9th round also after F function. Since the difference for the first and second differential characteristic is 19 60 00 00 and 00 00 03 d4 respectively in 9th round, so we can get collision at 22 bits after E expansion box with probability 2^{-22}, therefore we can filter out all wrong pairs at filtering rate 2^{-54}.

The attack procedure using above given information is as follows:

1. Choose $2^{35.37}$ plaintext pairs satisfying input difference 00 00 00 00 00 00 03 d4 and encrypt them with DES first 7 and last 7 masked rounds.

2. For each plaintext pair, collect it's power traces and check for collision at input positions of $s_1, s_2, s_3, s_4, s_5, s_6, s_7$ and s_8 s-boxes in 8th round and if collision doesn't occur discard that pair.

3. For remaining pairs check for collision at input positions of s_1, s_2, s_3, s_4 and s_5 s-boxes in 9th round and if collision doesn't occur discard that pair.

4. Now using those pairs which satisfy above test, analyse s_6, s_7 and s_8 s-boxes in the first round using difference distribution table which suggest key candidates because s_6, s_7 and s_8 are active in the first round, active means s-box having nonzero input difference.

5. output key which give maximum hits.

Right key will give 8 hits(explained in § 3.1) always as 8 plaintext pairs expected to be right pairs, since the differential characteristic holds with probability $2^{32.37}$ and we choose $2^{35.37}$ plaintext pairs . since the filtering rate is 2^{-54}, so there will be no wrong pair out of $2^{35.37}$ chosen plaintexts. Now, we have to calculate that wrong key will give how many hits.
In case of each active s-boxes s_6, s_7 and s_8 respectively, there will be 2, 2 and 2 equivalent keys. Suppose, for s_6 we have a set S of 6 plaintext pairs in Fig. 3.4, then equivalent means $\{a{\cdot}b | b \in S\}$, thus $S = 07_x \cdot S$. So $k, k \oplus 07_x$ are equivalent where k is any key candidate suggested through difference distribution table of s_6. For, any right pair following the differential characteristic, 56 keys will never be suggested, so 56 wrong keys will give 0 hits and since we are remaining with 8 keys out of which one is correct and 2 will be equivalent to it, so 6 will be wrong keys.

