A Framework for Describing Block Cipher Cryptanalysis

Raphael C.-W. Phan, *Member*, *IEEE*, and Mohammad Umar Siddiqi

Abstract—Block ciphers provide confidentiality by encrypting confidential messages into unintelligible form, which are irreversible without knowledge of the secret key used. During the design of a block cipher, its security against cryptanalysis must be considered. History has shown that a cipher designed without an adequate treatment of this will often lead to flaws and attacks by other researchers, sometimes devastatingly so. The problem for an aspiring cipher designer is that there are no standard texts on block cipher cryptanalysis because it is a fast changing field. The commonly available references are academic journals and conference proceedings, which may not be easy to grasp for researchers new to cryptanalysis. This paper presents the Xi framework, which is designed to compactly describe the block cipher cryptanalysis techniques regardless of their individual differences. This provides the cryptanalyst with a general framework to describe attacks on block ciphers, with the additional capabilities of allowing specification of the technical details of each different type of attack and of comparison of their respective strengths. Comparing different distinguishers in this framework also allows us to see natural generalizations and trigger nice open problems. We then show how to apply this Xi framework to the description of various attacks on popular and recent block ciphers.

Index Terms—Encryption, cryptanalysis, block ciphers, framework, generalization, distinguishers.

1 Introduction

Block ciphers [25] are secret-key encryption algorithms that encrypt plaintext messages n bits at a time into unintelligible ciphertext, under the control of a secret key K. Examples of block ciphers include the Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) [12] adopted by the US National Institute of Standards and Technology (NIST) in 2001 as a Federal Information Processing Standard (FIPS) [27] for protection of nonclassified sensitive information.

During the design of a block cipher, its security against cryptanalysis needs to be considered. Cryptanalysis typically involves studying how resistant a cipher is against distinguishing attacks and key-recovery attacks. *Distinguishing attacks* are those that show that a cipher's structure exibits some identifiable nonrandomness that allows someone to differentiate between a black box containing the cipher E_K and a black box containing a random permutation $\pi : \{0,1\}^n \rightarrow \{0,1\}^n$. *Key-recovery attacks* are those where a cryptanalyst strives to obtain the secret key K; thus any ciphertext can be decrypted back to the plaintext.

The most naive key-recovery attack is the *exhaustive key search*, which involves guessing all possible values of K by brute force and, for each guess, verifying via trial decryption of the ciphertext if the guessed value is correct. Hence, the overall effort is $2^{|K|}$ trial decryptions, where $|K|$ denotes the length of K in bits. This attack applies to any cipher, regardless of its internal structure, and is only dependent on the size of the key K. Because of this, the exhaustive key search is generally used as a benchmark. If other cryptanalytic attacks exist on a cipher that require less effort than exhaustive key search, the cipher is considered broken because it would then be better to use any other unbroken $|K|$-bit key cipher.

Cryptanalysis is important during the design of a cipher to show how secure a cipher is, how sound the design principles are, and whether there are any observable flaws in the structure. Results in cryptanalysis thus serve as certificational strengths of a cipher. Users like to be assured of this for any security system that they are using.

To date, modern cryptanalysis methods for block ciphers are numerous as compared to the early 1990s when the *differential cryptanalysis* [6], the first generic cryptanalysis technique to apply to a large class of ciphers, was introduced. For an aspiring cryptanalyst, understanding the concepts of block cipher cryptanalysis is a hard task since there is no defined textbook on the subject. The sole references are the journal papers and proceedings of conferences, which are technical and suitable mostly for advanced cryptanalysis researchers.

In this paper, we take the first step toward bridging the gap between these two extreme sides of the cryptanalytic research community by generalizing the concepts behind various cryptanalytic techniques and defining the *Xi* (Ξ) *framework* as a general framework for compactly describing cryptanalytic attacks on block ciphers. Independently, in [33], a *commutative diagram framework* was proposed to include some block cipher cryptanalysis techniques. We shall briefly discuss the similarities and differences between both frameworks in Section 5.

