Received 23 December 2022, accepted 7 January 2023, date of publication 11 January 2023, date of current version 18 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3236240

METHODS

Bayesian Modeling for Differential Cryptanalysis of Block Ciphers: A DES Instance

VINCENZO AGATE, FEDERICO CONCONE, ALESSANDRA DE PAOLA, (Member, IEEE), PIERLUCA FERRARO, GIUSEPPE LO RE, (Senior Member, IEEE), AND MARCO MORANA
Department of Engineering, University of Palermo, 90128 Palermo, Italy

Corresponding author: Marco Morana (marco.morana@unipa.it)

ABSTRACT Encryption algorithms based on block ciphers are among the most widely adopted solutions for providing information security. Over the years, a variety of methods have been proposed to evaluate the robustness of these algorithms to different types of security attacks. One of the most effective analysis techniques is differential cryptanalysis, whose aim is to study how variations in the input propagate on the output. In this work we address the modeling of differential attacks to block cipher algorithms by defining a Bayesian framework that allows a probabilistic estimation of the secret key. In order to prove the validity of the proposed approach, we present as case study a differential attack to the Data Encryption Standard (DES) which, despite being one of the methods that has been most thoroughly analyzed, is still of great interest to the scientific community since its vulnerabilities may have implications on other ciphers.

INDEX TERMS Differential cryptanalysis, Bayesian networks, probabilistic inference, DES.

I. INTRODUCTION

Among the many different encryption methods adopted by the modern systems, algorithms operating on fixed-length *blocks* of bits are still one of the most popular. The strength of these methods is constantly being studied by means of approaches that aim to assess their robustness to specific attacks, or the presence of vulnerabilities to generic threats. In this context, *differential cryptanalysis* is one of the most effective and relevant approaches. The idea at the basis of differential cyrptanalysis is to evaluate how any change in the plaintext impacts the ciphertext. Then, the results of the analysis can be used to estimate the set of the most probable keys.

In this paper we present a Bayesian framework for modelling differential attacks to block cipher algorithms; in particular, given the importance of the Data Encryption Standard (DES) in the design of many block cipher algorithms, a case study focused on the cyrptanalysis of DES is addressed.

The Data Encryption Standard (DES) [1] was the first symmetric cipher heavily adopted all over the world and it was the most used cipher up to the beginning of 2000s. Deep analyses of DES led to the definition of several cryptanalysis techniques, and many results achieved for DES are also valid for the wider class of block ciphers.

Today, the limited size of the secret key adopted by DES (56 bit) and the computational power of modern computers entail that DES is not considered secure for ciphering sensitive data. Nevertheless, DES is still widely adopted in various scenarios, such as those characterized by low security requirements, if resource-constrained devices are required to implement security mechanisms, or when huge amount of data have to be protected. The authors of [2], for instance, propose the adoption of DES to ensure privacy in a graduate project management system. Similarly, the need to protect a large amount of data while keeping the computational costs low moved the authors of [3] to choose DES for data encryption in an ERP. DES is often exploited to protect data exchanged between Internet of Things devices, which are characterized by severe resource requirements [4], [5], [6]. DES could also be employed as a tool for providing companies with proper data protection policies that represent a fair trade-off between security goals and computational costs [7]. Moreover, DES is a building block of Triple DES [8], [9],

The associate editor coordinating the review of this manuscript and approving it for publication was Xiali Hei.

a solution adopted to overcome the limitations imposed by the DES key size. Thus, it is interesting to investigate the vulnerabilities of DES also for possible implications on other block ciphers.

Several works in the scientific literature identified and analyzed some of the main vulnerabilities of DES, through the definition of new cryptanalysis techniques. One of these approaches is the differential cryptanalysis [10], a chosen plaintext attack designed for iterated cryptosystems, which analyzes how the difference between two plaintexts propagates in the resulting ciphered texts when using the same key. The differential cryptanalysis focuses on the S-Box, the unique non-linear component of DES, and allows to reduce the computational cost in comparison with an exhaustive key search. Differential Cryptanalysis has been adopted also to perform attacks to other symmetric cyphers, such as AES [11]. Moreover, several machine learning approaches have been adopted in recent years to improve differential cryptanalysis, or to provide a new perspective on it. The authors of [12], for instance, proposed the adoption of neural networks to attack DES, and evaluated the performance by using different network structures. In [13], several meta-heuristics, such as genetic algorithms and simulated annealing, are exploited to formulate a differential attack on DES. The experiments performed on a DES reduced to six rounds demonstrates the suitability of the approach. The authors of [14] and [15] relied on deep neural networks to design a differential distinguisher to attack different block ciphers based on the Feistel network.

We propose an original formalization of the differential cryptanalysis based on the adoption of Bayesian Networks (BN), a probabilistic graph model framework that uses Bayesian inference to perform probability computations. We aim to describe the statistical behavior of S-Boxes when a pair of plaintexts, with a given difference, is provided for ciphering. The diagnostic inference enabled by BNs, allows a probabilistic estimation of the secret key, by considering the difference between plaintexts and the difference between the corresponding ciphered texts.

Such formalization, preliminary described in [16], eases the definition of an algorithm for attacking the DES, based on the differential cryptanalysis.

The paper is organized as follows. In Section II, a brief description of DES is provided, in order to introduce the adopted notation. Section III describes some related works presented in the literature. In Section IV, the original formulation of the differential cryptanalysis is introduced. Section V describes the proposed Bayesian model of the DES differential cryptanalysis. Finally, Section VII states our conclusions.

II. THE ADOPTED DES NOTATION

DES is a symmetric cipher which transforms a 64-bit plaintext P in a 64-bit ciphertext T. Such mapping is parameterized by a 64-bit key, reduced to a 56-bit key because of the use of 8 parity bits. It is an iterated cipher based on the Feistel scheme, which processes plaintext through a series of transformations named *rounds*, as showed in Fig. 1. The encryption process consists of 16 rounds, which are preceded by an initial permutation and followed by the corresponding inverse permutation. Each round is parameterized by a 48-bit subkey $S_{K_X} \in \mathbb{Z}_2^{48}$, depending on the round X and the initial key K.

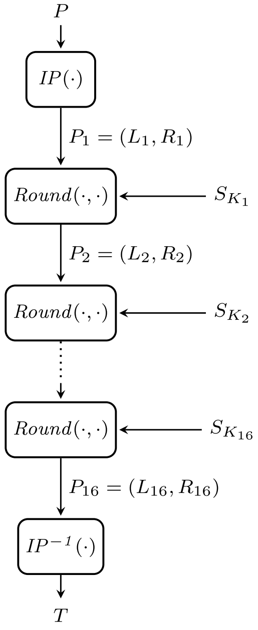

FIGURE 1. DES general block scheme.

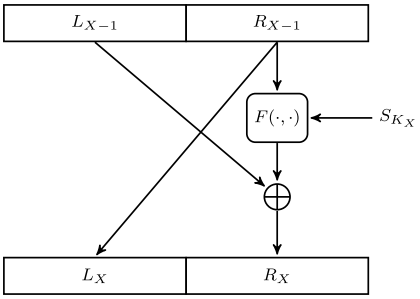

FIGURE 2. Block scheme of the round processing.

At each round X, the 64-bit input is divided into two parts, left and right, which are processed separately. The right part becomes the left one of the next round without any further processing. Both halves are processed according to the Feistel scheme, in order to produce the right part of the next round, as showed in Fig. 2. In particular, for each round $X = 2, \ldots, 16$, the following equations hold;

$$\begin{cases} L_X = R_{X-1}, \\ R_X = L_{X-1} \oplus F(R_{X-1}, S_{K_X}), \end{cases} \tag{1}$$

where F, named Feistel function, determines the non-linear behavior of DES.

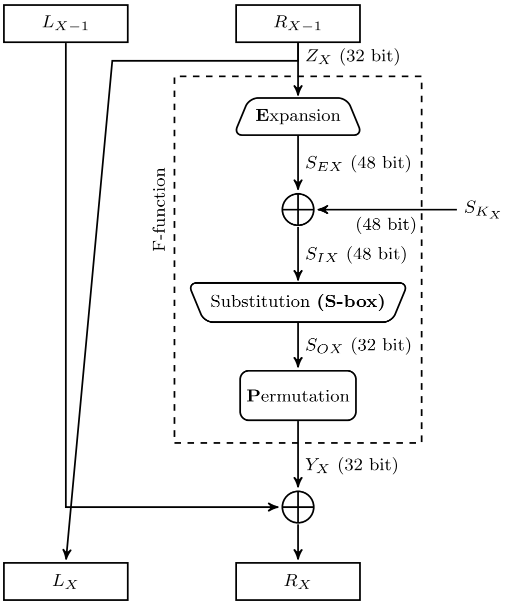

FIGURE 3. Detailed block scheme of DES Feistel function.