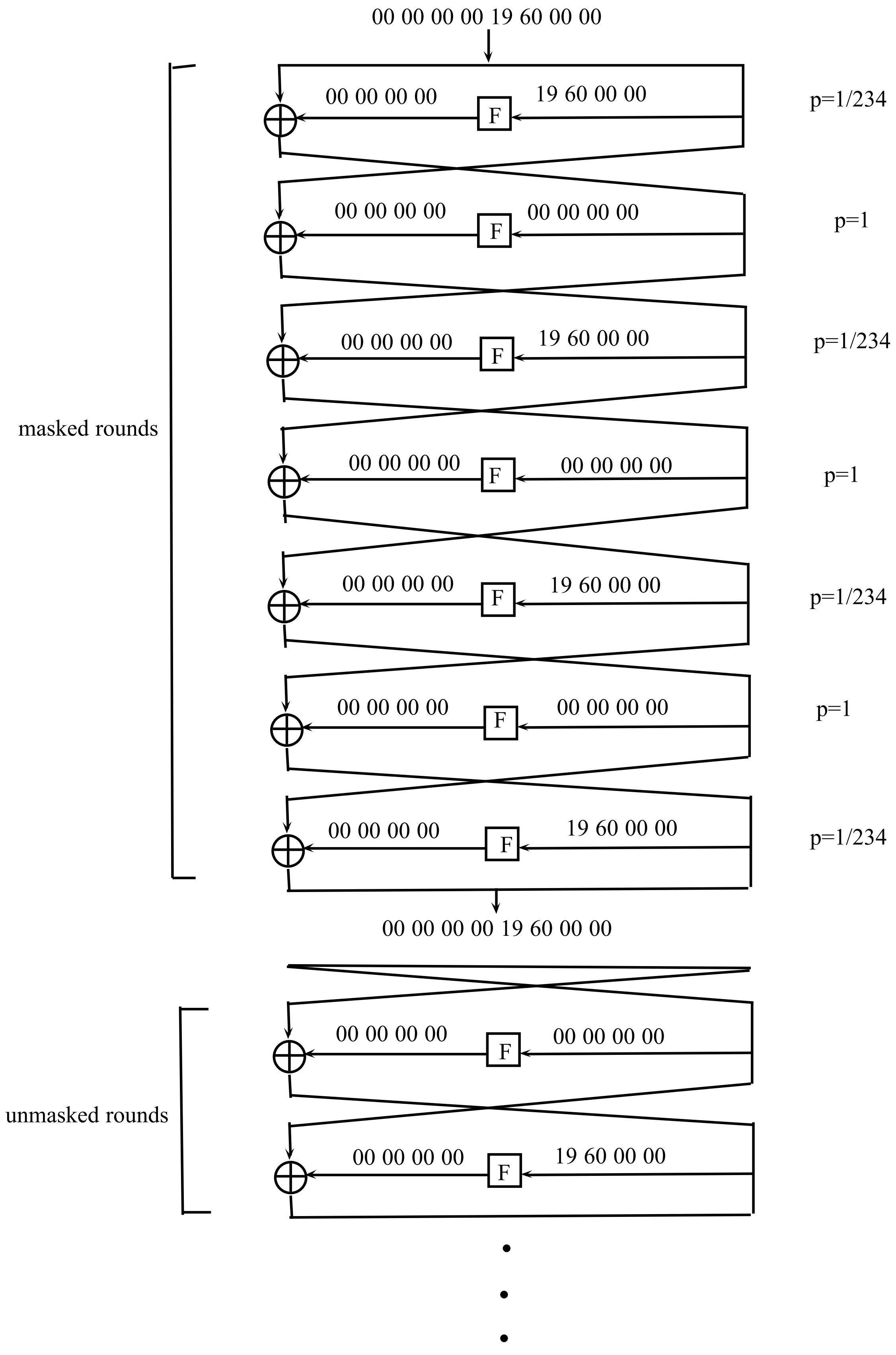


Figure 3.1: first differential characteristic

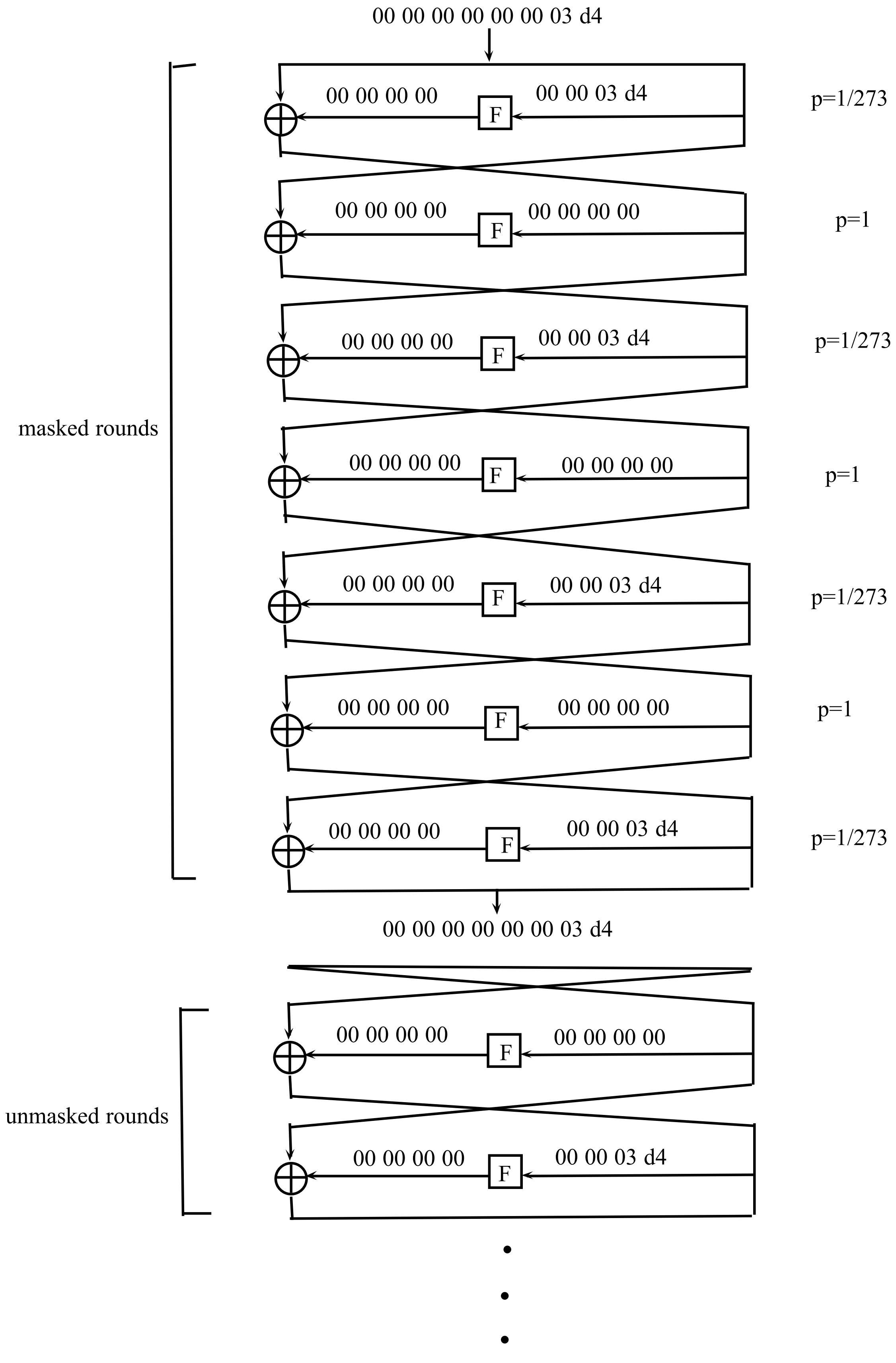


Figure 3.2: second differential characteristic

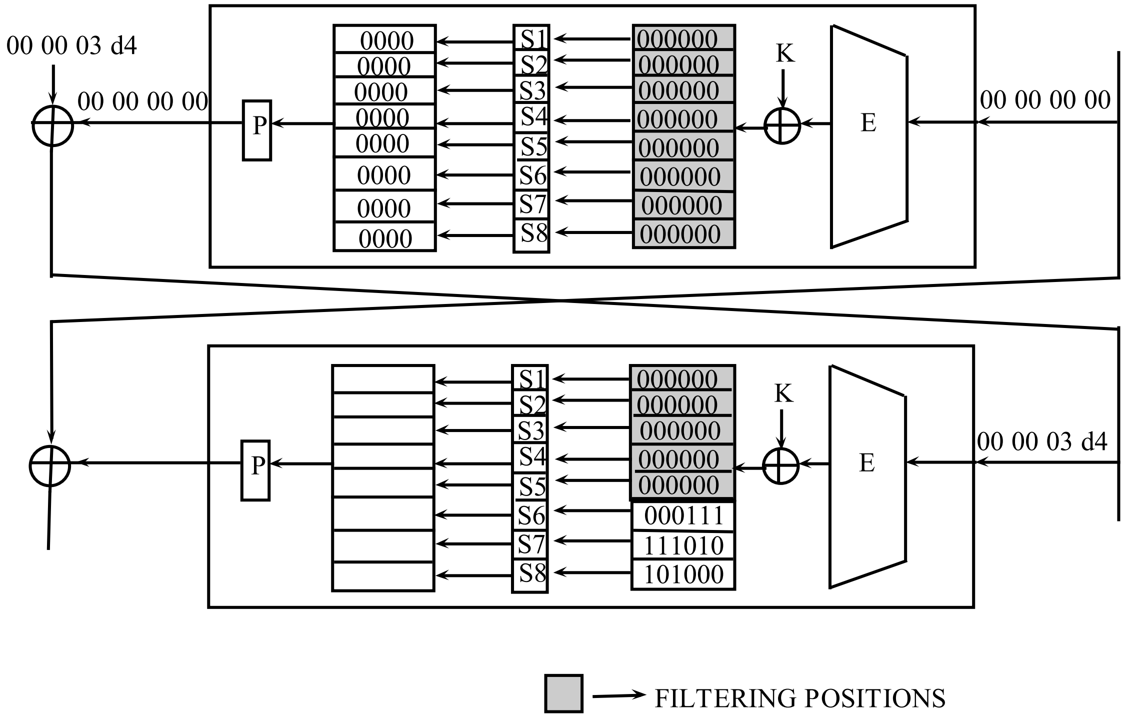


Figure 3.3: Filtering in 8^{th} and 9^{th} round for second characteristic

(010010 010101) (010101 010010) (110000 110111) (110111 110000) (111001 111110) (111110 111001)

plaintext pairs satisfying 000111 ⟶ 0000 at S6

(000100 111110) (001000 110010) (001011 110001) (001100 110110) (010000 101010) (010010 101000) (010110 101100) (011101 100111)
(100111 011101) (101000 010010) (101010 010000) (101100 010110) (110001 001011) (110010 001000) (110110 001100) (111110 000100)

plaintext pairs satisfying 111010 ⟶ 0000 at S7

(001110 100110) (010011 111011) (010101 111101) (010111 111111) (011101 110101) (100110 001110)
(110101 011101) (111011 010011) (111101 010101) (111111 010111)

plaintext pairs satisfying 101000 ⟶ 0000 at S8

Figure 3.4: possible input pairs for s_6, s_7 and s_8

Therefore wrong key hits for these 6 wrong keys = $(6t-2t)/6$ hits, where t is number of right pairs i.e 8. So, wrong key will give 5.33 hits.
