Noname manuscript No.
(will be inserted by the editor)

Practical Complexity Differential Cryptanalysis and Fault Analysis of AES

Michael Tunstall

the date of receipt and acceptance should be inserted later

Abstract This paper presents a survey of practical complexity differential cryptanalysis of AES and compares this to attacks that have been proposed for differential fault analysis. Naturally, the attacks in each vein of research are applicable in the other but use different models. In this paper we draw from both topics to improve attacks proposed in the literature. We re-evaluate the so-called Square attack and the use of impossible differentials in terms of differential fault analysis using a weaker model than previously considered in the literature. Furthermore, we propose two new attacks applicable to both differential cryptanalysis and differential fault analysis. The first is a differential cryptanalysis of four-round AES based on a differential that occurs with a non-negligible probability. The second is an application of the Square attack to a five-round AES that requires 2^8 ciphertexts and a time complexity equivalent to approximately $2^{37.5}$ AES encryptions.

Keywords Differential Cryptanalysis · Advanced Encryption Standard · Differential Fault Analysis

1 Introduction

The Advanced Encryption Standard (AES) [14] was standardized in 2001 from a proposal by Daemen and Rijmen [9,10]. It has since been analyzed with regard to numerous attacks ranging from purely theoretical cryptanalysis to attacks that require some extra information, from some side channel for example [19], to succeed. In this paper we present a survey of differential cryptanalysis of AES and present improvements where possible, and contrast this with differential fault analysis.

Differential cryptanalysis analyses pairs of plaintexts and seeks to exploit how the difference between these plaintexts propagates through a block cipher. Typically, reduced round variants of block ciphers are analyzed, since analyzing an entire modern block cipher is typically computationally infeasible. Differential fault analysis uses similar techniques to analyze differences induced in an instance of a block cipher. These differences are typically considered to be induced by a fault in the computation provoked by an attacker. The mechanisms for inducing faults into a microprocessor range from light [24] to glitches in the signals supplied to a microprocessor [3].

In this paper we give a summary of key recovery attacks on reduced round variants of AES and how these apply to differential fault analysis. In our analysis we define an attack to be practical, rather than purely theoretical, if it requires an effort equivalent to less than 2^{56} AES encryptions as proposed by Biryukov et al. [5]. To this end we describe attacks using the chosen plaintext model, and further define a chosen difference model that corresponds to faults being injected in the computation of a block cipher a certain number of rounds before the end of a block cipher. We define a differential cryptanalysis of a four-round variant of AES, that is valid in both models.

Phan and Yen [22] evaluated the use of the Square attack [9,10] and impossible differentials [4] to attack an instance of AES where faults have been injected at certain points during the computation of a block cipher. This attack has been re-evaluated by Kim [15] using a weaker model and proposes versions of this attack that require fewer plaintexts. In this paper we define the relationship between the number of acquisitions and the

Department of Computer Science, University of Bristol,
Merchant Venturers Building, Woodland Road,
Bristol BS8 1UB, United Kingdom.
E-mail: tunstall@cs.bris.ac.uk

effort required to conduct an attack and provide a new version of the Square attack applied to a five-round AES.

We do not consider attacks that have been designed to attack more than five rounds, and thus have a high complexity, such as the meet-in-the-middle attacks [11] and impossible differential attacks [7,17] that have been proposed to attack reduced round variants of AES.

The organization of this paper is as follows. In Section 2 we define the notation and operations that we will use in this paper to describe AES. In Section 3 we define a model for an attacker, and describe some observations concerning AES that we will refer to later in Section 4. We describe attacks based on differential cryptanalysis and the square attack in Sections 5 and 6 respectively. These attacks are summarized in Section 7, and we conclude in Section 8.

2 Preliminaries

In this paper, multiplications are considered to be polynomial multiplications over $\mathbb{F}_{2^8}$ modulo the irreducible polynomial $x^8+x^4+x^3+x+1$. It should be clear from the context when a mathematical expression contains integer multiplication.

2.1 The Advanced Encryption Standard

Algorithm 1: The AES-128 encryption function.

Input: The 128-bit plaintext block P and key K.
Output: The 128-bit ciphertext block C.

```
1  X ← AddRoundKey(P, K)
2  for i ← 1 to 10 do
3  |   X ← SubBytes(X)
4  |   X ← ShiftRows(X)
5  |   if i ≠ 10 then
6  |   |   X ← MixColumns(X)
7  |   end
8  |   K ← KeySchedule(K, i)
9  |   X ← AddRoundKey(X, K)
10 end
11 C ← X
12 return C
```

The structure of the Advanced Encryption Standard (AES) , as used to perform encryption, is illustrated in Algorithm 1. Note that we restrict ourselves to considering AES-128 and that in discussing the AES we consider that all variables are arranged in a 4×4 array of bytes, known as the state matrix. For example the 128-bit plaintext $P = (p_1, p_2, \ldots, p_{16})_{(256)}$ is arranged as follows:

$$\begin{pmatrix} p_1 & p_5 & p_9 & p_{13} \\ p_2 & p_6 & p_{10} & p_{14} \\ p_3 & p_7 & p_{11} & p_{15} \\ p_4 & p_8 & p_{12} & p_{16} \end{pmatrix}$$

The encryption itself is conducted by the repeated use of a round function that comprises the following operations executed in sequence:

The `SubBytes` operation is the only nonlinear step of the block cipher, consisting of a substitution table applied to each byte of the state. This replaces each byte of the state matrix by its multiplicative inverse, followed by an affine mapping. Thus the input byte x is related to the output y by $y = A\,x^{-1} + B$, where A and B are constant matrices. In the remainder of this paper we will refer to the function S as this substitution table and S^{-1} as its inverse.

The `ShiftRows` operation is a byte-wise permutation of the state that operates on each row.

The `KeySchedule` operation generates the next round key from the previous one. The first round key is the input key with no changes, subsequent round keys are generated using the function S, defined above, and XOR operations. The round counter is required as a different constant (RCON) is used each time the operation is instantiated.

The `MixColumns` operation operates on the state column by column. Each column of the state matrix is considered as a vector where each of its four elements belong to $\mathbb{F}(2^8)$. A 4×4 matrix M whose elements are also in $\mathbb{F}(2^8)$ is used to map this column into a new vector. This operation is applied to the four columns of the state matrix. Here M and its inverse M^{-1} are defined as

$$M = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix} \quad \text{and} \quad M^{-1} = \begin{pmatrix} 14 & 11 & 13 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{pmatrix} .$$

All the elements in M and M^{-1} are elements of $\mathbb{F}(2^8)$ expressed in decimal.

The `AddRoundKey` operation XORs each byte of the array with a byte from a corresponding subkey.

3 Attack Model and Complexity

In this section we define the attack models and the maximum complexity we will consider so that we can describe attacks on reduced round variants of AES in later sections.