The Feistel function implemented by DES (see Fig. 3) is defined as follows:

$$F(R_{X-1}, S_{K_X}) = P(S(E(R_{X-1}) \oplus S_{K_X})), \quad (2)$$

where E, S and P represent respectively the *expansion* function, the *substitution* performed by the S-box and the *permutation* function.

Since the only non-linear component of the DES F-function is the S-box, it constitutes the main contributor to DES security. One of the properties of S-Box is the uniform distribution of the probability of producing a given output. Nevertheless, authors of [10] have shown that, taken two different inputs for a given S-Box characterized by some known difference, then the probability distribution of the difference between the corresponding outputs is not uniform. Differential cryptanalysis [10] exploits such a vulnerability in order to reduce the computational effort for determining the secret key, and the same idea underlies the approach described in the present work.

III. RELATED WORK

As previously mentioned, a deeper comprehension of S-Box behavior could make the whole cipher more vulnerable. Many works in the literature analyze properties of S-Boxes in order to find DES vulnerabilities and to define design criteria for strong block ciphers. Authors of [17] analyze properties of S-Boxes with respect to the statistical distributions of produced output and the statistical dependence of output bits given the knowledge of one or more input bits. In [18] some general criteria to design S-Boxes are discussed. Authors analyzed both static and dynamic properties. Static properties impose that partial information about input and output does not reduce the uncertainty of unknown input or output, and guarantee the maximum output uncertainty. Dynamic properties impose that partial information about changes in input and output does not reduce the uncertainty of unknown inputs or outputs. Authors stated that the uncertainty should not be reduced when the attacker has information about the past history of S-Boxes processing. Other studies indicate that the latest approaches [19], [20], [21], also known as *strong* S-Boxes, are vulnerable due to the adoption of fixed point or reverse fixed point, which can be an exploitable weakness in cryptography. Authors of [22], for example, address the exploitable weakness of fixed point and reverse fixed point contained in many S-Boxes. Then, they designed a S-Box construction algorithm based on ICQM that eliminates the weakness through backtracking.

On the basis of the properties discussed so far, many cryptanalysis methods were proposed in the literature to violate S-Boxes. An algebraic approach is proposed in [23], which defines the set of criteria to determine the set of non-linear algebraic constraints which describes the I/O relationship of S-Boxes. Exploiting this set of constraints, the whole cipher is described as a system of multivariate non-linear equations, that can be solved through the algorithm proposed in [24]. It should be noted that the equations representing S-Boxes are exact, i.e. not approximated. On the contrary, the author of [25] proposed a linear approximation of S-boxes and DES, which is valid with some probability. This method is an example of stochastic attack.

Instead of focusing on the behavior of a single S-Box, authors of [26] focus on the probabilistic behavior of pairs of adjacent S-Boxes. They found that input bits of two adjacent S-Boxes are strictly related by some bits of the key, due to the expansion phase. Thus, the probability distribution of the output of these two adjacent S-Boxes, conditioned on key bits is not uniform. On the basis of such vulnerability, authors proposed an attack with computational complexity comparable to the exhaustive key search.

Authors of [10], which propose the differential cryptanalysis, studied how input differences affect the resulting output difference. Their attack traces differences through the transformations, discovers where the cipher exhibits non-random behavior, and exploits such properties to recover the secret key. Another interesting work discussing the differential cryptoanalysis is presented in [27]. Here, the authors study the propagation of differences from round to round to find specific differences which propagate with relatively high probability. The cryptanalysis technique is applied to DES reduced to *i*-rounds, with $i \in [3, 8]$ and, for each, the differentiation between wrong and right pairs is made to get relevant key bits and retrieve the secret key.

IV. ORIGINAL FORMULATION OF DIFFERENTIAL CRYPTANALYSIS

The vulnerability at the basis of the differential cryptanalysis [10] originates from the non-uniform distribution of the difference between two outputs, given the difference between

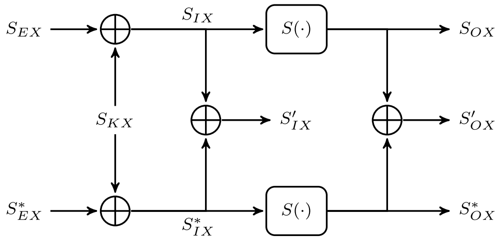

FIGURE 4. Notation adopted for describing differential cryptanalysis.

two inputs, for different keys. Nowadays, it is a technique adopted to breach many reduced-round block cyphers, such as SPECK [28], LEA [29], GIFT [30], and Midori64 [31].

This section summarizes the original formulation of the differential attack, with the notation showed in Fig. 4.

Let S_{EX} and S^*_{EX} be two outputs from the expansion function at round X, S_{IX} and S^*_{IX} the following two inputs to the S-Box $S(\cdot)$, and $S_{OX} = S(S_{IX})$ and $S^*_{OX} = S(S^*_{IX})$ the resulting outputs from the S-Box. The differences between S-Box inputs and outputs are obtained through the bitwise xor and are indicated as follows:

$$\begin{cases} S'_{IX} = S_{IX} \oplus S^*_{IX}, \\ S'_{OX} = S_{OX} \oplus S^*_{OX}. \end{cases} \tag{3}$$

The vulnerability exploited by the differential attack is that the probability distribution of the difference between two outputs, conditioned by the difference of the two corresponding input, i.e., $p(S'_{OX}|S'_{IX})$, is not uniform. This characteristic makes S-Boxes weak from a *dynamic* point of view, according to analysis proposed in [18].

Let's consider N pairs of output from the expansion function characterized by the same difference. As showed in Fig. 4, the relationship between each pair of outputs from the expansion function, the subkey, and the S-Box inputs is expressed by the following equations:

$$\begin{cases} S_{IX} = S_{EX} \oplus S_{KX}, \\ S^*_{IX} = S^*_{EX} \oplus S_{KX}. \end{cases} \tag{4}$$

Consequently, the difference between S_{IX} and S^*_{IX} is equal to the difference between S_{EX} and S^*_{EX}:

$$S'_{IX} = S_{IX} \oplus S^*_{IX} = S_{EX} \oplus S^*_{EX}. \tag{5}$$

Thus, given the knowledge of the expanded pairs (S_{EX}, S^*_{EX}), it is also known the difference between S-Box inputs, i.e., S'_{IX}, without knowing separate values. This knowledge does not allow to foresee the difference between S-Box outputs. Indeed, due to the non-linear behavior of S-Boxes is not obvious that two input pairs with the same difference produce the same output difference; on the contrary many values for the output difference are possible.

The critical point is that only some output differences are possible starting from a given input difference, and the probability distribution of these values is not uniform. For each pair (S_{EX}, S^*_{EX}), it is possible to observe the following output pair (S_{OX}, S^*_{OX}), and to compute the differences between inputs and between outputs, i.e., S'_{IX} and S'_{OX}, according to Eq. 3 and 4. Moreover, it is possible to select the set of possible keys which can produce the observed differences, by exploiting the equation $S_{KX} = S_{EX} \oplus S_{IX}$. Thus, each pair (S_{EX}, S^*_{EX}) produces a set of candidate keys, and the true secret key belongs to the intersection of these sets. Consequently, it is necessary to repeat this evaluation until such intersection is a singleton.

The logic behind the differential cryptanalysis attack can be described through the following simplified pseudocode:

```
$K \leftarrow \{k : k \in \mathbb{Z}_2^{56}\}$
forEach $(S_{EX}, S^*_{EX}) \in \mathbb{Z}_2^{48} \times \mathbb{Z}_2^{48}$
    Produce the corresponding pair $(S_{OX}, S^*_{OX})$
    $K' \leftarrow \emptyset$
    forEach $(S_{IX}, S^*_{IX}) \in \mathbb{Z}_2^{48} \times \mathbb{Z}_2^{48}$, such that
            $S_{IX} \oplus S^*_{IX} = S_{EX} \oplus S^*_{EX}$
        If $S(S_{IX}) \oplus S(S^*_{IX}) = S_{OX} \oplus S^*_{OX}$
            $S_{KX} = S_{IX} \oplus S_{EX}$
            $S^*_{KX} = S^*_{IX} \oplus S^*_{EX}$
            $K' = K' \cup \{S_{KX}, S^*_{KX}\}$
    $K = K \cap K'$
    If $|K| = 1$ then return $k \in K$
```

V. BAYESIAN NETWORKS MODELS

Bayesian networks (BN) [32] are a graph-based formalism capable of expressing probabilistic cause/effect relationships between random variables. Such framework is adopted in machine learning for performing probabilistic inference. In this work, we model through BNs the statistical dependence driven by the secret key between input differences and output differences, as found in [10], and we exploited it to determine the secret key.