In case of s_7, 46 keys will be suggested only and out of which 44 will be wrong keys as two are equivalent keys i.e $k, k \oplus 58_x$ and 16 plaintext pairs satisfies the input output difference of s_7 shown in Fig. 3.4, therefore wrong key will give = $(16t-2t)/44$ i.e 2.545 hits.
In case of s_8, 22 keys will be suggested only and out of which 20 will be wrong keys as two are equivalent keys i.e $k, k \oplus 40_x$ and 10 plaintext pairs satisfies the input output difference of s_8 shown in Fig. 3.4, therefore wrong key will give = $(10t-2t)/20$ i.e 3.2 hits.

So, finally through this differential characteristic, we are able to recover 15(5+5+5) bit key information using s_6, s_7 and s_8.
Therfore the data complexity of attack through this characteristic is $2^{36.37}$.

Similarly, we apply the same attack procedure on first differential characteristic shown in Fig. 3.1 with input difference in first round $00\ 00\ 00\ 00\ 19\ 60\ 00\ 00_x$. In first differential characteristic [14] the probability of following the characteristic is $2^{-31.5}$, so we choose $2^{34.5}$ chosen plaintexts and therefore 8 plaintext pairs are expected to be right pairs. So right key is expected to be suggested 8 times.
The filtering rate is same as for above characteristic i.e 2^{-54}, so no wrong pair will occur in this case also.
In this differential characteristic [14], s_1, s_2 and s_3 are active means have nonzero difference. Similarly to the above analysis 14, 8 and 10 key candidates will be suggested through the distribution table of s_1, s_2 and s_3 respectively and each have 2, 2 and 2 equivalent keys. In case of s_1, wrong key will give = $(14t-2t)/42$ hits, as 44 keys will be suggested in case of s_1 and 2 are equivalent keys, so 42 will be wrong keys and t is number of right pairs, so wrong key hits=2.29 hits . Similarly in case of s_2 and s_3, wrong will give 4 and 0.52 hits respectively.
Through this differential characteristic also, we are able to recover 15(5+5+5) bit key information using s_1, s_2 and s_3.
The data complexity of the attack through this characteristic is $2^{35.5}$.