Currently, there exists no uniform method or notation for describing cryptanalytic attacks. Attacks are described in paragraph upon paragraph of text and, hence, do not

- *R.C.-W. Phan is with the Information Security Research (iSECURES) Lab, Swinburne Univerity of Technology (Sarawak Campus), 1st Floor, State Complex, 93576 Kuching, Malaysia. E-mail: rphan@swinburne.edu.my.*
- *M.U. Siddiqi is with the Faculty of Engineering, International Islamic University Malaysia, PO Box 10, 50728 Kuala Lumpur, Malaysia. E-mail: umarsiddiqi@iiu.edu.my.*

Manuscript received 18 Oct. 2004; revised 23 Nov. 2005; accepted 20 Apr. 2006; published online 21 Sept. 2006.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-0329-1004.

provide a concise view for the interested reader. The Xi framework permits compact representation of these attacks and can easily be translated to pseudocode and later to source code for test implementations, which further enhances the reader's understanding of the concepts. This framework also allows for highlighting the similar abstract structure of various cryptanalytic techniques and, so, may enable the fusion of ideas used in a certain technique to other techniques and trigger some interesting open problems as a natural consequence. We highlight some of these fusions in Section 3.

Section 2 first outlines the basic notations to be used throughout this paper and then describes a general cryptanalytic attack in algorithmic form. Section 3 next discusses various block cipher cryptanalysis techniques by viewing them as exploiting unique distinguishers. Section 4 then presents two examples of how the proposed (Ξ) framework can be used to describe attacks on the DES and AES. In Section 5, we also relate between our framework and that proposed in [33]. We finally conclude in Section 6 and motivate some open problems.

2 The Xi (Ξ) Framework

In this section, we present the basic notations used in the Xi framework for representing cryptanalytic attacks on block ciphers. Concrete application examples will be presented in the later sections.

2.1 Basic Notations

The preliminary notations include symbols used to denote the various parts of a block cipher:

- $\mathcal{P} = \{0,1\}^n \triangleq$ set of all possible n-bit texts.
- $\mathcal{K} = \{0,1\}^k \triangleq$ set of all possible k-bit secret keys.
- $P \in \mathcal{P} \triangleq$ n-bit plaintext.
- $C \in \mathcal{P} \triangleq$ n-bit ciphertext.
- $K \in \mathcal{K} \triangleq$ k-bit secret key.
- $P^* = (P_1, P_2, \ldots) \triangleq$ a sequence of distinct plaintexts.
- $C^* = (C_1, C_2, \ldots) \triangleq$ a sequence of distinct ciphertexts.
- $n = |P| = |C| \triangleq$ block length of plaintext/ciphertext in bits.
- $k = |K| \triangleq$ block length of secret key in bits.
- $E_K(P) = P \xrightarrow{E_K\#r} C \triangleq$ encryption of P to C under K, where E_K is an r-round iterated block cipher.
- $r \triangleq$ total number of rounds of block cipher E_K.
- $E_K^{-1}(.) \triangleq$ inverse of $E_K(.)$.
- $I \triangleq$ identity permutation.
- $\phi \triangleq$ null set.
- $const = \{c_1, c_2, \ldots\} \triangleq$ set of arbitrary constants.
- $? \triangleq$ any arbitrary value.
- $(0^t 1^u ?^v) \triangleq$ a consecutive sequence of t "0"s, followed by u "1"s and then v arbitrary bits.
- $x \leftarrow y \triangleq$ assignment of value y to x.

A block cipher, $E_K(\cdot)$, commonly consists of r iterations of a so-called *round function*, $F(\cdot, RK_i)$, that is keyed by a different round key, RK_i; $i = 1 \ldots r$ in each round:

$$E_K(\cdot) = F(\cdot, RK_r) \circ F(\cdot, RK_{r-1}) \ldots \circ F(\cdot, RK_2) \circ F(\cdot, RK_1). \tag{1}$$

These round keys are derived from the secret key K via a so-called *key schedule*. Oftentimes, recovering bits of some round keys immediately means the cryptanalyst has recovered some bits of K.

It is assumed [31] that the cryptanalyst has access to some plaintexts and corresponding ciphertexts. Then, during cryptanalysis, we are often interested in searching for properties of some middle rounds of a cipher because the outer rounds not covered by these middle rounds can be peeled off by guessing the round keys in these outer rounds and working inward from both the plaintext and ciphertext ends toward those middle rounds. These middle rounds, i.e., the original block cipher with some outer rounds removed from either end, are called the *reduced cipher*, G. To the best of our knowledge, this term was first defined in [15]. For this purpose, we further define:

- $G\#s \triangleq$ reduced cipher comprising the first, last, or middle s consecutive rounds, $s \leq r$.
- $x = \{0,1\}^n \triangleq$ input of reduced cipher, G.
- $y = \{0,1\}^n \triangleq$ output of reduced cipher, G.
- $x^* = (x_1, x_2, \ldots) \triangleq$ a sequence of distinct inputs to G.
- $y^* = (y_1, y_2, \ldots) \triangleq$ a sequence of distinct outputs of G.
- $h_a\#t \triangleq$ arbitrary t outer rounds of E_K prior to G.
- $h_b\#t \triangleq t$ arbitrary t outer rounds of E_K after G.