In the graphic model adopted by BNs, nodes represent random variables and directed links represent the cause/effect dependence between two nodes. BNs allow to represent the joint probability distribution of several variables through a set of conditioned probability distributions, each associated to a link, and a set of a priori probability distributions, for nodes without antecedents.

In this section, we will present the BNs which model a single S-Box, the Feistel function and the whole DES, and then we will present the algorithms for attacking such elements through exact inference, and analyze their computation complexity. We will prove that the exact inference for attacking the whole DES has a high computation cost, and consequently we will propose an algorithm based on approximate inference.

A. SINGLE S-BOX ATTACK

For the construction of the BN for attacking the S-Box, the original notation reported in [10] is adopted.

It is useful to recall that the S-Box consists in a set of eight S-Boxes, indicated as $Si(\cdot)$ with $(i = 1, \ldots, 8)$, each of which accepts 6 bits as input and produces 4 bits as output. So, the input to a S-Box can be considered divided into eight

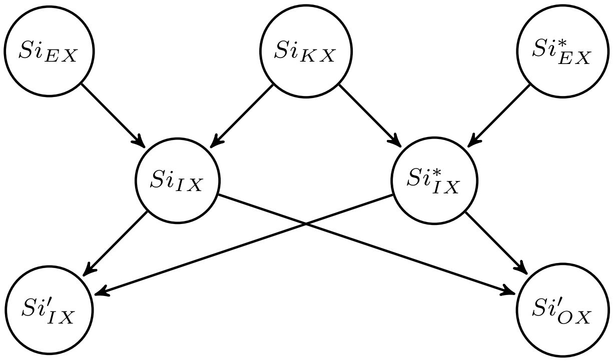

FIGURE 5. **Bayesian network for the inference on a single S-Box (*SBox-BN*).**

6-bit blocks. According to the adopted notation, (Si_{EX}, Si^*_{EX}) indicate the two i-th 6-bit blocks of two different outputs from the expansion function, Si_{KX} indicates the i-th 6-bit block of the subkey, (Si_{IX}, Si^*_{IX}) represent the two inputs to the i-th S-Box $Si(\cdot)$, Si'_{IX} represents the difference between the two inputs to the i-th S-Box, and finally Si'_{OX} indicates the difference between the two 4-bit outputs from the i-th S-Box.

The probabilistic inference exploits the known value of some random variables, named *evidence*, and infer the probability distribution of a set of unknown random variables, named *target nodes*.

Since the differential cryptanalysis exploits a chosen plaintext attack, i.e. a circumstance where the adversary is capable to trigger the encryption of arbitrary messages and to observe the corresponding plaintext-ciphertext pair, the set $(Si_{EX}, Si^*_{EX}, Si'_{IX}, Si'_{OX})$ constitutes the evidence and the key blocks Si_{KX} represent the target nodes.

In order to build the BN we complied with the following assumptions:

- The Si_{KX}, Si_{EX} and Si^*_{EX} variables are not influenced by other random variables, thus they are represented as nodes without antecedents; their a priori probability distribution is considered as uniform.
- The input to the i-th S-Box, Si_{IX}, depends only on Si_{EX} and Si_{KX}, according to Eq. 4, which are the sole parents of the Si_{IX} node (analogously for Si^*_{IX}).
- The input difference for the i-th S-Box, Si'_{IX}, depends only on these two inputs, according to the first part of Eq. 3, thus the only two parents of the Si'_{IX} node are the Si_{IX} and Si^*_{IX} nodes.
- Since the outputs of the S-Box are relevant only when considered in their difference, it is not necessary to represent them explicitly as separated nodes; instead, it is convenient to adopt a single node for representing their difference, Si'_{OX}.
- The output difference Si'_{OX} depends on the two S-Box inputs Si_{IX} and Si^*_{IX}, according to the following equation:

$$Si'_{OX} = Si_{OX} \oplus Si^*_{OX} = S(Si_{IX}) \oplus S(Si^*_{IX}), \qquad (6)$$

thus the Si_{IX} and Si^*_{IX} nodes are the only parents of the Si'_{OX} node.

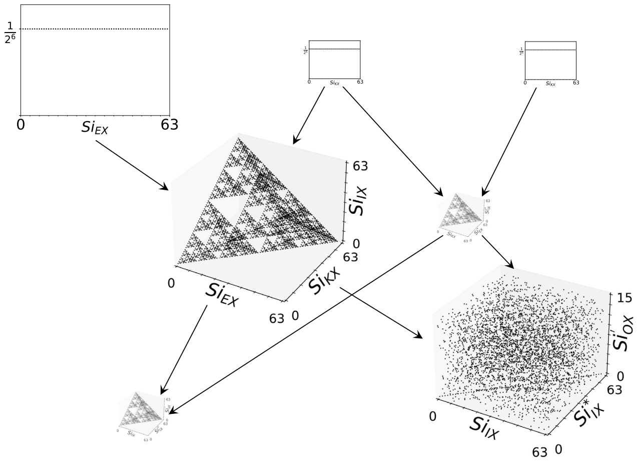

FIGURE 6. **Flow of the probability distributions through the SBox-BN. The three most representative plots are highlighted.**

The resulting BN, named *SBox-BN*, is showed in Fig. 5. The full definition of the BN requires the formalization of (i) the a priori probability distributions for nodes without parents and (ii) the conditioned probability distributions for other nodes.

Let us represent as $\delta_n(X)$ the Kronecker delta applied to a n-bit string, taking value one if and only if all bits of its argument are equal to zero. Then, the probability distributions of the *SBox-BN* are expressed as follows:

- $p(Si_{EX} = si_{ex}) = p(Si^*_{EX} = si^*_{ex}) =$
  $= p(Si_{KX} = si_{kx}) = \frac{1}{2^6}, \forall si_{ex}, si^*_{ex}, si_{kx} \in \mathbb{Z}_2^6$,
  because of the hypothesis of uniform distribution;
- $p(Si_{IX} = si_{ix} | Si_{EX} = si_{ex}, Si_{KX} = si_{kx}) =$
  $= \delta_6(si_{ix} \oplus si_{ex} \oplus si_{kx}), \ \forall si_{ix}, si_{ex}, si_{kx} \in \mathbb{Z}_2^6$,
  because of Eq. 4;
- $p(Si^*_{IX} = si^*_{ix} | Si^*_{EX} = si^*_{ex}, Si_{KX} = si_{kx}) =$
  $= \delta_6(si^*_{ix} \oplus si^*_{ex} \oplus si_{kx}), \ \forall si^*_{ix}, si^*_{ex}, si_{kx} \in \mathbb{Z}_2^6$,
  because of Eq. 4;
- $p(Si'_{IX} = si'_{ix} | Si_{IX} = si_{ix}, Si^*_{IX} = si^*_{ix}) =$
  $= \delta_6(si'_{ix} \oplus si_{ix} \oplus si^*_{ix}), \forall si'_{ix}, si_{ix}, si^*_{ix} \in \mathbb{Z}_2^6$,
  because of Eq. 3;
- $p(Si'_{OX} = si'_{ox} | Si_{IX} = si_{ix}, Si^*_{IX} = si^*_{ix}) =$
  $= \delta_4(si'_{ox} \oplus Si(si_{ix}) \oplus Si(si^*_{ix}))$,
  $\forall si'_{ox} \in \mathbb{Z}_2^4$ and $si_{ix}, si^*_{ix} \in \mathbb{Z}_2^6$, because of Eq. 3.

The flow of the probability distributions through the Bayesian Network depicted in Fig. 5 is summarized in Fig. 6, where the three plots show the most significant distributions within the *SBox-BN*. The probabilities of all nodes at level 0, e.g., Si_{EX}, are uniformly distributed (see the plot in the upper left corner); that is all outcomes are equally likely with a probability of $1/2^6$. The two nodes at level 1, as well as their child Si'_{IX}, are characterized by a distribution in which the probability of most configurations is zero, while the remaining possible hypotheses have constant probability values. The 3D-plots in Fig. 6 represent this probability; the axes refer to the variables involved in the probability distribution equation, while the points indicate where the probability assumes nonzero values. By observing the plot for Si'_{IX}, it can be noticed that only a subset of keys, characterized by an extremely regular pattern, is retained over all possible combinations. The bottom right plot shows the probability distribution of the node Si'_{OX}, which is characterized by the lack of a regular patterns because of the non-linearity introduced by the S-Box.