So, we can get two equivalent keys through both differential characteristic corresponding to each active s-boxes i.e s_1, s_2, s_3, s_6, s_7 and s_8 and now, we have to find correct subkey among these 2 equivalent keys and we can find correct subkey corresponding to s_4 and s_5 by exhaustive search. Therefore, in our attack we can find full round key i.e 48-bit information in the first round. The data complexity in this phase is = $(2 \times 2 \times 2 \times 2^6 \times 2^6 \times 2 \times 2 \times 2) = 2^{18}$.
Therefore the total data complexity of our attack is $2^{36.37} + 2^{35.5} + 2^{18} = 2^{36.99}$. The time complexity is $2^{36.99}$ measurements and $2^{35.99}$curve comparisons.

Chapter 4

Conclusion and Future work

In our work, we present the first differential based side channel collision attack on TWINE-80 with the first and last 7, 8 and 9-rounds masked with the data complexity of $2^{22.58}$, $2^{32.58}$ and $2^{46.17}$ respectively. Similar attack would also work for TWINE-128. This is the first side channel attack demonstrated on TWINE-80 and shows that more than 20 rounds needs to be masked to guarantee security against side channel attack.

In our work, we also improved the Jongsung et al.'s work [14]. In that work, in 7-masked round attack they were able to get 15-bit key information only in the first round, but in our work we improved it using one more charachteristic mentioned in [15] and we recover full 48-bit subkey. The data complexity of our attack is $2^{36.99}$. The time complexity is $2^{36.99}$ measurements and $2^{35.99}$curve comparisons.

In future work, one may try to perform the above described attacks on some other block ciphers or lightweight block ciphers.

Bibliography

[1] Akkar, M., and Goubin, L. A generic protection against high-order differential power analysis. In *Fast Software Encryption, 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003, Revised Papers* (2003), T. Johansson, Ed., vol. 2887 of *Lecture Notes in Computer Science*, Springer, pp. 192–205.

[2] Biham, E., and Shamir, A. Differential cryptanalysis of des-like cryptosystems. In *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings* (1990), A. Menezes and S. A. Vanstone, Eds., vol. 537 of *Lecture Notes in Computer Science*, Springer, pp. 2–21.

[3] Biryukov, A., Bogdanov, A., Khovratovich, D., and Kasper, T. Collision attacks on aes-based MAC: alpha-mac. In *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings* (2007), P. Paillier and I. Verbauwhede, Eds., vol. 4727 of *Lecture Notes in Computer Science*, Springer, pp. 166–180.

[4] Biryukov, A., Derbez, P., and Perrin, L. Differential analysis and meet-in-the-middle attack against round-reduced TWINE. *IACR Cryptology ePrint Archive 2015* (2015), 240.

[5] Biryukov, A., and Khovratovich, D. Two new techniques of side-channel cryptanalysis. In *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings* (2007), P. Paillier and I. Verbauwhede, Eds., vol. 4727 of *Lecture Notes in Computer Science*, Springer, pp. 195–208.

[6] Bogdanov, A. Improved side-channel collision attacks on AES. In *Selected Areas in Cryptography, 14th International Workshop, SAC 2007, Ottawa, Canada, August 16-17, 2007, Revised Selected Papers* (2007), C. M. Adams, A. Miri, and M. J. Wiener, Eds., vol. 4876 of *Lecture Notes in Computer Science*, Springer, pp. 84–95.

[7] Bogdanov, A. Multiple-differential side-channel collision attacks on AES. In *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings* (2008), E. Oswald and P. Rohatgi, Eds., vol. 5154 of *Lecture Notes in Computer Science*, Springer, pp. 30–44.

[8] Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., Seurin, Y., and Vikkelsoe, C. PRESENT: an ultra-lightweight block cipher. In *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings* (2007), P. Paillier and I. Verbauwhede, Eds., vol. 4727 of *Lecture Notes in Computer Science*, Springer, pp. 450–466.

[9] Bogdanov, A., and Pyshkin, A. Algebraic side-channel collision attacks on AES. *IACR Cryptology ePrint Archive 2007* (2007), 477.