Example 1. $P \xrightarrow{h_a\#1} x \xrightarrow{G\#5} y \xrightarrow{h_b\#2} C$ denotes that E_K consists of eight rounds in total, which can be further said to consist of a five-round reduced cipher, G, in the middle, while h_a (one-round) and h_b (two-round) denote the outer rounds not covered by the reduced cipher.

2.2 General Description of Cryptanalytic Attacks

A cryptanalytic attack basically involves first searching for an identifiable nonrandomness property within some middle s rounds (a.k.a. the reduced cipher) of the cipher structure that allows us to differentiate between a black box containing that reduced cipher and a black box containing a random permutation $\pi : \{0,1\}^n \rightarrow \{0,1\}^n$. This is known as the *distinguishing property* (or, simply, *distinguisher*) and further allows us to recover keys in the outer rounds not covered by the distinguisher. Therefore, the phases in cryptanalysis are to first search for such a distinguisher (distinguisher search) and then collect enough text information (text collection) to further mount a key-recovery attack to obtain some round keys in the outer rounds. The general description of cryptanalytic attacks is based on the following algorithm:

Algorithm 1. A General Cryptanalytic Attack

1. Distinguisher Search Phase.

   Recall that $x \xrightarrow{G\#s} y$. Then, by exploiting the structure of E_K, find a distinguisher Ξ for G of the form:

$$\Xi(x^*, y^*)\#s \Rightarrow (x^* \overset{p^*}{\longleftrightarrow} y^*). \tag{2}$$

This simply means that our distinguisher, Ξ, takes as input the sequences x^* and y^* and outputs a correlation between them that occurs with a nonrandom probability, p^*. This nonrandom distinguisher for G will be used later to compare with a random permutation, π, where

$$\pi(x^*, y^*)\#s \Rightarrow (x^* \xleftrightarrow{p} y^*) \tag{3}$$

occurs with probability, p. If the difference $(p^* - p)$ in these two probabilities is nonnegligible, i.e., $|p^* - p| > \epsilon$ for some negligible ϵ, then $\Xi(x^*, y^*)$ is useful. The first step and often the hardest part in cryptanalysis is to search for distinguishers that maximize this $|p^* - p|$. A successful discovery of such distinguishers easily translates to a successful key-recovery attack, i.e., Steps 2 and 3 below would then follow naturally.

2. Text Collection Phase.
   Define:

$$\begin{aligned} &P_i = p_1 \parallel p_2 \parallel \ldots |p_l \quad \in \mathcal{P} \text{ for } i \in \{0, 1, \ldots, m\} \\ &\text{where} \quad p_j \in \{0, const, ?\} \text{ for } j \in \{0, 1, \ldots, l\} \\ &\text{and} \quad |p_j| = n/l. \end{aligned}$$

Then, for all $i \in \{0, 1, \ldots, m\}$, obtain P_i and the corresponding $C_i = E_K(P_i)$.

This phase assumes the cryptanalyst has access to a black box oracle that performs either encryption or decryption using the unknown secret key K on plaintexts or ciphertexts supplied by the cryptanalysts. This is a common assumption in cryptanalysis and examples of these scenarios in practical applications may be found in [31].

How flexible the cryptanalyst needs to be when supplying the inputs to the oracle determines what type of attack he is to mount. For instance, if a cipher can be attacked only by the fact that he knows some plaintexts corresponding to some ciphertexts, he is mounting known-plaintext attacks [31]. If, in order to attack, he needs to choose what plaintexts to input to the encryption oracle to obtain the ciphertexts, he is mounting chosen-plaintext attacks [31]. The purpose of classifying attacks in this way is so that it is clear what classes of attacks are applicable when certain text information is accessible to an attacker.

In general, chosen-plaintext attacks give the cryptanalyst more restrictions since the plaintexts have to follow certain relationships among them, e.g., they must be equal in the least significant bit, etc. Differential cryptanalysis [6] and its variants, Square attacks [11] and boomerang attacks [32], are typically chosen-plaintext attacks, while linear cryptanalysis [24], interpolation attacks [15], and slide attacks [9] are typically known-plaintext attacks. Chosen-plaintext attacks can be made [31] to work with only known plaintexts, but at the cost of an increase in the number of texts obtained and, thus, are often not desirable. On the other hand, known-plaintext attacks will often require much fewer texts and complexity if chosen plaintext queries are available to the cryptanalyst, but this means he would have to work in the chosen-plaintext attack model, which is very restricted and more difficult to realize in practical scenarios.