Under such BN model, given the two outputs si_{ex} and si^*_{ex} from the expansion function and the corresponding output difference si'_{ox} from the S-Box at the round X, the most probable secret key corresponds to the greatest conditioned probability among keys that produce $si'_{ix} = si_{ex} \oplus si^*_{ex}$ as input difference and si'_{ox} as output difference, as follows:

$$\hat{k_x} = \underset{k_x \in \mathbb{Z}_2^6}{argmax}\ p(Si_{KX} = k_x | Si_{EX} = si_{ex}, Si^*_{EX} = si^*_{ex}, \\ Si'_{IX} = si'_{ix}, Si'_{OX} = si'_{ox}). \quad (7)$$

By applying rules for manipulating probability expressions in BNs, it is possible to obtain the explicit formulation of such conditioned probability:

$$p(Si_{KX} | Si_{EX}, Si^*_{EX}, Si'_{IX}, Si'_{OX}) \\ = \eta_1 \sum_{\sigma_1} p(Si'_{IX} | Si_{IX}, Si^*_{IX}) p(Si'_{OX} | Si_{IX}, Si^*_{IX}), \quad (8)$$

where η_1 is a normalization factor which makes 1 the sum of all terms of the probability distribution, and σ_1 is the set of all (Si_{IX}, Si^*_{IX}) pairs obtained through the XOR of the possible secret key with the given input evidence:

$$\sigma_1 = \{(Si_{IX}, Si^*_{IX}) : \quad Si_{IX} = Si_{EX} \oplus Si_{KX} \text{ and} \\ Si^*_{IX} = Si^*_{EX} \oplus Si_{KX}\}. \quad (9)$$

It is worth noting that, since Si_{IX} and Si^*_{IX} are restricted to a single value, the sum in Eq. 8 corresponds to a single value, as expressed by the following equation:

$$\begin{aligned} p(Si_{KX} | Si_{EX}, Si^*_{EX}, Si'_{IX}, Si'_{OX}) &= \\ = \eta_1 p(Si'_{IX} | Si_{IX} &= Si_{EX} \oplus Si_{KX}, \\ Si^*_{IX} &= Si^*_{EX} \oplus Si_{KX}) \times \\ p(Si'_{OX} | Si_{IX} &= Si_{EX} \oplus Si_{KX}, \\ Si^*_{IX} &= Si^*_{EX} \oplus Si_{KX}). \end{aligned} \quad (10)$$

For the sake of brevity, we omitted the detailed proof, that nevertheless can be found in [33].

In order to narrow down the set of possible keys, it is possible to evaluate a non-normalized version of Eq. 10, by ignoring the normalizing factor η_1. Indeed, the probability distributions describing the *SBox-BN* are expressed through the Kronecker delta; thus, Eq. 10 can provide only two values: 0 for all keys that have been excluded, and a constant value η_1 for all keys that are still possible. Such a value can be determined by imposing that the sum of all the residual probabilities is equal to 1. However, since the purpose of Eq. 10 is merely to identify the residual set of keys, the computing of a specific value for η_1 is irrelevant.

Algorithm 1 - `prob_key_SBox_attack` - Algorithm for Computing the Probability That a Key Block Is Correct by Attacking the *i*-Th S-Box

```
Data: i: the index of the selected S-Box Ψ: a set of multiple
      evidences $\Theta = \{si_{ex}, si^*_{ex}, si'_{ix}, si'_{ox}\}$;
Result: p: the array of $2^6$ values, representing the
        non-normalized probability distribution over the set
        of possible key blocks.
begin
    $p \leftarrow$ new array $[2^6]$;
    for $k_{ix} = 0 : (2^6 - 1)$ do
        $p[k_{ix}] = 1$;
    for all $\Theta = \{si_{ex}, si^*_{ex}, si'_{ix}, si'_{ox}\} \in \Psi$ do
        for $k_{ix} = 0 : (2^6 - 1)$ do
            $si_{ix} = si_{ex} \oplus k_{ix}$;
            $si^*_{ix} = si^*_{ex} \oplus k_{ix}$;
            $p[k_{ix}] = p[k_{ix}] \times \delta_4(si'_{ox} \oplus Si(si_{ix}) \oplus Si(si^*_{ix})) \times$
                                $\times \delta_6(si'_{ix} \oplus si_{ix} \oplus si^*_{ix})$;
return p;
```

The sets of possible keys obtained by attacking a S-Box with two different evidence sets may be different. Since the true secret key belongs to each of these sets, their intersection is never void. With a sufficient quantity of data, by performing multiple attacks with different evidences, the repeated intersection of the obtained key sets produces the singleton containing only the secret key.

The assumption of the independence of the evidence sets allows to express the probability distribution of the secret key conditioned by all the evidence sets as the product of the probability conditioned by each single evidence set:

$$p(Si_{KX} | \Psi) = \eta_2 \prod_{j=1}^{n} p(Si_{KX} | \Theta_j), \quad (11)$$

where η_2 is a normalization factor and Ψ is the set of multiple evidences:

$$\begin{aligned} \Psi &= \{\Theta_1, \ldots, \Theta_n\} \\ &= \left\{ \{Si_{EX}, Si^*_{EX}, Si'_{IX}, Si'_{OX}\}_1, \right. \\ &\quad \left. \ldots, \{Si_{EX}, Si^*_{EX}, Si'_{IX}, Si'_{OX}\}_n \right\}. \end{aligned} \quad (12)$$

In order to find the most probable key, it is possible to evaluate the not normalized version of Eq. 11 and Eq. 8 as described by the Algorithm 1.

B. FEISTEL FUNCTION ATTACK

The same approach of the previous section can be generalized in order to attack a single instance of the Feistel function, by analyzing its output (named Y_X in Fig. 3). Under the hypothesis of chosen plaintext attack, it is possible to select a pair of inputs to the Feistel function, $Z_X, Z^*_X \in \mathbb{Z}_2^{32}$, and then

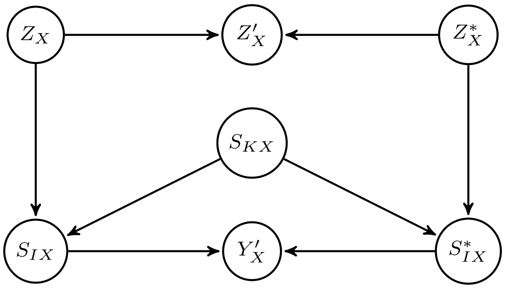

FIGURE 7. Bayesian network for the inference on the Feistel function (*Feistel-BN*).

observe the difference between the corresponding outputs, Y'_X, obtained according to the following equation:

$$Y'_X = P(S(E(Z_X) \oplus S_{KX})) \oplus P(S(E(Z^*_X) \oplus S_{KX})), \quad (13)$$

where P, S and E are respectively the permutation function, the substitution performed by the S-box, and the expansion function. By observing that $S_{IX} = E(Z_X) \oplus S_{KX}$ and $S^*_{IX} = E(Z^*_X) \oplus S_{KX}$, and by exploiting the linearity property of the permutation function, it is possible to obtain the following system of equations:

$$\begin{cases} Z'_X = Z_X \oplus Z^*_X, \\ S_{IX} = E(Z_X) \oplus S_{KX}, \\ S^*_{IX} = E(Z^*_X) \oplus S_{KX}, \\ Y'_X = P(S(S_{IX}) \oplus S(S^*_{IX})). \end{cases} \quad (14)$$

Each variable in such system can be considered as a random variable, and their relationships can be represented through the BN showed in Fig. 7, named *Feistel-BN*. Its probability distributions are expressed as follows:

- $p(Z_X = z_x) = p(Z^*_X = z^*_x) = \frac{1}{2^{32}}$, $\forall z_x, z^*_x \in \mathbb{Z}_2^{32}$, because of the hypothesis of uniform distribution;
- $p(S_{KX} = s_{kx}) = \frac{1}{2^{48}}$, $\forall s_{kx} \in \mathbb{Z}_2^{48}$, because of the hypothesis of uniform distribution;
- $p(Z'_X = z'_x | Z_X = z_x, Z^*_X = z^*_x) = \delta_{32}(z'_x \oplus z_x \oplus z^*_x)$, $\forall z'_x, z_x, z^*_x \in \mathbb{Z}_2^{32}$, because of the first part of Eq. 14;
- $p(S_{IX} = s_{ix} | Z_X = z_x, S_{KX} = s_{kx}) =$ $= \delta_{48}(s_{ix} \oplus E(z_x) \oplus s_{kx})$, $\forall z_x \in \mathbb{Z}_2^{32}$ and $\forall s_{ix}, s_{kx} \in \mathbb{Z}_2^{48}$, because of the second part of Eq. 14;
- $p(S^*_{IX} = s^*_{ix} | Z^*_X = z^*_x, S_{KX} = s_{kx}) =$ $= \delta_{48}(s^*_{ix} \oplus E(z^*_x) \oplus s_{kx})$, $\forall z^*_x \in \mathbb{Z}_2^{32}$ and $\forall s^*_{ix}, s_{kx} \in \mathbb{Z}_2^{48}$, because of the third part of Eq. 14;
- $p(Y'_X = y'_x | S_{IX} = s_{ix}, S^*_{IX} = s^*_{ix}) =$ $= \delta_{32}(y'_x \oplus P(S(s_{ix}) \oplus S(s^*_{ix})))$, $\forall s_{ix}, s^*_{ix} \in \mathbb{Z}_2^{48}$ and $\forall y'_x \in \mathbb{Z}_2^{32}$, because of the fourth part of Eq. 14;

The goal of the attack on the Feistel function is to find the most probable set of keys, given the known evidence, obtained by maximizing the following likelihood:

$$p(S_{KX} | Z_X, Z^*_X, Y'_X). \quad (15)$$

Algorithm 2 - `prob_key_attack_Feistel` - Algorithm for Computing the Not Normalized Probability Distribution Over Possible Keys by Attacking the Feistel Function

```
Data: Ω: a set of multiple evidences $\Phi = \{z_x, z^*_x, y'_x\}$
Result: p: the array of $2^{48}$ values, representing the
        non-normalized probability distribution over the set
        of possible keys.
begin
    //separate all the evidences in blocks for each S-Boxes
    for all $\Phi = \{z_x, z^*_x, y'_x\} \in \Omega$ do
        split Φ in $\Theta[i] = \{si_{ex}, si^*_{ex}, si'_{ix}, si'_{ox}\}$, with
          $i = 1:8$;
    for $i = 1:8$ do
        Let $\Psi[i]$ the set of all the evidences $\Theta[i]$;

    //compute the probability distribution for key blocks
    $p_{SBox} \leftarrow$ new array $[8][2^6]$;
    for $i = 1:8$ do
        $p_{SBox}[i][*] =$ prob_key_SBox_attack
          $(i, \Psi[i])$;
    //compute the probability distribution for keys
    $p \leftarrow$ new array $[2^{48}]$;
    for $k_x = 0 : (2^{48} - 1)$ do
        $p[k_x] = 1$;
        for $i = 1:8$ do
            //select the probability of the i-th 6-bit
            //key block
            $keyBlock = selectBits(k_x, (i*6), (i*6+5))$;
            $p[k_x] = p[k_x] \times p_{SBox}[i][keyBlock]$;
    return p;
```

Albeit the construction of the probability distribution over a 48-bit key, by expanding Eq. 15, requires 2^{48} steps, it is possible to reduce the computational complexity by exploiting the linearity of the $P(\cdot)$ and $E(\cdot)$ functions. Let us recall that the XOR between the output of the expansion function, $E(\cdot)$, and the secret key, is the concatenation of the inputs to eight S-boxes, and that the input to the permutation function $P(\cdot)$ is the concatenation of the outputs from the eight S-boxes, as expressed by the following equations:

$$\begin{cases} E(Z_X) \oplus S_{KX} = S1_{IX} || \dots || S8_{IX}, \\ E(Z^*_X) \oplus S_{KX} = S1^*_{IX} || \dots || S8^*_{IX}, \\ P^{-1}(Y_X) = S1_{OX} || \dots || S8_{OX}, \\ P^{-1}(Y^*_X) = S1^*_{OX} || \dots || S8^*_{OX}. \end{cases} \quad (16)$$

Then, the Feistel function can be violated by attacking each single S-Box and then by obtaining the full 48-bit key by concatenating the partial results:

$$S_{KX} = S1_{KX} || \dots || S8_{KX}. \quad (17)$$

Thus, the actual computational cost for attacking the whole Feistel function is eight times the cost for attacking a single S-Box, since the following equation holds:

$$p\left(S_{KX} \mid Z_X, Z^*_X, Y'_X\right) = \prod_{i=1}^{8} p\left(Si_{KX} \mid Si_{EX}, Si^*_{EX}, Si'_{OX}\right). \quad (18)$$

The Algorithm 2 describes how to perform the attack. Its computational cost is dominated by the evaluation of the probability distribution for key blocks. Namely, the other components, i.e., the separation of the evidences in blocks and the composition of the whole probability distribution, may be easily optimized, although in the pseudocode they are described in an extended form for the sake of readability.

C. DES ATTACK

In the following we describe the BN which models the attack on the whole DES. We show that, differently from the attack on the Feistel function, it is not affordable to attack the complete DES through exact inference since the computational cost grows exponentially. We propose, hence, an algorithm for attacking DES through approximate inference.

In the following description we neglect the initial and final permutations, since they do not affect the probabilistic analysis. Let P and P^* be two plain texts input to DES, P' be their difference, and (L', R') the left and right parts of P', each constituted by 32 bits. Let us indicate the difference between the two outputs from DES as T', and (l', r') its left and right parts. Moreover, let us assume that the two plain texts are independently chosen.

The relationships among variables involved in the first round of DES are described by the following equations:

$$\begin{cases} Z_1' = Z_1 \oplus Z_1^*, \\ Y_1' = F(Z_1, S_{K1}) \oplus F(Z_1 \oplus Z_1', S_{K1}). \end{cases} \tag{19}$$

The difference between the inputs to the second round can be obtained by considering the variables involved in the first round, as follows:

$$\begin{aligned} Z_2' &= (Y_1 \oplus L) \oplus (Y_1^* \oplus L^*) = (Y_1 \oplus Y_1^*) \oplus (L \oplus L^*) \\ &= Y_1' \oplus L'. \end{aligned} \tag{20}$$

The iteration of such procedure leads to the formulation of the following system of equations, that expresses relationships among the variables involved in all the rounds of DES:

$$\begin{cases} R' = R \oplus R^*, \\ L' = L \oplus L^*, \\ Z_1' = R', \\ Y_1' = F(Z_1, S_{K1}) \oplus F(Z_1 \oplus Z_1', S_{K1}), \\ Z_2' = L' \oplus Y_1', \\ \vdots \\ Z_X' = Y_{X-1}' \oplus Z_{X-2}', \\ Y_X' = F(Z_X, S_{KX}) \oplus F(Z_X \oplus Z_X', S_{KX}), \\ \vdots \\ l' = Z_n', \\ Y_n' = F(Z_n, S_{Kn}) \oplus F(Z_n \oplus Z_n', S_{Kn}), \\ r' = Y_n' \oplus Z_{n-1}'. \end{cases} \tag{21}$$

These relationships are graphically represented by the Bayesian Network, named *DES-BN*, showed in Fig. 8. It is worth noting that the structure of the *DES-BN* is based on the simplifying assumption that subkeys are mutually independent, as also proposed in [10], since such assumption allows to simplify the evaluation of the BN conditioned probabilities.

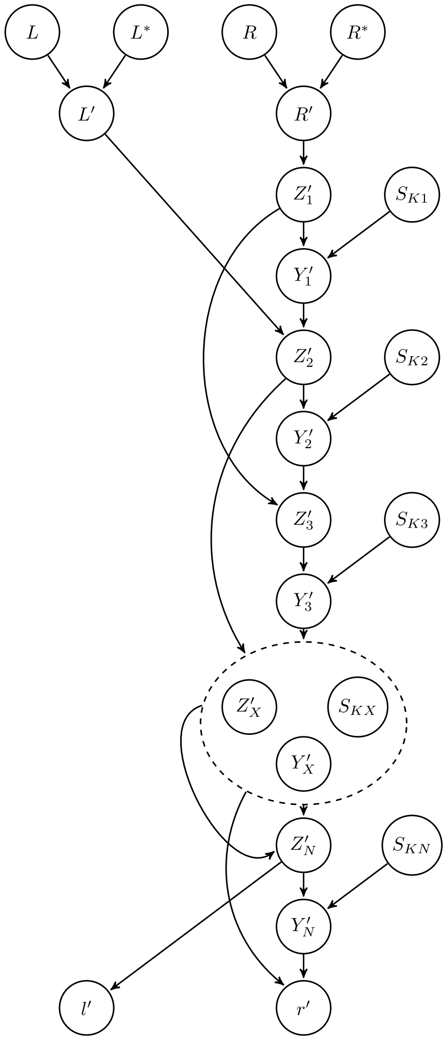

FIGURE 8. Bayesian network for inference over DES composed by *N* rounds (*DES-BN*).