3.1 Attack Model

In the remainder of this paper we detail some low complexity attacks on a reduced round AES. We use two models to describe our attacks:

1. **Chosen Plaintext Model** — This is one of the typical scenarios used in the differential cryptanalysis of a block cipher. An attacker is able to encipher arbitrary plaintexts under a fixed unknown key and obtain the ciphertext. The practicality of an attack is influenced by the number of chosen plaintexts required to conduct a given attack. The time complexity of attacks in this model is considered to the number of enciphering operations, or equivalent, of the algorithm under attack.
2. **Chosen Difference Model** — In this case an attacker is able to encipher two related but unknown plaintexts. That is, an attacker is able to encipher two plaintexts with a chosen difference. Note that this does not mean that the exact difference can be chosen by an attacker, but that the number and position of bytes that have a non-zero difference can be chosen. The practicality of the attack is influenced by the number of pairs of ciphertexts required that are produced by plaintexts produced with a difference of a known size. The time complexity of attacks in this model is considered to the number of enciphering operations, or equivalent, of a full 10-round AES. We also assume that the attacker has access to an oracle that can be used to test whether a given key hypothesis is correct. A justification for this model is given below in Section 3.3.

In the following sections we consider how attacks can be mounted on a reduced round version of AES, and we assume that the last round does not contain a `MixColumns` function as in the full AES. This is important to note since it has been shown that the lack of a `MixColumns` function does have a negative impact on the security of AES [12]. Indeed, many of the attacks described in the paper would have a substantially higher complexity if the `MixColumns` function was included in the last round.

3.2 Attack Complexity

In the models defined above there are two things that dictate the complexity of the attack. The first is the number of observations (either plaintext-ciphertext pairs or ciphertext pairs) that need to be acquired by an attacker. In the chosen plaintext model this can be considered to be straightforward. However, in the chosen difference model acquiring a ciphertext pair requires a certain amount of effort. Typically, this involves obtaining one ciphertext by enciphering an arbitrary plaintext, and then obtaining a faulty ciphertext by enciphering the same plaintext while injecting a fault in the device under attack. This typically takes a considerable amount of time when compared to a standard PC implementation, since fault attacks are usually considered to be limited to resource constrained devices such as smart cards. A more significant problem is that each fault injection will stress a device and will, with a certain probability, render it inoperable. An attacker will therefore seek to minimize the number of fault injections required to conduct an attack. We do not define a strict upper limit to the number of acquisitions since this will vary considerably for different fault injection mechanisms.

The second is computational complexity of the analysis required to derive the secret key. Biryukov et al. define the upper bound of such an attack to be the effort equivalent to computing 2^{56} AES operations [5]. They cite the computation of 2^{55} DES operations several years ago and the unsuccessful collision finding effort on SHA-1 that was expected to require 2^{61} operations.

3.3 On the Chosen Difference Model

In describing fault attacks one typically uses an attack model where an attacker is able to affect a single byte during the computation of AES. There are numerous examples in the literature of fault attacks where a byte has been modified allowing an implementation of AES to be broken. Typically, these are examples of an attack proposed by Piret and Quisquater [23] that is described in Section 5.2. An example of an attack that produces a modification on chosen bytes is only likely to be possible in certain circumstances. For example, Ali et al. [1] describe a fault that affects a diagonal line across the state matrix (as required by one of the attacks described in Section 5.4). However, no details are given on why this could be produced are given, and one has to assume that the effect is implementation specific.

The role of this model is to encompass what is possible by a strong attacker, and to find a middle ground between the generic and the specific. Many fault injection mechanisms will correspond to a weaker model and only a subset of the attacks described in this paper will be possible. However, some of the attacks that require strong assumptions have been shown to be possible in some instances [1]. Other mechanisms will be more specific and will fall outside the proposed model. For example, faults have been described that affect a particular

opcode on a given microprocessor that allows very specific attacks to be defined [2]. The intention behind this model is to thoroughly evaluate the generic case rather than use a weak model requiring that attacks on specific effects are designed on an *ad hoc* basis.

4 Observations

In this section we describe some observations on the structure of AES that we will refer to in later sections.

Observation 1: Given a differential between two values x, y, where $x \neq y$ and the non-zero XOR difference between these values is $x \oplus y = \alpha$. For a known β there will exist a certain number of values for x and y that will satisfy $S(x) \oplus S(y) = \beta$. For a random differences α, β there will be four solutions with a probability $1/256$, two solutions with a probability $126/256$ and zero solutions with a probability $129/256$. This is a well-known result and the basis for all differential cryptanalysis of AES.

Observation 2: We consider $y = M\,x$ for

$$y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

where each x_i and y_j, for $i, j \in \{1, 2, 3, 4\}$, is an element in $\mathbb{F}_{2^8}$ and M is defined above in Section 2. Given any four bytes from

$$\{y_1, y_2, y_3, y_4, x_1, x_2, x_3, x_4\},$$

the remaining four can be computed. Trivially, we know this is also true if we consider the differentials, since, if $y = M\,x$ and $y' = M\,x'$ then $y \oplus y' = M\,(x \oplus x')$. This is equivalent to the fact that the words (x, y) form an MDS code.

Observation 3: Again, we consider $y = M\,x$ for

$$y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

where each x_i and y_j, for $i, j \in \{1, 2, 3, 4\}$, is an element in $\mathbb{F}_{2^8}$ and M is defined above in Section 2. The number of bytes in y that are zero given a certain number of bytes of x are equal to zero can be counted for all possible inputs. This gives a probability of observing the number of output bytes set to zero (n) given the number of input bytes that are set to zero (m), given $n, m \in \{0, 1, 2, 3, 4\}$, and are tabulated in Table 1.

5 Differential Cryptanalysis

In this section we describe attacks based on differential cryptanalysis. We describe attacks based on other properties of AES in the next section.

5.1 Analyzing One-Round AES

It has been demonstrated by Bouillaguet et al. [6] that one known plaintext-ciphertext pair can enable an attacker to mount an attack by inspection that requires an exhaustive search of 2^{40}. This attack functions by guessing 40 bits of one subkey and, combined with the knowledge of the plaintext and ciphertext, deducing a possible key. For brevity, we do not consider this form of attack since it is only applicable to very few rounds. Moreover, a straightforward attack is widely known where two plaintext-ciphertext pairs are available, and is described here for completeness.

5.1.1 Chosen Difference

If an attacker is only able to determine the amount of bytes that are different between two plaintexts there is no information to exploit since the difference is unknown.

5.1.2 Chosen Plaintext

An attacker defines two plaintexts where the XOR difference between the two plaintexts is not zero for any byte. One can then look at the which bytes of the last subkey can produce the observed difference in the two input plaintexts. This will give 2^{16} hypotheses for the last subkey. Bouillaguet et al. note that the number of hypotheses can be further reduced if bytes of the first subkey are generated independently and verified. This reduces the time complexity of an attack from 2^{16} encryptions to 2^{12} applications of the `SubBytes` function to individual bytes [6].

5.2 Analyzing Two-Round AES

5.2.1 Chosen Difference

If an attacker sets a difference in the first byte in each column of state matrix of the plaintext the difference will propagate as shown in Figure 1. In this example an attacker has a difference in the plaintexts at indexes $\{1, 5, 9, 13\}$.

Table 1 Given a number of differentials in bytes on input to the `MixColumns` function the probability of this affecting a given number of output bytes is shown.