3. Key-Recovery Phase.
   Define:

- $\{RK_a\} \triangleq$ set of all possible guesses of round key in h_a.
- $\{RK_b\} \triangleq$ set of all possible guesses of round key in h_b.
- $RK_a^j \in \{RK_a\} \triangleq$ a current guess of RK_a.
- $RK_b^j \in \{RK_b\} \triangleq$ a current guess of RK_b.

Also note that $==$ denotes equality check, while $\leftarrow$ denotes assignment. Then, the key-recovery phase is given by the steps below:

```
Guess $\forall(RK_a, RK_b)$ and do
(i) $\forall(P_i, C_i)$, compute
    $\tilde{x_i} \leftarrow h_a(P_i, RK_a^j)$ and
    $\tilde{y_i} \leftarrow h_b^{-1}(C_i, RK_b^j)$.
(ii) Compute $A = \mathcal{D}(\Xi, \tilde{x_i}, \tilde{y_i})$ defined by:-
    If $\mathcal{D}(\Xi, \tilde{x_i}, \tilde{y_i}) == \Xi(x^*, y^*)\#s \Rightarrow (x^* \xleftrightarrow{p^*} y^*)$,
        return $A = 1$.
    Otherwise if $\mathcal{D}(\Xi, \tilde{x_i}, \tilde{y_i}) == \pi(x^*, y^*)\#s \Rightarrow (x^* \xleftrightarrow{p} y^*)$,
        return $A = 0$.
(iii) Compute $\mathcal{F}(A, RK_a^j, RK_b^j, \{RK_a\}, \{RK_b\})$ defined by:-
    Case of $|\{RK_a\}| = 2^{|RK_a|}$, AND $|\{RK_b\}| = 2^{|RK_b|}$ (Sieving)
      If $A \neq 1$,
          $\{RK_a\} \leftarrow \{RK_a\} - RK_a^j$
      and $\{RK_b\} \leftarrow \{RK_b\} - RK_b^j$.
OR
    Case of $\{RK_a\} = \phi$, AND $\{RK_b\} = \phi$ (Counting)
      If $A == 1$,
          $\{RK_a\} \leftarrow \{RK_a\} + RK_a^j$
      and $\{RK_b\} \leftarrow \{RK_b\} + RK_b^j$.
```

The key-recovery phase is commonly known as the actual analysis phase because it is during this time that the cryptanalyst uses the distinguisher (from Phase 1) and the texts (from Phase 2) to recover some bits of the secret key K. The overall computational effort of cryptanalysis is mostly due to this phase because it is required to guess all possible round key values in the outer rounds (i.e., RK_a, RK_b) and, for each guess, process all collected texts (i.e., P_i, C_i) by working inward through the outer rounds toward the middle s rounds, see Step (i). Then, with these results, we proceed to Step (ii), which performs a distinguishing process $\mathcal{D}$ to determine whether it is the reduced cipher G or a random permutation that is detected. Key guesses that cause the reduced cipher to be detected will be put in the list of possibly correct keys, while the wrong guesses are discarded from the list, see Step (iii). Sometimes this is called the filtering process, $\mathcal{F}$. Depending on the type of distinguisher used, this key list is built by either sieving or counting. Sieving means to start from a list of all possible outer round key values, gradually discarding the wrong keys until the list is small enough to include only a few possibly correct keys. Counting means to start from an empty list and then gradually add possibly correct outer round keys to it. Except for impossible differential attack, which uses sieving because key guesses that cause an impossible property are discarded from the list and the list therefore slowly dwindles, the other attacks use counting because the distinguisher property has a much higher probability than for a random case, thus starting from an empty list, then keying guesses that induce this property are quite likely to be correct keys, so they are added to the list.

Since distinguishers vary from one cryptanalytic technique to another, in the next section, we will describe each type in turn for a better understanding of what is being exploited in each.

3 Distinguishers

In this section, we will describe the distinguishers for the various cryptanalytic techniques and highlight the common structures between them. In some cases where specific notations are required, they will be defined as necessary.