The goal of the attack on the whole DES, given a single evidence set $\Gamma = (P, P^*, T, T^*)$, is to find the set of keys that maximizes the following likelihood:

$$p(S_{K1}, S_{K2}, \ldots, S_{KN}|\Gamma), \tag{22}$$

where $N = 16$ is the number of rounds.

It is possible to prove that such likelihood can be expressed as follows:

$$\begin{aligned} p\left(S_{K1}, \ldots, S_{KN} \mid \Gamma\right) &= \eta_6 p\left(R\right) p\left(R^*\right) p\left(L\right) p\left(L^*\right) \\ &\times p\left(R' \mid R, R^*\right) p\left(L' \mid L, L^*\right) \times \prod\nolimits_{X=1}^{N} p\left(S_{KX}\right) \\ &\times \sum\nolimits_{\substack{Z_1', \ldots, Z_N' \\ Y_1', \ldots, Y_N'}} \Big[p\left(Z_1' \mid R'\right) \times p\left(Z_2' \mid Y_1', L'\right) \\ &\times \prod\nolimits_{X=3}^{N} p\left(Z_X' \mid Y_{X-1}', Z_{X-2}'\right) \\ &\times p\left(l' \mid Z_N'\right) p\left(r' \mid Z_{N-1}', Y_N'\right) \\ &\times \prod\nolimits_{X=1}^{N} p\left(Y_X' \mid Z_X', S_{KX}\right) \Big]. \end{aligned} \tag{23}$$

Algorithm 3 - `prob_key_attack_DES` - Algorithm for Computing the Approximate Probability Distribution Over Possible Subkeys by Attacking DES With Multiple Evidences

```
Data: $N$: the number of rounds composing DES (N=16);
      $\Upsilon$: a set of multiple evidence $\Gamma = (P, P^*, T, T^*)$; $M$:
      the number of samples used by the sampling
      algorithm;
Result: $S_K$: the $N$-size set of subkeys;
begin
    $S_K \leftarrow$ new array $[N]$; // Subkeys set;

    // For each round from N down to 3rd round
    for $X = N : 3$ do
        $\Omega = \emptyset$ // Set of multiple evidences for attacking the
          Feistel function;

        // For each evidence set, build data to attack the
          Feistel function:
        for all $\Gamma = (P, P^*, T, T^*) \in \Upsilon$ do
            $(l, r)$ = left and right parts of $T$;
            $(l^*, r^*)$ = left and right parts of $T^*$;
            $(l', r')$ = left and right parts of
                        $T' = T \oplus T^*$;
            $Z_X = l$;
            $Z_X^* = l^*$;
            $Z'_{X-1}$ = argmax (histogram
                 (samples_$Z'_X(P, P^*, \text{M}, \text{X-1})$));
            $Y'_X = r' \oplus Z'_{X-1}$;
            $\Omega = \Omega \cup (Z_X, Z_X^*, Y'_X)$;
        // Attack the Feistel function:
        $p$ =prob_key_attack_Feistel($\Omega$);
        $S_K[X]$ = argmax $(p)$;

        // Update the evidence sets:
        for all $\Gamma_i = (P, P^*, T, T^*) \in \Upsilon$ do
            $temp_l = l$;
            $temp_{l^*} = l^*$;
            $l = r \oplus F(l, S_K[X])$;
            $l^* = r \oplus F(l^*, S_K[X])$;
            $r = temp_l$;
            $r^* = temp_{l^*}$;
            udpate $\Gamma \leftarrow (P, P^*, (l, r), (l^*, r^*))$;

    Break 3-round DES through exact inference;

    return $S_K$;
```

The research of the optimal key by exploiting the Eq. 23, through a backward exact inference process requires an high computational cost, that makes infeasible such an approach.

Instead, it is possible to exploit the forward inference in order to estimate the most probable difference propagation through different rounds, and then exploit a statistical sampling technique, as described in [32], to estimate the subkey for each round.

In particular, for the last round N, the following relationships among variables hold:

$$\begin{cases} l' = Z'_N, \\ r' = Y'_N \oplus Z'_{N-1}, \end{cases} \Rightarrow \begin{cases} l' = Z'_N, \\ Y'_N = r' \oplus Z'_{N-1}. \end{cases} \tag{24}$$

Algorithm 4 - `sample_`Z'_X - Algorithm for Drawing a Set of Samples for the Z'_X Variables of All DES Rounds, Given a Single Evidence

```
Data: $(P, P^*)$: the evidence of a pair of plaintext; $M$: the
      number of samples used by the sampling algorithm;
      $X$: the round containing the $Z'_X$ to be sampled;
Result: $Z'_X$: the set of samples for the $Z'_X$ variable;
begin
    $Z \leftarrow$ new array $[X+1][M]$;
    $Z' \leftarrow$new array $[X+1][M]$;

    // Round 1 variables are part of the given evidence
    $(L, R)$ = left and right parts of $P$;
    $(L^*, R^*)$ = left and right parts of $P^*$;
    $(L', R')$ = left and right parts of $P' = P \oplus P^*$;
    set all values $Z[1][*]$ with $R$;
    set all values $Z'[1][*]$ with $R'$;
    set all values $Z[0][*]$ with $L$;
    set all values $Z'[0][*]$ with $L'$;

    // For each round from 2 to X-1
    for $x$= 2 : X-1 do
        for $m$ = 1 : M do
            // draw M samples
            $S_{K(x-1)}$ = random(0: $2^{48} - 1$);
            $Y'_{(x-1)} = F(Z[x-1][m], S_{K(x-1)}) \oplus$
              $F(Z[x-1][m] \oplus Z'[x-1][m], S_{K(x-1)})$;
            $Z'[x][m] = Y'_{(x-1)} \oplus Z'[x-2][m]$;
    return $Z'[X]$;
```

If Z'_{N-1} were known, the best way to obtain the subkey S_{KN} should be to compute the value of Y'_N through Eq. 24, and then use the attack on the Feistel function of the last round, by using Z'_N and Y'_N as input. Unfortunately, such piece of information is not available, and its exact inference through the BN would be computationally too expensive. We propose to sample the *DES-BN* in order to estimate the most probable value of Z'_{N-1} by exploiting the structure of the *DES-BN* and the only exact information available, i.e., the given evidence. Given the estimated value of Z'_{N-1} it is possible to backwards iterate the same procedure for the remaining $N-1$ rounds until the construction of a probability distribution for all subkeys. Such attack is described by Algorithm 3. At each round, multiple evidences are exploited to attack the Feistel function, in order to find the most probable subkey.

The algorithm to sample an objective node consists in sorting all nodes of the Bayesian Network according to its topology, and then sampling the probability distribution of all nodes that precede the objective node and finally sampling the objective node. In order to estimate the most probable value of Z'_X, in a given round X, our algorithm starts from the exact knowledge of P and P^* and follows all causal links in the path to the Z'_X, drawing a random value for each unknown parent node. This procedure allows to obtain a possible value for Z'_X. The iteration of such procedure produces a set of samples of Z'_X, and by analyzing the resulting histogram it is possible to select the most frequent

TABLE 1. Number of plaintext-ciphertex pairs (*Texts*) and time-to-succeed (*TTS*) required to attack 8 S-Boxes with different random keys by means of the algorithm `prob_key_SBox_attack`.