No of Affected Bytes	Out(0)	Out(1)	Out(2)	Out(3)	Out(4)
In(0)	1	0	0	0	0
In(1)	0	0	0	0	1
In(2)	0	0	0	$\frac{4}{255} \approx \frac{1}{2^6}$	$\frac{251}{255}$
In(3)	0	0	$\frac{2}{12675} \approx \frac{1}{2^{13.4}}$	$\frac{1004}{65025} \approx \frac{1}{2^6}$	$\frac{12803}{13005}$
In(4)	0	$\frac{4}{16581375} \approx \frac{1}{2^{22}}$	$\frac{502}{5527125} \approx \frac{1}{2^{13.4}}$	$\frac{51212}{3316275} \approx \frac{1}{2^6}$	$\frac{3264761}{3316275}$

$$\underset{\text{Plaintext}}{\begin{pmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}} \rightarrow \underset{\text{First Round}}{\begin{pmatrix} 2\,\alpha & 2\,\beta & 2\,\gamma & 2\,\delta \\ 3\,\alpha & 3\,\beta & 3\,\gamma & 3\,\delta \\ \alpha & \beta & \gamma & \delta \\ \alpha & \beta & \gamma & \delta \end{pmatrix}} \rightarrow \underset{\text{Ciphertext}}{\begin{pmatrix} x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \\ x_4 & x_8 & x_{12} & x_{16} \end{pmatrix}}$$

Fig. 1 Propagation of a four-byte difference across two rounds of AES.

An attacker can verify key hypotheses on the last subkey in groups of four bytes by checking that hypotheses will produce the differences caused by the (unknown) values $\{\alpha, \beta, \gamma, \delta\}$ [23]. An attacker can conduct an exhaustive search on the last subkey in four groups of 2^8 hypotheses, and this will reduce the key hypotheses for the last subkey to 2^{32} for the reasons noted in Observation 1.

5.2.2 Chosen Plaintext

Attacking a two round AES with chosen plaintexts has the same start as the attack under the chosen difference model. However, an attacker has more information given that the plaintexts are known and some of the 2^{32} hypotheses can be eliminated.

If we consider Figure 1, we know θ_i, for $i \in \{1, 2, 3, 4\}$, and the corresponding plaintexts. Given Observation 1 we can note that $\alpha, \beta, \gamma, \delta$ will each have 2^7 possible values rather than 2^8 considered in the previous attack. This means that an attacker would be able to reduce the key hypotheses to 2^{28} [6].

In both cases the same analysis can be conducted if an attacker changes bytes at indexes $\{2, 6, 10, 14\}$, $\{3, 7, 11, 15\}$ or $\{4, 8, 12, 16\}$, i.e., one row of the state matrix.

5.3 Analyzing Three-Round AES

5.3.1 Chosen Difference

If an attacker sets a difference in one byte of the plaintexts it will propagate as shown in Figure 2. In this example an attacker has a difference in the plaintexts at index 1.

As with the attack on two rounds described above, an attacker can verify key hypotheses on the last subkey in groups of four bytes by checking that hypotheses will produce the differences caused by $\{\alpha, \beta, \gamma, \delta\}$. Again, this will reduce the key hypotheses for last subkey to 2^{32} [23] for the reasons noted in Observation 1.

This analysis can be continued by verifying which key values also produce the differences $\{2\,\theta, 3\,\theta, \theta, \theta\}$ shown in Figure 2. This will reduce the number of key hypotheses for the last subkey to 2^8 [25]. This will require 2^{32} operations consisting of one `SubBytes` operation and a quarter of `MixColumns` operation. If we estimate this to equate to the effort required to compute a third of a round of AES, the time complexity of the attack will be $2^{32} \cdot \frac{1}{3} \cdot \frac{1}{10} \approx 2^{27}$.

5.3.2 Chosen Plaintext

Attacking a three round AES with chosen plaintexts has the same start as the attack described above for the chosen difference model. As for the attack on two round AES, an attacker has more information given that the plaintexts are known. Given Observation 1 we can note that θ will have 2^7 possible values rather than the 2^8 considered in the previous attack, reducing the number of key hypotheses from 2^8 to 2^7. As above, this will require 2^{32} one round deciphering operations that will have a time complexity of $2^{32} \cdot \frac{1}{3} \cdot \frac{1}{3} \approx 2^{29}$.

5.4 Analyzing Four-Round AES

5.4.1 Chosen Difference

An attacker can create two plaintexts with a difference in four bytes along a diagonal line within the state matrix. In Figure 3 we show an example where the bytes at indexes $\{1, 6, 11, 16\}$ represented by ζ_i for i in $\{1, 2, 3, 4\}$. These four bytes will produce a difference in one column after the first round, shown as θ_i for i in $\{1, 2, 3, 4\}$. There is a structure in the difference at the end of the second round. However, this cannot be directly exploited since there is no straightforward way of separating the key bytes to derive hypotheses on small

$$\underset{\text{Plaintext}}{\begin{pmatrix}\zeta & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{pmatrix}} \rightarrow \underset{\text{First Round}}{\begin{pmatrix}2\theta & 0 & 0 & 0\\ 3\theta & 0 & 0 & 0\\ \theta & 0 & 0 & 0\\ \theta & 0 & 0 & 0\end{pmatrix}} \rightarrow \underset{\text{Second Round}}{\begin{pmatrix}2\alpha & \beta & \gamma & 3\delta\\ 3\alpha & 2\beta & \gamma & \delta\\ \alpha & 3\beta & 2\gamma & \delta\\ \alpha & \beta & 3\gamma & 2\delta\end{pmatrix}} \rightarrow \underset{\text{Ciphertext}}{\begin{pmatrix}x_1 & x_5 & x_9 & x_{13}\\ x_2 & x_6 & x_{10} & x_{14}\\ x_3 & x_7 & x_{11} & x_{15}\\ x_4 & x_8 & x_{12} & x_{16}\end{pmatrix}}$$

Fig. 2 Propagation of a one-byte difference across three rounds of AES.

$$\underset{\text{Plaintext}}{\begin{pmatrix}\zeta_1 & 0 & 0 & 0\\ 0 & \zeta_2 & 0 & 0\\ 0 & 0 & \zeta_3 & 0\\ 0 & 0 & 0 & \zeta_4\end{pmatrix}} \rightarrow \underset{\text{First Round}}{\begin{pmatrix}\theta_1 & 0 & 0 & 0\\ \theta_2 & 0 & 0 & 0\\ \theta_3 & 0 & 0 & 0\\ \theta_4 & 0 & 0 & 0\end{pmatrix}} \rightarrow \underset{\text{Second Round}}{\begin{pmatrix}2\gamma_1 & \gamma_2 & \gamma_3 & 3\gamma_4\\ 3\gamma_1 & 2\gamma_2 & \gamma_3 & \gamma_4\\ \gamma_1 & 3\gamma_2 & 2\gamma_3 & \gamma_4\\ \gamma_1 & \gamma_2 & 3\gamma_3 & 2\gamma_4\end{pmatrix}} \rightarrow \underset{\text{Third Round}}{\begin{pmatrix}\epsilon_1 & \epsilon_5 & \epsilon_9 & \epsilon_{13}\\ \epsilon_2 & \epsilon_6 & \epsilon_{10} & \epsilon_{14}\\ \epsilon_3 & \epsilon_7 & \epsilon_{11} & \epsilon_{15}\\ \epsilon_4 & \epsilon_8 & \epsilon_{12} & \epsilon_{16}\end{pmatrix}} \rightarrow \underset{\text{Ciphertext}}{\begin{pmatrix}x_1 & x_5 & x_9 & x_{13}\\ x_2 & x_6 & x_{10} & x_{14}\\ x_3 & x_7 & x_{11} & x_{15}\\ x_4 & x_8 & x_{12} & x_{16}\end{pmatrix}}$$

Fig. 3 Propagation of a four-byte difference across four rounds of AES.

portions of a subkey. That is, there does not appear to be a method of deriving the secret key that would be significantly quicker than an exhaustive key search.