3.1 Differential Cryptanalysis

Differential cryptanalysis [6] considers how a pair of inputs (x, x') with a specific difference Δx propagates with some probability p^* through some s rounds of a cipher to result in a corresponding output pair (y, y') with a specific difference Δy. For this attack to work, sets of plaintext pairs with the desired input difference Δx need to be obtained, thus this is a type of chosen-plaintext attack [31].

Let:

$$\begin{aligned}\Delta x = x \circ (x')^{-1} &\triangleq \text{input difference to } G\\ \Delta y = y \circ (y')^{-1} &\triangleq \text{output difference of } G.\end{aligned}$$

Here, (x, x') and (y, y') are a distinct pair of inputs and outputs of G, respectively, while $\circ$ denotes the group operation on the elements in the cipher, usually taken to be the exclusive-OR, $\oplus$. In such a case, every element is its own inverse and, so, we have $\Delta x = x \oplus x'$ and $\Delta y = y \oplus y'$.

With this in place, then the *differential distinguisher* [6] is defined as:

$$\Xi^{\text{Differential}}(x^*, y^*)\#s = \Xi(x, x', y, y')\#s \Rightarrow (\Delta x \xrightarrow{p^*} \Delta y),$$

where $|p^* - (p = 2^{-n})| > \epsilon$. The distinguisher, Ξ, uses both the input and output pairs, (x, x') and (y, y'), and states that the difference between the input pairs, Δx, would result in an output pair with difference Δy, with a probability, p^*, after going through the s-round reduced cipher. Note that, in order for this distinguisher to be useful in an attack, p^* must be significantly different from the probability, p, that the output difference Δy equals a specific n-bit string. Here, $p = 2^{-n}$ since the probability[1] of an output pair, (y, y') having a specific difference, Δy, between them is 2^{-n}.

With this differential distinguisher, key-recovery (see Algorithm 1) then proceeds by guessing the round keys of outer rounds, working from both the plaintext and ciphertext ends to peel off the outer rounds until the middle s rounds are reached, and then checking if this differential distinguisher exists. A key guess that causes this is added to the list of possible key values, and this process is repeated for all plaintext-ciphertext pairs. At the end, the key value that has been added to the list the greatest number of times is likely to be the correct key.

3.2 Linear Cryptanalysis

Linear cryptanalysis [24] does not require plaintexts to have any specific difference or any other relationship between them and, thus, has fewer practical restrictions for the cryptanalyst. It is a type of known-plaintext attack [31] since the cryptanalyst requires some plaintexts with corresponding ciphertexts, but the plaintexts do not have to be chosen to satisfy any particular relationships.

Let:

$$\begin{aligned}a, b &\triangleq \{0,1\}^n\\ \bullet &\triangleq \text{dot product}\\ \wp(.) &\triangleq \text{parity function.}\end{aligned}$$

The a and b are often known as the input and output bit masks, respectively, and are used in dot products with the input and output of G to select only certain bits while masking off the other bits. Meanwhile, the parity function, $\wp$, simply exclusive-ORs all the bits of a binary string. A *linear distinguisher* [24] is then defined as:

$$\Xi^{\text{Linear}}(x^*, y^*)\#s = \Xi(x, a, y, b)\#s \Rightarrow (\wp(x \bullet a) \overset{p^*}{=} \wp(y \bullet b)),$$

where $|p^* - (p = 1/2)| > \epsilon$. This distinguisher states that the parity of certain bits of the input of G are equal to the parity of some other bits of the output of G, with nonrandom probability p^*. Again, a similar requirement exists that p^* be as different from p as possible. Note here that p is the probability that the parity of a random binary string has a certain 1-bit value, thus $p = 1/2$ since there are only two possible values, "1" or "0."

3.3 Higher-Order Differential Cryptanalysis

The conventional dth-*order differential distinguisher* [17] is defined as:

$$\Xi^{\text{d}-\text{Differential}}(x^*, y^*)\#s = \Xi(x_1, x_2, \ldots, x_{2^d}, y_1,\ y_2\ , \ldots, y_{2^d})\#s$$
$$\Rightarrow \left(\bigoplus_{x \in x^*} G(x) = \bigoplus_{y \in y^*} y \overset{p^*}{=} c\right),$$

where $|p^* - (p = 2^{-n})| > \epsilon$ and $c \in const$. Of course, there are other variants of the higher-order differential attack, such as the one using rational expressions [26], in which case the distinguisher above would differ.