Key	Texts	TTS (ms)
30	3	0.15
1A	3	0.15
41	4	0.29
5F	3	0.19
5D	4	0.23
1A	3	0.13
29	3	0.15
45	3	0.14
48	3	0.13
15	4	0.2

(a) Test S-Box 1

Key	Texts	TTS (ms)
31	3	0.17
32	3	0.14
4B	2	0.06
33	3	0.14
27	3	0.12
60	3	0.13
22	3	0.14
4C	4	0.19
29	3	0.15
50	3	0.12

(b) Test S-Box 2

Key	Texts	TTS (ms)
59	3	0.14
51	4	0.18
61	4	0.18
31	3	0.15
5E	3	0.14
50	3	0.12
37	4	0.18
26	3	0.12
2D	3	0.13
18	3	0.12

(c) Test S-Box 3

Key	Texts	TTS (ms)
44	2	0.07
25	3	0.14
56	3	0.16
52	3	0.14
15	3	0.13
4F	2	0.11
45	3	0.16
14	2	0.11
3B	3	0.14
20	3	0.16

(d) Test S-Box 4

Key	Texts	TTS (ms)
56	3	0.18
1E	3	0.14
31	3	0.14
50	3	0.14
1B	3	0.14
5C	3	0.14
5C	2	0.07
40	3	0.15
3E	3	0.17
57	2	0.08

(e) Test S-Box 5

Key	Texts	TTS (ms)
60	3	0.13
49	4	0.18
5A	3	0.13
49	2	0.07
14	3	0.14
2F	2	0.06
15	3	0.16
29	3	0.16
29	3	0.14
30	3	0.14

(f) Test S-Box 6

Key	Texts	TTS (ms)
47	3	0.15
30	3	0.13
4D	3	0.11
3A	3	0.12
1B	3	0.12
43	3	0.12
41	3	0.14
2C	3	0.14
1B	2	0.08
2E	3	0.12

(g) Test S-Box 7

Key	Texts	TTS (ms)
58	2	0.07
58	3	0.12
18	3	0.12
2D	3	0.15
43	3	0.14
24	3	0.13
2E	3	0.12
63	3	0.13
5C	3	0.16
22	3	0.15

(h) Test S-Box 8

sample. With an adequate number of samples the histogram approximates the probability distribution of Z'_X, thus the most frequent sample can be considered an approximation of the most probable value. This sampling strategy is described in Algorithm 4.

VI. PERFORMANCE EVALUATION

A first assessment of the performance of the proposed approach concerned the evaluation of the complexity of the four algorithms it consists of.

The computational complexity of the algorithm to attack a single SBox (Algorithm 1) is $O_{SBox} = O(2^b|\Psi|)$, where $|\Psi|$ is the number of exploited evidences, and b is the number of bits composing the key block accepted as input by one of the eight S-Boxes, i.e., $b = 6$.

The evaluation of the probability distribution during the attack to the Feistel Function, according to Algorithm 2, has a complexity $O_{Feistel} = O(n_s * 2^b * |\Psi|)$, where $|\Psi|$ is the number of exploited evidences in the attack on a single S-Box, n_s is the number of S-Boxes, and b is the number of bits of the key block used by one of the eight S-Boxes, i.e., $n_s = 8$ and $b = 6$.

The complexity of the sampling procedure (Algorithm 4) depends on the number of samples required to obtain the convergence of the probability distribution, i.e, M, and on the round to be sampled, i.e., X. The upper bound of such complexity is determined by considering the last round, i.e., $X = N$, as in the following equation: $O_{sampling} = O(X*M) < O(N*M)$.

It is worth noting that the samples generated during the graph descent can be reused during the backtracking, thus obtaining a more efficient procedure than the expanded Algorithm 4.

The computational cost for attacking the whole DES (Algorithm 3) is expressed by the following equation:

$$\begin{aligned} O_{DES} &= O(N(|\Upsilon|O_{sampling} + O_{Feistel} + |\Upsilon|)) < \\ &= O(N|\Upsilon|(N*M + 2^b)), \end{aligned} \tag{25}$$

where $|\Upsilon|$ is the number of elements constituting the evidence set, M is the number of samples required by the sampling algorithm, N is the number of round of DES, and b si the number of input bits to a single S-Box. Since $b = 6$ and $N = 16$, it follows that 2^b and N^2 can be considered as constant. Consequently, the computational complexity can be expressed as follows:

$$O_{DES} = O(|\Upsilon|M). \tag{26}$$

Such result is coherent with the expected complexity of a chosen plaintext attack, which directly depends on the number of plaintext-ciphertext pairs.

Another set of experiments was run in order to find the number of plaintext-ciphertext pairs needed to attack each of the 8 S-boxes by means of the Algorithm 1. Tests were executed on a multi-core server equipped with 4 Intel Xeon 2.00 GHz by reporting the time-to-succeed (in milliseconds) when using 10 different random keys. Results, shown in Table 1, indicate that on average 3 plaintext-ciphertext pairs are needed to accomplish the attack on every S-box. It can be observed that the time-to-succeed (TTS) are in general very low, and no noticeable variations are evident as the random keys and the S-boxes vary. This aspect was further inspected by evaluating the average TTS (see Fig. 9) and the

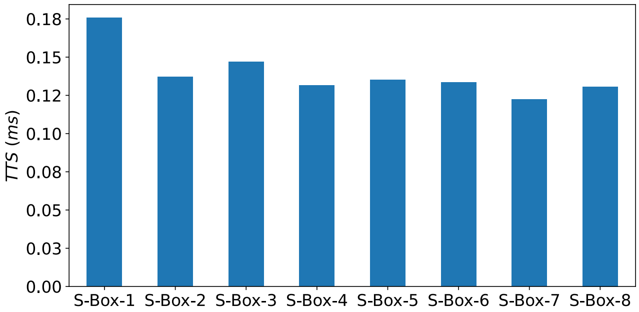

FIGURE 9. Average time-to-succeed (ms) for each S-box attack with the `prob_key_SBox_attack`.

TABLE 2. Results of the attacks conducted against four variations of DES reduced to three, four, five and six rounds.

Rounds	Time (s)	Texts	Samples
3	0.063	3	0
4	0.128	10	39.7
5	0.505	100	27.8
6	2.019	250	44.72

corresponding variance values, which are about 10^{-3} *ms* for each experiment.

Finally, we extended the experimental evaluation to four distinct versions of DES reduced to three, four, five, and six rounds, respectively. Results we obtained (see Table 2) are comparable with the performance of the original differential crypyanalysis [10], [34]. In particular, for each variation of DES, we considered the average execution time (*Time*), the number of chosen plaintext-ciphertext pairs (*Texts*), and the number of required samples obtained by the `sample_`Z'_X algorithm (*Samples*). It is worth noticing that changing the number of available plaintext-ciphertext pairs significantly impacts on the number of samples required to accomplish the attack. The values reported in Table 2 are those that minimize the computational complexity of the whole attack (Eq. 25).

This preliminary assessment leads us to conclude that the number of plaintext-ciphertext pairs required to attack a full 16-round DES is not lower than the threshold of 2^{47} that exists for the standard differential attack approach.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new formulation of differential cryptanalysis through Bayesian networks, a framework for performing probabilistic inference that is widely adopted in the field of machine learning. Exploiting such model we designed an algorithm for attacking DES through approximate inference on such Bayesian Network model. Our preliminary experimental evaluation, performed on a version of DES with a reduced number of rounds, showed that the proposed method is equivalent to the original differential cryptanalysis, with respect to required input data and convergence time. Beyond its effectiveness, the computational aspect represents the main limitation of the approach. Indeed, the Bayesian framework, in its current form, does not perform significantly better than other traditional cryptanalysis approaches. However, the formulation of the attack using Bayesian Networks gives several insights for improvement. To be more specific, we plan to evaluate more advanced forward sampling techniques, such as importance sampling, in order to verify the possibility to reduce the convergence time and to minimize the sample inputs. Furthermore, since multiple evidences are mutually independent, the reduction of the convergence time can be achieved by exploiting a massive parallel architecture. Finally, although the hypothesis of mutual independence of subkeys allows to reduce the computational cost, it introduces many contradictory hypothesis about subkeys of different rounds. In a future work we will investigate the introduction of a new BN model modeling the linear relationship among subkeys.

REFERENCES

[1] *Data Encryption Standard*, Standard FIPS-Pub.46, Nat. Bur. Standards, U.S. Dept. Commerce, Washington, DC, USA, 1997.

[2] L. Guangyong and L. Ruolan, "Analysis and design of graduation design management system based on DES encryption algorithm," in *Proc. 4th Int. Conf. Adv. Electron. Mater., Comput. Softw. Eng. (AEMCSE)*, Mar. 2021, pp. 682–685.

[3] W. Zehao, Y. Chen, H. Yi, and J. Hao, "Research on encryption of accounting data using DES algorithm," in *Proc. World Autom. Congr. (WAC)*, Oct. 2022, pp. 513–516.

[4] S. K. Mousavi, A. Ghaffari, S. Besharat, and H. Afshari, "Security of Internet of Things based on cryptographic algorithms: A survey," *Wireless Netw.*, vol. 27, no. 2, pp. 1515–1555, Feb. 2021.

[5] P. Chandi, A. Sharma, A. Chhabra, and P. Gupta, "A DES-based mechanism to secure personal data on the Internet of Things," in *Proc. ICCCE*, A. Kumar and S. Mozar, Eds. Singapore: Springer, 2019, pp. 45–53.

[6] N. Tihanyi, "Report on the first DES fixed points for non-weak keys: Case-study of hacking an IoT environment," *IEEE Access*, vol. 10, pp. 77802–77809, 2022.