However, if we assume that $\theta_2 = 0$ an attack can be constructed. The propagation of this difference is shown in Figure 4 where a structure is produced at the end of the third round that can be exploited. This attack proceeds in a similar manner to the attacks described in Section 5.3. For each column there are 2^{24} possible differences (see Observation 2) that will allow 2^{24} key hypotheses to be determined for a column of the last subkey (see Observation 1), leading to 2^{96} key hypotheses for the last subkey. Likewise, if for example $\theta_1 = 0$ and $\theta_2 = 0$ the same reasoning would lead to 2^{64} key hypotheses for the last subkey.

If we assume that an attacker is somehow able to determine that $\theta_2 = 0$, then four acquisitions would be enough to reduce the number of key hypotheses to one. Furthermore, any remaining hypotheses could be verified by checking that the difference at the end of the second round has the structure shown in Figure 4 for each of the four pairs of ciphertexts.

The probability of one of θ_i, for $i \in \{1, 2, 3, 4\}$, being equal to zero is $1/2^6$ as described in Observation 3. One would therefore expect to require $4 \times 2^6 = 256$ ciphertexts to have four ciphertexts where at least one of θ_i is equal to zero. However, an attacker has no way of knowing how many, or which, different bytes of θ_i have been set to zero. An attacker is, therefore, obliged to test all the possibilities combinations in a set of 256 acquisitions to find them. This will contain $4^4\binom{256}{4} \approx 2^{35}$ combinations.

In a simulation of this attack we have found that evaluating a set of four pairs of ciphertexts corresponds to approximately 2^{25} executions of a 10-round AES. One can also note that the 4^4 possible combinations where one of, θ_i for i in $\{1, 2, 3, 4\}$, can be treated in parallel with negligible increase in the required computational effort. The expected computational effort required to conduct an attack therefore corresponds to $\binom{256}{4}2^{25} \approx 2^{52}$.

The same reasoning can be applied if an attacker can determine more than one of θ_i, for $i \in \{1, 2, 3, 4\}$, is equal to zero. These attacks are summarized in Table 2. As one would expect, there is a trade-off between the number of pairs of ciphertexts and the computational effort required to achieve an attack.

5.4.2 Chosen Plaintext

The same differential cryptanalysis can be applied to chosen plaintexts in a straightforward manner, with the added advantage that an attacker can be sure that each plaintext is distinct. The attack can be further optimized by comparing every ciphertext with every other ciphertext. That is, $\binom{11}{8} = 330$ comparisons can be made from 11 chosen plaintext-ciphertext pairs reducing the number of acquisitions required where one assumes that one of θ_i, for $i \in \{1, 2, 3, 4\}$, is equal to zero. The modifications to the attack are summarized in Table 3.

An attack described by Bouillaguet et al. [6] also functions on four rounds and requires ten chosen plaintexts, where the plaintexts have a difference in the bytes on one column of the state matrix. This functions by partially decrypting the last two rounds by guessing 40 bits of the last two subkeys to determine one byte of the state matrix at the end of the second round. These bytes can be XORed together to get the difference at the output of the `Mixcolumns` operation in the second round. The same thing is done with the plaintext, where 40 bits of the first two subkeys allow us to determine the input of one byte to the `Mixcolumns` operation in the second round. Given that the plaintext is chosen to ensure that the XOR difference of all the other bytes in the column at this point have an XOR difference of zero the difference of the output of the `MixColumns` operation for this byte can be computed. This is repeated for all ten chosen plaintexts and should lead to a re-

$$\underset{\text{Plaintext}}{\begin{pmatrix}\zeta_1 & 0 & 0 & 0\\ 0 & \zeta_2 & 0 & 0\\ 0 & 0 & \zeta_3 & 0\\ 0 & 0 & 0 & \zeta_4\end{pmatrix}} \rightarrow \underset{\text{First Round}}{\begin{pmatrix}\theta_1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ \theta_3 & 0 & 0 & 0\\ \theta_4 & 0 & 0 & 0\end{pmatrix}} \rightarrow \underset{\text{Second Round}}{\begin{pmatrix}2\gamma_1 & \gamma_2 & \gamma_3 & 0\\ 3\gamma_1 & 2\gamma_2 & \gamma_3 & 0\\ \gamma_1 & 3\gamma_2 & 2\gamma_3 & 0\\ \gamma_1 & \gamma_2 & 3\gamma_3 & 0\end{pmatrix}} \rightarrow$$

$$\underset{\text{Third Round}}{\begin{pmatrix}2a_1 \oplus 3a_2 \oplus a_3 & 2b_1 \oplus 3b_2 \oplus b_3 & 2c_1 \oplus c_2 \oplus c_3 & 3d_1 \oplus d_2 \oplus d_3\\ a_1 \oplus 2a_2 \oplus 3a_3 & b_1 \oplus 2b_2 \oplus b_3 & c_1 \oplus 3c_2 \oplus c_3 & 2d_1 \oplus 3d_2 \oplus d_3\\ a_1 \oplus a_2 \oplus 2a_3 & b_1 \oplus b_2 \oplus 3b_3 & c_1 \oplus 2c_2 \oplus 3c_3 & d_1 \oplus 2d_2 \oplus 3d_3\\ 3a_1 \oplus a_2 \oplus a_3 & 3b_1 \oplus b_2 \oplus 2b_3 & 3c_1 \oplus c_2 \oplus 2c_3 & d_1 \oplus d_2 \oplus 2d_3\end{pmatrix}} \rightarrow \underset{\text{Ciphertext}}{\begin{pmatrix}x_1 & x_5 & x_9 & x_{13}\\ x_2 & x_6 & x_{10} & x_{14}\\ x_3 & x_7 & x_{11} & x_{15}\\ x_4 & x_8 & x_{12} & x_{16}\end{pmatrix}}$$

Fig. 4 Propagation of a four-byte difference across four rounds of AES where $\theta_2 = 0$.

Table 2 A summary of the differential attacks applied to a four-round AES. The computational effort has been determined by simulation or through previous estimates (such as given in Section 5.3).

No. of θ_i equal to zero	Probability	Acquisitions	Computational Effort	Overall Time Complexity
1	$1/2^6$	256	2^{25}	$\binom{256}{4}2^{25} \approx 2^{52}$
2	$1/2^{13.4}$	$2^{14.4}$	2^{23}	$\binom{2^{14.4}}{2}2^{23} \approx 2^{51}$
3	$1/2^{22}$	2^{22}	2^{27}	$2^{22} \cdot 2^{27} = 2^{49}$

Table 3 A summary of the differential attacks applied to a four-round AES. The computational effort has been determined by simulation or through previous estimates (such as given in Section 5.3).

No. of θ_i equal to zero	Probability	Acquisitions	Computational Effort	Overall Time Complexity
1	$1/2^6$	12	2^{28}	$\binom{256}{4}2^{28} \approx 2^{55}$
2	$1/2^{13.4}$	30	2^{26}	$\binom{2^{14.4}}{2}2^{26} \approx 2^{54}$
3	$1/2^{22}$	2^{11}	2^{30}	$2^{22} \cdot 2^{30} = 2^{52}$

maining 2^{32} key hypotheses, from a computation with a time complexity of $10\left(\left(2^{40} \times 2 \times 4\right)/4\right) \approx 2^{44}$. We note that this attack is not possible in the chosen difference model.