A dth-order or higher-order differential distinguisher is an extension of the conventional first-order differential distinguisher described in Section 3.1, by making use of more than one difference between any pair (e.g., x_i, x_j; $i \neq j$). The outputs of G are then all summed and checked if a constant c results. Notice that the probability, p, is the same as for the case of the first-order differential distinguisher because, for any random n-bit constant c, the probability that it equals a certain n-bit value is 2^{-n}.

1. In more detail, given that y and y' are both n-bit strings, there are 2^n possible values that each of them may have. There are then ${}^{2^n}C_2 = 2^n!/[(2^n-2)!2!] \approx 2^{2n-1}$ possible ways to obtain the pair of values, y and y'. Out of these, there are 2^n instances when $y \oplus y' = \Delta y$. Therefore, the probability is $2^n/2^{2n-1} \approx 2^{-n}$ that any random pair, y and y', would have the difference Δy.

3.4 Truncated Differential Cryptanalysis

Let:

- $\Delta\alpha = x \circ (x')^{-1} = (\alpha_1 \| \alpha_2 \| \ldots \| \alpha_l) \triangleq$ truncated input difference to G,
- $\Delta\beta = y \circ (y')^{-1} = (\beta_1 \| \beta_2 \| \ldots \| \beta_l) \triangleq$ truncated output difference of G,

where $\alpha_i, \beta_i \in \{0, const, ?\}$.

Here, we are interested in looking at wordwise (truncated) differences whose values could be 0, constant, or arbitrary, where the wordsize is less than blocksize n. Since some bit differences are arbitrary, there are no restrictions on what these bits of the plaintexts should be, thus the cryptanalyst can form more pairs with a given amount of texts. This is the advantage of truncated over conventional differences. The *truncated differential distinguisher* [17] is defined as:

$$\Xi^{\text{Truncated}}(x^*, y^*)\#s = \Xi(x, x', y, y')\#s \Rightarrow (\Delta\alpha \xrightarrow{p^*} \Delta\beta),$$

where $|p^* - (p = 2^{-t \times n/l})| > \epsilon$ and $t = \sum i$ when

$$\beta_i \in \{0, const\}.$$

Notice that this distinguisher is very similar to the differential distinguisher in Section 3.1 because a pair of inputs and outputs are used and the difference between them is observed. Thus, this distinguisher is used in a key-recovery attack in basically the same way as described in Section 3.1. What differs is that, in this case, the truncated difference can be specified more flexibly. The probability, p, here that a random output difference $\Delta\beta$ has a certain value depends on the number of fixed (0 or constant) word differences, β_i in $\Delta\beta$.

3.5 Impossible Differential Cryptanalysis

The *impossible differential distinguisher* [18], [3] is defined as:

$$\Xi^{\text{Impossible}}(x^*, y^*)\#s = \Xi(x, x', y, y')\#s \Rightarrow (\Delta x \overset{p^*=0}{\longrightarrow} \Delta y),$$

where $|p^* - (p = 2^{-n})| > \epsilon$. The impossible differential distinguisher is a special case of the differential distinguisher in Section 3.1, the only difference being that the nonrandom probability, p^*, is 0, which is now much smaller than the probability, $p = 2^{-n}$, that a random difference Δy equals a certain fixed n-bit string.

To use this distinguisher in a key-recovery attack, the basic idea here is to use this impossible property to do sieving [13]. We start with a list of all possible keys of the outer rounds and, for each such guess, we work inward toward the middle s rounds that make up the distinguisher and check for this property. A random key guess would cause this property with probability $p = 2^{-n}$, while the correct key will never cause this; thus key guesses that cause this impossible property are definitely wrong and therefore removed (hence sieving) from the list of possible keys. This is repeated until only a few possible keys remain in the list.

3.6 Boomerang Attack

The boomerang attack [32] uses a *boomerang distinguisher* that resembles a second-order differential distinguisher in that four inputs and outputs of G are used, with the special property that their exclusive-OR sum in the middle of the cipher results in the constant 0. First, let the reduced cipher, G, consist of two halves, g_a and g_b, respectively.

Let:

$x_1, \ldots, x_4$	$\triangleq$	inputs to g_a (and therefore of G)
$w_1, \ldots, w_4$	$\triangleq$	outputs of g_a (and therefore inputs to g_b)
$y_1, \ldots, y_4$	$\triangleq$	outputs of g_b (and therefore of G).