[7] F. Pereira, P. Crocker, and V. R. Q. Leithardt, "PADRES: Tool for PrivAcy, data REgulation and security," *SoftwareX*, vol. 17, Jan. 2022, Art. no. 100895.

[8] W. Tuchman, "Hellman presents no shortcut solutions to the DES," *IEEE Spectr.*, vol. S-16, no. 7, pp. 40–41, Jul. 1979.

[9] D. Coppersmith, D. B. Johnson, and S. M. Matyas, "A proposed mode for triple-DES encryption," *IBM J. Res. Develop.*, vol. 40, no. 2, pp. 253–262, Mar. 1996.

[10] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *J. Cryptol.*, vol. 4, no. 1, pp. 3–72, 1991.

[11] M. Tunstall, "Practical complexity differential cryptanalysis and fault analysis of AES," *J. Cryptograph. Eng.*, vol. 1, no. 3, p. 219, 2011.

[12] S. Andonov, J. Dobreva, L. Lumburovska, S. Pavlov, V. Dimitrova, and A. P. Mitrovikj, "Application of machine learning in DES cryptanalysis," in *Proc. ICT Innov.*, 2020, pp. 1–11.

[13] K. Dworak and U. Boryczka, "Breaking data encryption standard with a reduced number of rounds using metaheuristics differential cryptanalysis," *Entropy*, vol. 23, no. 12, p. 1697, Dec. 2021.

[14] T. Yadav and M. Kumar, "Differential-ML distinguisher: Machine learning based generic extension for differential cryptanalysis," in *Proc. Int. Conf. Cryptol. Inf. Secur. Latin Amer.* Cham, Switzerland: Springer, 2021, pp. 191–212.

[15] G. Wang and G. Wang, "Improved differential-ML distinguisher: Machine learning based generic extension for differential analysis," in *Proc. Int. Conf. Inf. Commun. Secur.* Cham, Switzerland: Springer, 2021, pp. 21–38.

[16] A. De Paola, L. Gagliano, and G. Lo Re, "Bayesian system for differential cryptanalysis of DES," *IERI Proc.*, vol. 7, pp. 15–20, Jan. 2014.

[17] E. F. Brickell, J. H. Moore, and M. R. Purtill, "Structure in the *S*-boxes of the DES," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 263. Berlin, Germany: Springer, 1987, pp. 3–8.

[18] M. H. Dawson and S. E. Tavares, "An expanded set of S-box design criteria based on information theory and its relation to differential-like attacks," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 547. Berlin, Germany: Springer, 1991, pp. 352–367.

[19] M. A. Khan, A. Ali, V. Jeoti, and S. Manzoor, "A chaos-based substitution box (S-box) design with improved differential approximation probability (DP)," *Iranian J. Sci. Technol., Trans. Elect. Eng.*, vol. 42, pp. 219–238, Jun. 2018.

[20] M. Khan and Z. Asghar, "A novel construction of substitution box for image encryption applications with gingerbreadman chaotic map and S_8 permutation," *Neural Comput. Appl.*, vol. 29, no. 4, pp. 993–999, Feb. 2018.

[21] A. Belazi and A. A. A. El-Latif, "A simple yet efficient S-box method based on chaotic sine map," *Optik*, vol. 130, pp. 1438–1444, Feb. 2017.

[22] H. Liu, A. Kadir, and C. Xu, "Cryptanalysis and constructing S-box based on chaotic map and backtracking," *Appl. Math. Comput.*, vol. 376, Jul. 2020, Art. no. 125153.

[23] N. T. Courtois and G. V. Bard, "Algebraic cryptanalysis of the data encryption standard," in *Cryptography and Coding* (Lecture Notes in Computer Science), vol. 4887. Berlin, Germany: Springer, 2007, pp. 152–169.

[24] N. T. Courtois, P. Sepehrdad, P. Sušil, and S. Vaudenay, "*ElimLin* algorithm revisited," in *Fast Software Encryption* (Lecture Notes in Computer Science), vol. 7549. Berlin, Germany: Springer, 2012, pp. 306–325.

[25] M. Matsui, "Linear cryptanalysis method for DES cipher," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 765. Berlin, Germany: Springer, 1994, pp. 386–397.

[26] D. Davies and S. Murphy, "Pairs and triplets of DES S-boxes," *J. Cryptol.*, vol. 8, no. 1, pp. 1–25, Dec. 1995.

[27] T. K. Sivakumar, T. Sheela, R. Kumar, and K. Ganesan, "Enhanced secure data encryption standard (ES-DES) algorithm using extended substitution box (S-box)," *Int. J. Appl. Eng. Res.*, vol. 12, no. 21, pp. 11365–11373, 2017.

[28] A. D. Dwivedi, P. Morawiecki, and G. Srivastava, "Differential cryptanalysis of round-reduced SPECK suitable for Internet of Things devices," *IEEE Access*, vol. 7, pp. 16476–16486, 2019.

[29] A. D. Dwivedi and G. Srivastava, "Differential cryptanalysis of round-reduced LEA," *IEEE Access*, vol. 6, pp. 79105–79113, 2018.

[30] M. Cao and W. Zhang, "Related-key differential cryptanalysis of the reduced-round block cipher GIFT," *IEEE Access*, vol. 7, pp. 175769–175778, 2019.

[31] H. Zhao, G. Han, L. Wang, and W. Wang, "MILP-based differential cryptanalysis on round-reduced Midori64," *IEEE Access*, vol. 8, pp. 95888–95896, 2020.

[32] D. Koller and N. Friedman, "Probabilistic graphical models: Principles and techniques," in *Adaptive Computation and Machine Learning*. Cambridge, MA, USA: MIT Press, 2009.

[33] L. Gagliano, "Progettazione e implementazione di un sistema bayesiano per la crittoanalisi differenziale del DES," M.S. thesis, Dept. Eng., Univ. Palermo, Palermo, Italy, 2012.

[34] E. Biham and A. Shamir, *Differential Cryptanalysis of the Full 16-Round DES*. New York, NY, USA: Springer-Verlag, 1993, pp. 487–496.

VINCENZO AGATE received the bachelor's, master's, and Ph.D. degrees in computer engineering from the University of Palermo, Italy, in 2012, 2016, and 2020, respectively. He is currently an Assistant Professor with the University of Palermo. His current research interests include distributed systems, mobile crowdsensing, reputation management, and privacy-preserving systems.

FEDERICO CONCONE received the master's and Ph.D. degrees in computer engineering from the University of Palermo, Italy, in 2017 and 2021, respectively. He is currently a Postdoctoral Research Fellow with the University of Palermo. During his Ph.D. studies, his research mainly focused on social sensing and online social networks. His current research interests include cyber security, social network analysis, adversarial machine learning, and machine learning techniques with applications in smart environments.

ALESSANDRA DE PAOLA (Member, IEEE) received the bachelor's, master's, and Ph.D. degrees in computer engineering from the University of Palermo, Italy, in 2004, 2007, and 2011, respectively. She has been an Associate Professor of computer engineering with the University of Palermo, since 2021. Her current research interests include artificial intelligence applied to distributed systems, wireless sensor networks, ambient intelligence, and network security.

PIERLUCA FERRARO received the bachelor's, master's, and Ph.D. degrees in computer engineering from the University of Palermo, Italy, in 2010, 2013, and 2017, respectively. He has been an Assistant Professor of computer engineering with the University of Palermo, since 2020. His current research interests include mobile and pervasive computing, mobile crowdsensing, and privacy-preserving computation.

GIUSEPPE LO RE (Senior Member, IEEE) received the Laurea degree in computer science from the University of Pisa, in 1990, and the Ph.D. degree in computer engineering from the University of Palermo, in 1999. In 1991, he joined the Italian National Research Council (CNR), where he achieved a Senior Researcher position. He has been a Full Professor of computer engineering with the University of Palermo, since 2019. His current research interests include computer networks and distributed systems, focusing on reputation and security systems. He is a Senior Member of the IEEE Communication Society and the Association for Computer Machinery.

MARCO MORANA received the Laurea and Ph.D. degrees in computer engineering from the University of Palermo, Italy, in 2007 and 2011, respectively. He has been an Assistant Professor of computer engineering with the University of Palermo, since 2016. During his Ph.D. studies, his research mainly focused on computer vision and pattern recognition. His current research interests include parallel and distributed computing, social network analysis, cyber security, intelligent data analysis for user profiling, data fusion, and reasoning in smart environments.

Open Access funding provided by 'Università degli Studi di Palermo' within the CRUI CARE Agreement