6 Square Attack

The Square attack (so-called since it was first presented in the description of the block cipher Square [8]) was first presented in the original description of AES [9, 10]. The Square attack is based on a particular property arising from the structure of AES. For a set of 256 plaintexts where each byte at an arbitrary index is a distinct value and all the other bytes are equal, the XOR sum of the 256 state matrices at the end of the third round is a state matrix consisting of just zeros. This is because a given position in the state matrix will have all 256 possible values across the 256 state matrices before the last `MixColumns` operation. This property is lost after the `MixColumns` operation but the result of the XOR sum remains the same. Attacks based on this property in the context of our models are presented in this section.

6.1 Analyzing Four-Round AES

6.1.1 Chosen Difference

Phan and Yen describe an attack in the chosen difference model [22] based on the Square attack. If an attacker chooses a difference in one byte there are 256 possible ciphertexts that could be produced. Once an attacker has collected all 256 possible hypotheses the last subkey can be verified. This is achieved by testing hypotheses on the last subkey byte-by-byte where an attacker checks that the XOR sum of the input to the final round is equal to zero. This will return the correct subkey and one would also expect one additional incorrect hypothesis per byte, given that a random sequence will have an XOR sum equal to zero with a probability of 1/256. This will, therefore, result in an expected total number of key hypotheses for the last subkey of 2^{16}.

Kim notes that the number of hypotheses can be further reduced by analyzing the values of the bytes in the state matrix before the `MixColumns` operation in the penultimate round [15]. At each index the bytes will have a pairwise non-zero differential, i.e. across the 256 values at a given index there will be one of each possible value. This will determine the key value with an overwhelming probability, since the probability a random set of bytes at a given index will satisfy this condition is

$$\prod_{i=0}^{255} \frac{256-i}{256} \approx \frac{1}{2^{364}} \ .$$

Phan and Yen state that this attack would require 255 acquisitions [22] (one reference acquisition and 255 with a difference). However, this assumes that an attacker is able to determine the exact difference caused by the effect of a fault. In the chosen difference model used in this paper one would have to take enough acquisitions that all 256 ciphertexts have been seen. This is in an instance of the coupon collector's text described by Knuth [16], and one would expect $1561 \approx 2^{11}$ acquisitions to be taken before all the possible ciphertexts have been collected.

A related attack is based on the use of impossible differentials by Biham and Keller [4]. Any two ciphertexts in the Square attack described above will have a non-zero differential at the same point one would expect a XOR sum of zero for all 256 ciphertexts. In order to derive information on the last subkey, hypotheses can be verified in sets of 32 bits by conducting a partial decryption and assuring that the difference before the `MixColumns` operation in the penultimate round contains no bytes equal to zero. Conducting this analysis 2^{11} times allows the last subkey to be determined [22]. The time complexity of this attack is 2^{32} single round decryptions per ciphertext, i.e. $2^{11}\left(2^{32}/10\right) \approx 2^{39.5}$.

Biham and Keller [4] observed that an incomplete set of acquired ciphertexts that an attacker wishes to use to conduct a Square attack can be used to derive a secret key. This is possible since the a given number of distinct ciphertexts will have the same number of distinct values across the acquisitions for a given byte of the state matrix. Kim presents this as a variant of the Square attack [15]. However, one can observe that this also corresponds to an instance of impossible differential cryptanalysis since 57 distinct ciphertexts will allow a secret key to be determined since $\binom{57}{2} \approx 1561 \approx 2^{11}$ comparisons where a non-zero differential is present can be made (this another instance of the coupon-collectors problem). However, the time complexity is $57\left(2^{32}/10\right) \approx 2^{34.5}$ since one needs to verify that 57 values are distinct rather than having pairwise non-zero differentials. This is described in terms of fault analysis by Kim [15].

Given m different ciphertexts the expected number of remaining key hypotheses N for a 32 bits of the last subkey (corresponding to one column before the `MixColumn` in the penultimate round) can be defined as

$$\begin{aligned} N &= (2^{32}-1) \cdot \Pr(X=m)^4 + 1 \\ &= (2^{32}-1)\left(\frac{r_{(m)}}{r^m}\right)^4 + 1 \\ &= 1 + (2^{32}-1)\left(\frac{255_{(m)}}{255^m}\right)^4 \ , \end{aligned}$$

where $r_{(m)} = r\,(r-1)\ldots(r-m+1)$ and $\Pr(X=m)$ is the probability that the number of different values across m ciphertexts at a given position in the state matrix is equal to m. The time complexity of this variant of impossible differential cryptanalysis is shown in Figure 5 (left) given x acquired ciphertexts. The time complexity of the attack includes the effort required to produce the N ciphertexts. The time complexity falls below our practical bound to $2^{53.4}$ AES operations after 40 ciphertexts are acquired. The minimum time complexity is 2^{34} after 46 ciphertexts have been acquired, after which treating the ciphertext becomes more significant than the resulting exhaustive search.

Kim implicitly assumes that one can acquire m different ciphertexts with m faults. In the chosen difference model the amount of effort required to acquire m different ciphertexts is an instance of the classical occupancy problem. That is, how many distinct ciphertexts will one have after n faults have been injected. The probability of observing x different balls from r possibilities given n faults is

$$\Pr(x=k) = \frac{r_{(k)} \left\{ {n \atop i} \right\}}{r^n} \ ,$$

where we define $\left\{ {n \atop i} \right\}$ as a function that returns the Stirling numbers of the second kind. That is, the number of ways of partitioning n elements into i non-empty sets. The expectation of x is simply $E(x) = \sum_{i=1}^{r} i \Pr(x = i)$. The resulting time complexity when an attacker has to collect a certain number of ciphertexts is only slightly different given so few ciphertexts are required, as shown in Figure 5. The practical bound of $2^{53.4}$ AES operations occurs after 43 acquisitions, and the minimum time complexity of 2^{34} occurs at 51 acquisitions.

6.1.2 Chosen Plaintext

In the chosen plaintext model the Square attack, as described above, can be directly applied. That is, 256 plaintexts can be chosen that all differ on one byte