The boomerang distinguisher is defined as:

$$\Xi^{\text{Boomerang}}(x^*, y^*)\#s = \Xi(x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4)\#s \Rightarrow$$
$$(\Delta x = x_1 \oplus x_2, \Delta y = y_1 \oplus y_3 = y_2 \oplus y_4 \rightarrow \Delta x \overset{p^*}{=} x_3 \oplus x_4),$$

where $|p^* - (p = 2^{-n})| > \epsilon$. Note that the probability, p, is the same as for the case of the first-order differential distinguisher because we expect the difference $x_3 \oplus x_4$ to equal the fixed n-bit difference value Δx, which happens with probability 2^{-n}.

The boomerang attack is an adaptively-chosen-plaintext attack [31] because, after obtaining the ciphertexts corresponding to plaintext pairs with a desired difference, new ciphertext pairs are generated with a specific difference and then their corresponding plaintext pairs need to be obtained. Some improvements in [16], [4] convert this to the less restricted chosen-plaintext attack at the expense of more texts to be obtained.

3.6.1 *Fusion of Ideas from Different Distinguishers*

Using our framework allows us to compare between the different distinguishers, not only highlighting the similar structures, but also considering why differences exist and if such differences could be fused together. In fact, the natural consequences of considering a fusion of ideas from higher-order, truncated, impossible differential and boomerang-style distinguishers result in many interesting generalizations.

For example, borrowing from ideas of higher-order distinguishers, boomerang-style distinguishers can be enhanced by considering that, instead of just a pair (x_1, x_2) of inputs and outputs (y_1, y_2), we take a set of them at a time. This was used to form *boomerang amplifiers* with n-tuples in [16]. Such distinguishers can be viewed as *higher-order boomerang* distinguishers.

Further, we could consider *higher-order truncated differential* or *higher-order truncated boomerang* distinguishers that could either occur with probability p^* much greater than p for the random case or with probability $p^* = 0$. For example, recently, [5] presented a related-key impossible boomerang-style distinguisher for the COCONUT98 cipher that fused ideas from truncated, impossible differential and boomerang-style distinguishers.

3.7 Interpolation Attack

The *interpolation distinguisher* as originally proposed in [15] is based on the Lagrange interpolation formula and defined as:

$$\Xi^{\text{Interpolation}}(x^*, y^*)\#s = \Xi(x_1, x_2, \ldots, x_d, y_1, y_2, \ldots, y_d)\#s$$
$$\Rightarrow \left(y = f(x) = \sum_{i=1}^{d} y_i \prod_{i \le j \le d; j \ne i} \frac{x - x_j}{x_i - x_j} \right).$$

Specifically, the output, y, of the reduced cipher, G, is expressed as a polynomial, $f(x)$, constructed by using d other inputs, x_i, and corresponding outputs, y_i, of the reduced cipher, G. For this distinguisher to be effective, the time to construct such polynomials should be much smaller than the time to perform trial encryptions later on during the actual analysis phase. To apply this distinguisher to key-recovery, round keys RK_b of the outer rounds after G are guessed and, for each guess, the d ciphertexts are partially decrypted to the point right after G (denote these as y_i; $i \in \{1, 2, \ldots, d\}$). Use these d outputs, y_i, of the reduced cipher and their corresponding inputs, x_i, to form the polynomial $f(x)$. Then, use an extra x_i, y_i pair (not already used) to verify the correctness of the constructed $f(x)$. If this is verified, the current key guess is possibly correct; otherwise, it is wrong. Interpolation attacks are typically known-plaintext attacks, though chosen-plaintext variants are also possible.

Some other variants of this attack are found in [26], [1], [22] that use rational (instead of polynomial) expressions, Gaussian elimination instead of polynomial interpolation, and Rabin's [30] root finding algorithm, respectively.

3.8 Square Attack

Let:

$$g : \{0,1\}^w \to \{0,1\}^w \quad \stackrel{\Delta}{=} \quad \text{an internal component of } G,$$

where $w \le n$. Then, the *Square distinguisher* [11] is defined as:

$$\Xi^{\text{Square}}(x^*, y^*)\#s = \Xi(x_1, x_2, \ldots, x_{2^w}, y_1, y_2, \ldots, y_{2^w})\#s \Rightarrow$$
$$\left(\bigoplus_{x \in x^*} G(x) = \bigoplus_{y \in y^*} y \stackrel{p^*}{=} 0 \right),$$

where $|p^* - (p = 2^{-n})| > \epsilon$ or, if each w-bit sum of the output of G is independent, then we have $|p^* - (p = 2^{-w})| > \epsilon$. This distinguisher checks a collection of 2^w texts at a time such that, after going through the reduced cipher, G, the exclusive-OR sum of all texts is 0. Again, the probability, p, is the same as in differential cryptanalysis because the probability for a random n-bit (respectively, w-bit) sum equaling 0 is 2^{-n} (respectively, 2^{-w}).