[10] Cazorla, M., Marquet, K., and Minier, M. Survey and benchmark of lightweight block ciphers for wireless sensor networks. In *SECRYPT 2013 - Proceedings of the 10th International Conference on Security and Cryptography, Reykjavík, Iceland, 29-31 July, 2013* (2013), P. Samarati, Ed., SciTePress, pp. 543–548.

[11] Gierlichs, B., Batina, L., Preneel, B., and Verbauwhede, I. Revisiting higher-order DPA attacks:. In *Topics in Cryptology - CT-RSA 2010, The Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, March 1-5, 2010. Proceedings* (2010), J. Pieprzyk, Ed., vol. 5985 of *Lecture Notes in Computer Science*, Springer, pp. 221–234.

[12] Guo, J., Peyrin, T., Poschmann, A., and Robshaw, M. J. B. The LED block cipher. In *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings* (2011), B. Preneel and T. Takagi, Eds., vol. 6917 of *Lecture Notes in Computer Science*, Springer, pp. 326–341.

[13] Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B., Lee, C., Chang, D., Lee, J., Jeong, K., Kim, H., Kim, J., and Chee, S. HIGHT: A new block cipher suitable for low-resource device. In *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings* (2006), L. Goubin and M. Matsui, Eds., vol. 4249 of *Lecture Notes in Computer Science*, Springer, pp. 46–59.

[14] Kim, J., Lee, Y., and Lee, S. DES with any reduced masked rounds is not secure against side-channel attacks. *Computers & Mathematics with Applications 60*, 2 (2010), 347–354.

[15] Knudsen, L. R. Iterative characteristics of DES and s^2-des. In *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings* (1992), E. F. Brickell, Ed., vol. 740 of *Lecture Notes in Computer Science*, Springer, pp. 497–511.

[16] Kocher, P. C. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings* (1996), N. Koblitz, Ed., vol. 1109 of *Lecture Notes in Computer Science*, Springer, pp. 104–113.

[17] Kocher, P. C., Jaffe, J., and Jun, B. Differential power analysis. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings* (1999), M. J. Wiener, Ed., vol. 1666 of *Lecture Notes in Computer Science*, Springer, pp. 388–397.

[18] Kocher, P. C., Jaffe, J., Jun, B., and Rohatgi, P. Introduction to differential power analysis. *J. Cryptographic Engineering 1*, 1 (2011), 5–27.

[19] Ledig, H., Muller, F., and Valette, F. Enhancing collision attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings* (2004), M. Joye and J. Quisquater, Eds., vol. 3156 of *Lecture Notes in Computer Science*, Springer, pp. 176–190.

[20] Lee, Y., Kim, J., and Hong, S. Side-channel attacks on HIGHT with reduced masked rounds suitable for the protection of multimedia computing system. *Multimedia Tools Appl. 56*, 2 (2012), 267–280.

[21] Mangard, S. A simple power-analysis (SPA) attack on implementations of the AES key expansion. In *Information Security and Cryptology - ICISC 2002, 5th International Conference Seoul, Korea, November 28-29, 2002, Revised Papers* (2002), P. J. Lee and C. H. Lim, Eds., vol. 2587 of *Lecture Notes in Computer Science*, Springer, pp. 343–358.

[22] Nyberg, K. Generalized feistel networks. In *Advances in Cryptology - ASIACRYPT '96, International Conference on the Theory and Applications of Cryptology and Information Security, Kyongju, Korea, November 3-7, 1996, Proceedings* (1996), K. Kim and T. Matsumoto, Eds., vol. 1163 of *Lecture Notes in Computer Science*, Springer, pp. 91–104.

[23] Schramm, K., Leander, G., Felke, P., and Paar, C. A collision-attack on AES: combining side channel- and differential-attack. In *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings* (2004), M. Joye and J. Quisquater, Eds., vol. 3156 of *Lecture Notes in Computer Science*, Springer, pp. 163–175.