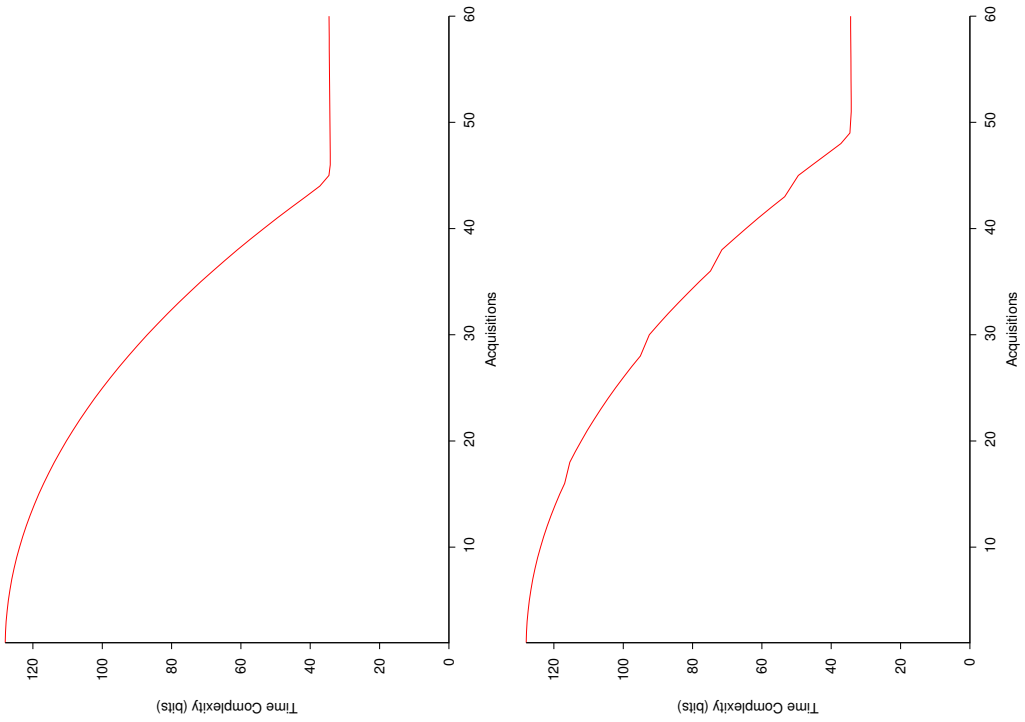

Fig. 5 The time complexity of the cryptanalysis of a four-round AES using impossible differential, where distinct bytes (left) and random bytes (right) are inserted into a fixed plaintext.

and the key uniquely identified from the result that the XOR sum of the intermediate state at the end of the third round is equal to zero. As above, the expected number of key hypotheses generated by the attack is 2^{16}. This can be improved if one repeats the analysis, i.e. one takes 2^9 chosen plaintexts and conducts the above analysis twice. This was originally detailed by Daemen and Rijmen in their AES proposal [9,10]. Kim's observation will also apply in this case and allow the number of hypotheses to be reduced to one from a set of 256 chosen plaintext-ciphertext pairs [15].

Attacks based on impossible differentials [4] will also function in the same manner, including the version based on acquisitions that would otherwise be used to conduct a Square attack. However, an attacker would be able to ensure that each plaintext is distinct and would not need to conduct acquisitions until a certain number of distinct ciphertexts have been collected.

6.2 Analyzing Five-Round AES

6.2.1 Chosen Difference

Phan and Yen describe an extension to the Square attack [22], that was also first presented in the original description of AES [9,10]. The Square attack can be extended by an extra round with an increase in the time complexity. Rather than analyzing the final subkey byte-by-byte one analyses the penultimate subkey.

In order to analyze the penultimate subkey one is obliged to guess 32 bits of the final subkey to determine one column of the state matrix before the XOR with the penultimate subkey. One can then compute the `MixColumns` operation on this column, and form hypotheses on a subkey equivalent to the penultimate subkey where one computes the `MixColumns` operation on the derived subkey to derive the penultimate subkey. Each evaluation reduces the potential key space by 1/256, and would need to conduct this analysis five times to uniquely determine the subkey [9,10].

If we define each of the 2^{32} partial decryptions as having a time complexity equivalent to 1/4 of a round the time complexity of the attack is

$$5\left(4\cdot 2^{40}/4\right)/10 \approx 2^{39} \ .$$

As in the Square attack described in Section 5.4, one cannot choose the values that are in the one-byte difference required to conduct the Square attack. One would therefore expect $5\times 1561 \approx 2^{13}$ acquisitions to be taken before all the possible ciphertexts have been collected.

One can envisage a much more efficient attack where one verifies that the hypotheses generated are valid subkeys as they are derived. That is, hypotheses on the

last subkey can verified, in than they must be generated from corresponding hypotheses on the penultimate subkey. One can start an analysis of 256 ciphertexts produced by encrypting 256 plaintexts that are identical except for one byte where each plaintext is distinct.

One can analyze two columns of the penultimate subkey by guessing eight bytes of the last subkey (in two sets of four bytes). This will give two sets of 2^{32} hypotheses with a time complexity $2 \cdot 2^{40}/4$ one round decryptions. One can then eliminate hypotheses in each set that are inconsistent. Given that we have hypotheses on eight bytes of the penultimate key and eight bytes of the last subkey, we will have three bytes of the last subkey that are produced by bytes included in these hypotheses. Given that the probability that all three bytes of a given hypotheses can be verified with a probability of $1/2^{24}$ one would expect that the two sets of 2^{32} hypotheses can be reduced to one set of 2^{40} hypotheses. A third set of 2^{32} hypotheses can then be generated for one of the remaining columns of the penultimate subkey and four bytes of the last subkey. There will be a further four bytes of the last subkey by that will be generated by bytes for which there are already hypotheses, and an element from the new set of of 2^{32} hypotheses will validate a hypothesis from the known set of 2^{40} hypotheses with a probability of $1/2^{32}$. One would therefore expect to combine these two sets to produce a set of 2^{40} hypotheses for 96 bits of the penultimate and 96 bits of the last subkey. A set of 2^{32} hypotheses can then be generated for the final column of the penultimate subkey and four bytes of the final subkey. At this point one can verify whether an entire subkey can be generated from the penultimate subkey. For each of the 2^{32} hypotheses generated there will be a $1/(2^8)^9 = 1/2^{72}$ (since there will be nine bytes in the last subkey that will not have been verified before). One would, therefore, expect to generate two hypotheses from the two sets of hypotheses. One that is correct and one fulfills the criteria by chance. A similar strategy has been proposed by Lucks for variants of AES with 192 and 256-bit keys [18].

In order to clarify how this attack functions an example is described below, although other versions will be possible (i.e. where one chooses to examine the columns of the penultimate subkey in a different order).

We define the last two subkeys as

$$K_9 = \begin{pmatrix} k_{9,1} & k_{9,5} & k_{9,9} & k_{9,13} \\ k_{9,2} & k_{9,6} & k_{9,10} & k_{9,14} \\ k_{9,3} & k_{9,7} & k_{9,11} & k_{9,15} \\ k_{9,4} & k_{9,8} & k_{9,12} & k_{9,16} \end{pmatrix}$$

and

$$K_{10} = \begin{pmatrix} k_{10,1} & k_{10,5} & k_{10,9} & k_{10,13} \\ k_{10,2} & k_{10,6} & k_{10,10} & k_{10,14} \\ k_{10,3} & k_{10,7} & k_{10,11} & k_{10,15} \\ k_{10,4} & k_{10,8} & k_{10,12} & k_{10,16} \end{pmatrix} .$$

If an attacker has a set of plaintexts where all the bytes except one are equal, and the remaining byte takes all possible values across the set of 256 plaintexts. One can conduct an analysis as described at the beginning of this section. That is, one can guess, for example, the key bytes $\{k_{10,1}, k_{10,14}, k_{10,11}, k_{10,8}\}$ which will allow hypotheses to be formed on each element of $\{k_{9,1}, k_{9,2}, k_{9,3}, k_{9,4}\}$ independently. Note that while the attack derives information on a transformed version of $\{k_{9,1}, k_{9,2}, k_{9,3}, k_{9,4}\}$ this can be easily inverted for each hypothesis for the whole set. With a time complexity of $2^{40}/4$ one round decryptions one would expect to obtain 2^{32} hypotheses for

$$\gamma_1 = \{k_{10,1}, k_{10,14}, k_{10,11}, k_{10,8}, k_{9,1}, k_{9,2}, k_{9,3}, k_{9,4}\} ,$$

which can be stored in a hash table.