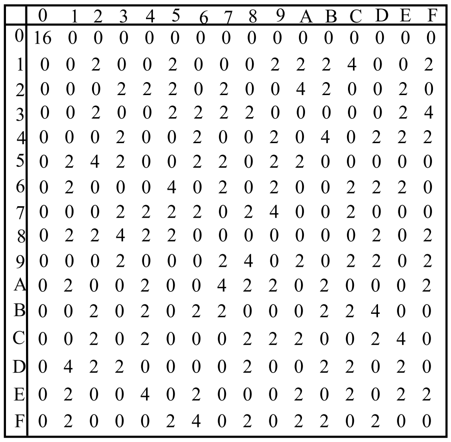
[24] Suzaki, T., and Minematsu, K. Improving the generalized feistel. In *Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers* (2010), S. Hong and T. Iwata, Eds., vol. 6147 of *Lecture Notes in Computer Science*, Springer, pp. 19–39.

[25] Suzaki, T., Minematsu, K., Morioka, S., and Kobayashi, E. TWINE: A lightweight block cipher for multiple platforms. In *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers* (2012), L. R. Knudsen and H. Wu, Eds., vol. 7707 of *Lecture Notes in Computer Science*, Springer, pp. 339–354.

APPENDIX

.1 Difference Distribution Table (DDT) of TWINE-80 SBox

In Fig. 1, rows denote the input differences and columns denote the output differences.



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

Figure 1: DDT of TWINE-80 Sbox.

.2 Calculation of Hamming Weight Probability

Here, our aim is to calculate the probability of any random pair (satisfying a given Δ_{in}) giving HW-collision at the input of any one chosen F-function F. We use the following notations to calculate the probability.

$outcoll$: collision occurs at output position of F-function.

$incoll$: collision occurs at input position of F-function.

k : key value, where $0 \leq k \leq 15$

$P[outcoll|incoll \wedge HW = i]$: probability of getting *HW-collision* at the output of the F-function given *HW-collision* occurs at input of F-function and HW of both inputs in the input pair $= i$, where $0 \leq i \leq 4$.

$P[incoll \wedge HW = i]$: probability of getting *HW-collision* at the input of the F-function when HW of both inputs in the input pair $= i$, where $0 \leq i \leq 4$.

The probability of getting *HW-collision* at both the input and output position of F^1 can be calculated as:

$$
\begin{aligned}
P[outcoll \wedge incoll)] = P[outcoll|incoll)] \times P[incoll] = \\
&\sum_{k=0}^{15}(P[outcoll|incoll \wedge HW = 0] \times P[incoll \wedge HW = 0]) \\
&+ \sum_{k=0}^{15}(P[outcoll|incoll \wedge HW = 1] \times P[incoll \wedge HW = 1]) \\
&+ \sum_{k=0}^{15}(P[outcoll|incoll \wedge HW = 2] \times P[incoll \wedge HW = 2]) \\
&+ \sum_{k=0}^{15}(P[outcoll|incoll \wedge HW = 3] \times P[incoll \wedge HW = 3]) \\
&+ \sum_{k=0}^{15} P[outcoll|incoll \wedge HW = 4] \times P[incoll \wedge HW = 4])
\end{aligned}
$$

$\sum_{k=0}^{15} P[outcoll|incoll \wedge HW = 0] \times P[incoll \wedge HW = 0] = \frac{16}{16} \times \frac{1}{256}$ (since, there is only one single input pair for which HW =0, i.e., (0000, 0000) and this pair gives *HW-collision* at the output of F-function for all 16 possible key values).

Programmatically we found that,

$\sum_{k=0}^{15} P[outcoll|incoll \wedge HW = 1] \times P[incoll \wedge HW = 1]{=}\frac{100}{16\times256}$

$\sum_{k=0}^{15} P[outcoll|incoll \wedge HW = 2] \times P[incoll \wedge HW = 2]{=}\frac{192}{16\times256}$

$\sum_{k=0}^{15} P[outcoll|incoll \wedge HW = 3] \times P[incoll \wedge HW = 3]{=}\frac{100}{16\times256}$

$\sum_{k=0}^{15} P[outcoll|incoll \wedge HW = 4] \times P[incoll \wedge HW = 4]{=}\frac{16}{16\times256}$

Thus,

$P[outcoll \wedge incoll)] = \frac{16}{16\times256} + \frac{100}{16\times256} + \frac{192}{16\times256} + \frac{100}{16\times256} + \frac{16}{16\times256} = \frac{424}{4096} \approx 2^{-3.27}$

Since, in § 2.3, *HW-collisions* happen at 16 positions, total *HW-collision* probability is $2^{-52.32}$.

.3 Proof of Lemma 1

We state the lemma again.

Lemma 1. Let A and B be two sets of subkey candidates obtained from differential characteristics 1 and 2 respectively. Let P be a set as defined below,

$$P = \{x | y \oplus z = x, \text{where}, y \in A \text{ and } z \in B\}$$

Here $|P| = 16$, i.e., P will have 16 elements. If all the 16 elements in P are unique, then only one single key will be common in both the sets - A and B.

We prove our lemma by contradiction. Let us consider the two sets of subkeys as follows:

Set A	Set B
k	k'
$k \oplus \mathrm{p}$	$k' \oplus u$
$k \oplus \mathrm{q}$	$k' \oplus v$
$k \oplus \mathrm{r}$	$k' \oplus w$

The elements of set P are ($k \oplus k'$, $k \oplus k' \oplus u$, $k \oplus k' \oplus v$, $k \oplus k' \oplus w$, $k \oplus k' \oplus p$, $k \oplus k' \oplus q$, $k \oplus k' \oplus r$, $k \oplus k' \oplus p \oplus u$, $\mathbf{k} \oplus \mathbf{k}' \oplus \mathbf{p} \oplus \mathbf{v}$, $k \oplus k' \oplus p \oplus w$, $k \oplus k' \oplus q \oplus u$, $k \oplus k' \oplus q \oplus v$, $k \oplus k' \oplus q \oplus w$, $\mathbf{k} \oplus \mathbf{k}' \oplus \mathbf{r} \oplus \mathbf{u}$, $k \oplus k' \oplus r \oplus v$, $k \oplus k' \oplus r \oplus w$).

Let us assume, $k \oplus k' \oplus p \oplus v = k \oplus k' \oplus r \oplus u$ (shown in bold in set P), or, $p \oplus v = r \oplus u$, or, $p \oplus v \oplus u = r$. This shows that all the elements in P are not unique. Now, further suppose, subkey $(k \oplus p)$ in Set A and subkey $(k' \oplus v)$ in Set B correspond to the actual subkey for F^2.

$\Longrightarrow k \oplus p = k' \oplus v$
$\Longrightarrow k \oplus k' = p \oplus v$
$\Longrightarrow k' = k \oplus p \oplus v$

Rewriting elements of Set B in terms of k we get Table 1 which reduces to Table 2 as shown below:

Here, we can see that two keys are common in both the sets, i.e., subkeys $(k \oplus p)$ and $(k \oplus r)$. This happened because, as $k \oplus k' \oplus p \oplus v = k \oplus k' \oplus r \oplus u$, $(p \oplus v \oplus u)$ got substituted by r in Set B and hence the overlap.

Thus, for the second differential characteristics in our attack, if the set P has repeated elements, then we cannot ensure that a distinct single subkey candidate will be obtained.

Table 1

Set A	Set B
k	$k \oplus p \oplus v$
$\mathbf{k} \oplus \mathbf{p}$	$\mathbf{k} \oplus \mathbf{p} \oplus \mathbf{v} \oplus \mathbf{u}$
$k \oplus \mathrm{q}$	$\mathbf{k} \oplus \mathbf{p} \oplus \mathbf{v} \oplus \mathbf{v}$
$\mathbf{k} \oplus \mathbf{r}$	$k \oplus p \oplus v \oplus w$

Table 2

Set A	Set B
k	$k \oplus p \oplus v$
$\mathbf{k} \oplus \mathbf{p}$	$\mathbf{k} \oplus \mathbf{r}$
$k \oplus q$	$\mathbf{k} \oplus \mathbf{p}$
$\mathbf{k} \oplus \mathbf{r}$	$k \oplus p \oplus v \oplus w$