One can then conduct the same analysis by making hypotheses on $\{k_{10,13}, k_{10,10}, k_{10,7}, k_{10,4}\}$ which will allow hypotheses to be derived on the elements of $\{k_{9,13}, k_{9,14}, k_{9,15}, k_{9,16}\}$. Likewise, this will provide 2^{32} hypotheses for

$$\gamma_2 = \{k_{10,13}, k_{10,10}, k_{10,7}, k_{10,4}, k_{9,13}, k_{9,14}, k_{9,15}, k_{9,16}\} .$$

However, as each element in γ_2 is generated the following relationships between the elements in γ_2 and γ_1 can be verified:

$$k_{10,1} = S(k_{9,14}) \oplus k_{9,1} \oplus \text{RCON}$$
$$k_{10,4} = S(k_{9,13}) \oplus k_{9,13} \quad k_{10,14} = k_{10,11} \oplus k_{9,14}$$

A given element of γ_2 will therefore have 2^8 elements in γ_1 that will satisfy these constraints, given that a incorrect key value will fulfill these relationships with probability $1/2^{24}$. The resulting 2^{40} hypotheses for $\{\gamma_1, \gamma_2\}$ can also be stored in a hash table.

One can continue in a straightforward manner to derive 2^{32} hypotheses for

$$\gamma_3 = \{k_{10,5}, k_{10,2}, k_{10,15}, k_{10,12}, k_{9,5}, k_{9,6}, k_{9,7}, k_{9,8}\} .$$

For each element in γ_3 an attacker can verify the following relationships with $\{\gamma_1, \gamma_2\}$:

$$k_{10,2} = S(k_{9,15}) \oplus k_{9,2}, \quad k_{10,5} = k_{10,1} \oplus k_{9,5}$$
$$k_{10,8} = k_{10,4} \oplus k_{9,8} \quad \text{and} \quad k_{10,14} = k_{10,10} \oplus k_{9,14}$$

Again the resulting 2^{40} hypotheses for $\{\gamma_1, \gamma_2, \gamma_3\}$ can be stored in a hash table.

Lastly, one can perform the same analysis on

$$\gamma_4 = \{k_{10,9}, k_{10,6}, k_{10,3}, k_{10,16}, k_{9,9}, k_{9,10}, k_{9,11}, k_{9,12}\} \ .$$

Each element from γ_4 can be used to see if a valid pair $\{K_9, K_{10}\}$ has been found by searching in $\{\gamma_1, \gamma_2, \gamma_3\}$ for values that satisfy the remaining relationships between the two subkeys.

$$\begin{aligned} k_{10,4} = S(k_{9,13}) \oplus k_{9,4}, \quad & k_{10,6} = k_{10,2} \oplus k_{9,6}, \\ k_{10,7} = k_{10,3} \oplus k_{9,7}, \quad & k_{10,9} = k_{10,5} \oplus k_{9,9}, \\ k_{10,10} = k_{10,6} \oplus k_{9,10}, \quad & k_{10,11} = k_{10,7} \oplus k_{9,11}, \\ k_{10,12} = k_{10,8} \oplus k_{9,12}, \quad & k_{10,13} = k_{10,9} \oplus k_{9,13}, \\ \text{and} \quad & k_{10,16} = k_{10,12} \oplus k_{9,16} \end{aligned}$$

Given that an incorrect hypothesis for $\{K_9, K_{10}\}$ will validate these equations with a probability of $\left(1/2^8\right)^9 = 1/2^{72}$, one would expect to have two hypotheses for $\{K_9, K_{10}\}$ (i.e. the correct key and one incorrect one that fulfills the criteria by chance) and therefore two hypotheses for the AES key. The overall time complexity of this attack would therefore be expected to be $\left(\left(4 \cdot 2^{40}/\,4\right)/10 \approx 2^{36.5}\right.$, and as with Phan and Yen's attack one would expect that 1561 acquisitions would need to be made to collect the 256 possible ciphertexts.

Using a set of acquisitions to exploit impossible differentials to recover a secret key is not described in the chosen difference model, since the time complexity of such an attack would not be below the bound set for a practical attack [7].

6.2.2 Chosen Plaintext

As above, the Square attack can be used to attack a five-round instance of AES in the chosen plaintext model by the method described above [9,10]. Naturally, this requires fewer plaintexts and has a slightly higher time complexity.

The new attack described above applies equally well to attacking an implementation to a five-round AES under the chosen plaintext model. One can choose 256 distinct plaintexts that differ in one byte. The resulting analysis will identical to that given above, but an attacker will be seeking to derive information on $\{K_4, K_5\}$ rather than $\{K_9, K_{10}\}$. The time complexity for this attack would be $\left(4 \cdot 2^{40}/4\right)/5 \approx 2^{37.5}$.

6.3 Analyzing More Rounds

There are attacks that conduct a differential cryptanalysis on an instance of AES that consist of up to eight rounds. These are extensions of the Square attack [13], impossible differentials [17] and meet-in-the-middle attacks [11], since attacks based on a straightforward differential spanning four rounds or more have been proven to be impossible [21]. However, these attacks typically require a large number of plaintexts and/or have a high time complexity, and are therefore beyond the scope of this paper.

7 Summary

In order to clarify the contribution of this paper the attacks presented in this paper are described alongside some of the most efficient attacks from the literature in Tables 4 and 5. We note that the numbers corresponding to fault attacks based on the Square attacks described in the literature [22,15] have been re-evaluated to fit our model. We assert that this represents a more accurate evaluation of these attacks since our model is intended to correspond to an attacker able to inject a fault at a given point in time during the computation of an AES.

We recall that the time complexity of Table 4 is defined in terms of a ten-round AES since it is intended to correspond to differential fault analysis, whereas the time complexity in Table 5 is defined in terms of the number of rounds in the variant under consideration.

8 Conclusion

In this paper we consider some low complexity differential cryptanalyses of reduced round variants of AES. Specifically, we consider instances of AES that consist of one to five rounds of AES in two attack models. The first is the standard chosen plaintext model considered in differential cryptanalysis. The second is a chosen difference model that is intended to correspond to faults being injected into an instance of AES a given number of rounds before the end of the block cipher.

We propose a new differential cryptanalysis of four rounds of AES under the chosen plaintext model that has a practical complexity. We further show that this attack can be readily applied to conduct a differential fault analysis. We also demonstrate that the Square attack has not been correctly evaluated as a fault attack, and is not efficient as described in the literature [22, 15]. We also provide a novel attack on a five-round AES based on the Square attack. A summary is given in Section 7.

The attacks described in this paper that correspond to the analysis of faults produced in an instantiation of AES have been evaluated using a chosen difference

Table 4 Summary of attacks under the chosen difference model.

	Rounds	Type	Acquisitions	Complexity
Piret and Quisquater [23,20]	2	Differential	1	2^{32}
Tunstall and Mukhopadhyay [25]	3	Differential	1	2^{27}
Phan and Yen [22]	4	Impossible Diff.	2^{11}	$2^{39.5}$
Phan and Yen [22]	4	Square	2^{11}	2^{16}
Biham and Keller [4,15]	4	Impossible Diff.	43	$2^{53.4}$
Biham and Keller [4,15]	4	Impossible Diff.	51	2^{34}
This paper	4	Differential	2^{8}	2^{52}
This paper	4	Differential	$2^{14.4}$	2^{51}
This paper	4	Differential	2^{22}	2^{49}
Phan and Yen [22]	5	Square	2^{13}	2^{39}
This paper	5	Square	2^{8}	$2^{36.5}$

Table 5 Summary of attacks under the chosen plaintext model.

	Rounds	Type	Acquisitions	Complexity
Bouillaguet et al. [6]	1	Differential	2	2^{12}
Bouillaguet et al. [6]	2	Differential	2	2^{28}
Bouillaguet et al. [6]	3	Differential	2	2^{32}
This paper	3	Differential	2	2^{29}
Bouillaguet et al. [6]	4	Differential	10	2^{44}
Biham and Keller [4]	4	Impossible Diff.	2^{11}	$2^{39.5}$
Daemen and Rijmen [9,10]	4	Square	2^{9}	2^{9}
Biham and Keller [4]	4	Impossible Diff.	40	$2^{53.4}$
Biham and Keller [4]	4	Impossible Diff.	46	2^{34}
This paper	4	Differential	12	2^{55}
This paper	4	Differential	30	2^{54}
This paper	4	Differential	2^{11}	2^{52}
Daemen and Rijmen [9,10]	5	Square	2^{11}	2^{40}
This paper	5	Square	2^{8}	$2^{37.5}$

model defined in Section 3.1. This model assumes a strong attacker who is able to affect a chosen number of bytes at a chosen set of positions in the state matrix. This allows for a broad range of attacks to be implemented, although not all attacks will be applicable to all methods of fault injection. Likewise, the effects produced by some fault injection mechanisms are not considered since they are likely to be specific to one family of microprocessor/FPGA.

Acknowledgements

The work described in this paper has also been supported in part the European Commission through the ICT Programme under Contract ICT2007216676 ECRYPT II and EPSRC via grant EP/I005226/1.

References

1. S. Ali, D. Mukhopadhyay, and M. Tunstall. Differential fault analysis of AES using a single multiple-byte fault. Cryptology ePrint Archive, Report 2010/636, 2010. `http://eprint.iacr.org/`.
2. F. Amiel, C. Clavier, and M. Tunstall. Collision fault analysis of DPA-resistant algorithms. In L. Breveglieri, I. Koren, D. Naccache, and J.-P. Seifert, editors, *FDTC 06*, volume 4236 of *LNCS*, pages 223–236. Springer, 2006.
3. H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan. The sorcerer's apprentice guide to fault attacks. *Proceedings the IEEE*, 94(2):370–382, 2006.
4. E. Biham and N. Keller. Cryptanalysis of reduced variants of Rijndael. unpublished, 1999. `http://www.madchat.fr/crypto/codebreakers/35-ebiham.pdf`.
5. A. Biryukov, O. Dunkelman, N. Keller, D. Khovratovich, and A. Shamir. Key recovery attacks of practical complexity on AES-256 variants with up to 10 rounds. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 299–319. Springer, 2010.
6. C. Bouillaguet, P. Derbez, O. Dunkelman, N. Keller, and P.-A. Fouque. Low data complexity attacks on AES. Cryptology ePrint Archive, Report 2010/633, 2010. `http://eprint.iacr.org/`.
7. J. H. Cheon, M. Kim, K. Kim, J.-Y. Lee, and S. Kang. Improved impossible differential cryptanalysis of Rijndael and Crypton. In K. Kim, editor, *ICISC 2001*, volume 2288 of *LNCS*, pages 39–49. Springer, 2002.
8. J. Daemen, L. Knudsen, and V. Rijmen. The block cipher Square. In E. Biham, editor, *FSE '97*, volume 1267 of *LNCS*, pages 149–165. Springer, 1997.
9. J. Daemen and V. Rijmen. AES proposal: Rijndael. In *AES Round 1 Technical Evaluation CD-1: Documentation*. NIST, August 1998. `http://www.nist.gov/aes`.
10. J. Daemen and V. Rijmen. *The Design of Rijndael: AES — The Advanced Encryption Standard*. Springer, 2002.

11. H. Demirci, I. Taşkin, M. Çoban, and A. Baysal. Improved meet-in-the-middle attacks on AES. In B. Roy and N. Sendrier, editors, *INDOCRYPT 2009*, volume 5922 of *LNCS*, pages 144–156. Springer, 2009.
12. O. Dunkelman and N. Keller. The effects of the omission of last round's MixColumns on AES. *Information Processing Letters*, 110(8–9):304–308, 2010.
13. N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, and D. Whiting. Improved cryptanalysis of Rijndael. In B. Schneier, editor, *FSE 2000*, volume 1978 of *LNCS*, pages 213–230. Springer, 2001.
14. FIPS PUB 197. Advanced encryption standard (AES). Federal Information Processing Standards Publication 197, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, November 2001.
15. C. H. Kim. Efficient methods for exploiting faults induced at AES middle rounds. Cryptology ePrint Archive, Report 2011/349, 2011. `http://eprint.iacr.org/`.
16. D. E. Knuth. *The Art of Computer Programming*, volume 2 / Seminumerical Algorithms. Addison-Wesley, 2nd edition, 1981.
17. J. Lu, O. Dunkelman, N. Keller, and J. Kim. New impossible differential attacks on AES. In D. R. Chowdhury, V. Rijmen, and A. Das, editors, *INDOCRYPT 2008*, volume 5365 of *LNCS*, pages 279–293. Springer, 2008.
18. S. Lucks. Attacking seven rounds of Rijndael under 196-bit and 256-bit keys. In *AES Candidate Conference 2000*, 2000. `http://csrc.nist.gov/encryption/aes/round2/conf3/aes3conf.htm`.
19. S. Mangard, E. Oswald, and T. Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer-Verlag, 2007.
20. D. Mukhopadhyay. An improved fault based attack of the Advanced Encryption Standard. In B. Preneel, editor, *AFRICACRYPT 2009*, volume 5580 of *LNCS*, pages 421–434. Springer, 2009.
21. S. Park, S. H. Sung, S. Chee, E.-J. Yoon, and J. Lim. On the security of Rijndael-like structures against differential and linear cryptanalysis. In Y. Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 176–191. Springer, 2002.
22. R. C.-W. Phan and S.-M. Yen. Amplifying side-channel attacks with techniques from block cipher cryptanalysis. In J. Domingo-Ferrer, J. Posegga, and D. Shreckling, editors, *CARDIS 2006*, volume 3928 of *LNCS*, pages 135–150. Springer, 2006.
23. G. Piret and J.-J. Quisquater. A differential fault attack technique against SPN structure, with application to the AES and KHAZAD. In C. D. Walter, Ç. K. Koç, and C. Paar, editors, *CHES 2003*, volume 2779 of *LNCS*, pages 77–88. Springer, 2003.
24. S. P. Skorobogatov. Semi-invasive attacks: A new approach to hardware security analysis. Technical Report UCAM-CL-TR-630, Computer Laboratory, University of Cambridge, April 2005.
25. M. Tunstall, D. Mukhopadhyay, and Subidh Ali. Differential fault analysis of the Advanced Encryption Standard using a single fault. In C. A. Ardagna and J. Zhou, editors, *WISTP 2011*, volume 6633 of *LNCS*, pages 224–233. Springer, 2011.