This is a chosen-plaintext attack because the collection of 2^w texts must satisfy the strict requirement that, at the input to G, they are all unique values in at least one w-bit word.

3.8.1 Fusion of Ideas from Different Distinguishers

Note that this distinguisher is especially similar to the higher-order differential distinguisher in that a collection of texts is used and the corresponding outputs of G are summed and checked if equal to a constant (which is 0 in this case). From this observation, we can view the Square attack as a special case of the higher-order differential attack, particularly a wth order differential attack where the constant $c = 0$. As a natural consequence, we could now consider more general cases of c, e.g., where only some bits of c are fixed (to either 0 or 1) while other bits are left arbitrary, resulting in what can be called a *truncated Square distinguisher* (for instance, see [21]). Further note that existing Square distinguishers in the literature are all deterministic. Comparing the higher-order and Square distinguisher suggests considering *probabilistic Square distinguishers* (for which $p^* < 1$) instead of deterministic Square distinguishers (for which $p^* = 1$). In fact, the existence of probabilistic Square distinguishers was raised as an open problem in [21] and is shown here to be a natural consequence of comparing different distinguishers in our framework.

The ability to relate the similar structure (e.g., using exclusive-OR sum) of different distinguishers (in this case, higher-order and Square) to consider natural generalizations of previous work and to trigger fresh open problems is one of the features of this framework.

3.9 Slide Attack

The *slide distinguisher* [9], [10] is defined as:

$$\Xi^{\text{Slide}}(x^*, y^*)\#s = \Xi(x, x', y, y')\#s = \Xi(P, P', C, C')\#r \Rightarrow$$
$$\begin{cases} x = h_a(P) & \stackrel{p^*}{=} x' = P' \\ y' = h_b^{-1}(C') & \stackrel{p^{**}=1}{=} y = C, \end{cases}$$

where $|p^* - (p = 2^{-n} \times 2^{-n})| > \epsilon$. A slide distinguisher is slightly involved in that a cryptanalyst needs to consider not only a pair of inputs and outputs of the reduced cipher, G, but also the corresponding pair of plaintexts and ciphertexts of the whole block cipher E_K. The two equations to the right of the brace in the distinguisher above are known as *slid equations* and are such that, when the upper equation is satisfied with some probability p^*, then the lower equation is automatically satisfied (probability $p^{**} = 1$). This gives an overall nonrandom probability, p^* of both equations being satisfied. For this to be satisfied randomly, the probability, p is $2^{-n} \times 2^{-n}$ since there are two equations, each involving n-bit binary strings.

The slide attack is typically a known-plaintext attack because, from a collection of plaintexts, the cryptanalyst could search for one or more useful pairs that satisfy the slid equations. This can be converted into a chosen-plaintext attack too, provided the outer rounds are weak enough that they do not require much effort to pass through. For example, three rounds of DES is weak [9], [10].

3.9.1 Fusion of Ideas from Different Distinguishers

Techniques from conventional differential and linear distinguishers have been applied to enhance the slide attack in [10], [14]. When viewed in our framework, this means, in the particular case of using differential techniques [10], to consider a set of Ps such that some plaintext difference propagates through h_a probabilistically through to the input of G, while, in the case of using linear techniques [14], this means to consider a set of Ps such that the parity of some bits of P is equal with some probability to the parity of some other bits of the input of G.

A further generalization of this is to consider using techniques from higher-order differential or Square distinguishers to enhance ideas in [10], i.e., to consider a set of plaintexts Ps with some desired difference between them.

An open problem is to consider if p^{**} could be less than one, in which case, we would have *probabilistic slide distinguishers*.

4 Example Applications of the Xi Framework

In this section, we will show how the Xi framework can be used to describe the cryptanalysis of popular block ciphers in a compact form.

4.1 Differential Cryptanalysis of DES

We consider describing a differential cryptanalysis of four-round DES by using the Xi framework. We first let the four-round DES encryption of plaintext, P, under key K be denoted by:

$$DES_K(P) = \rho_4(\rho_3(\rho_2(\rho_1(P, K)))). \tag{4}$$

Let $\Delta x = (0^2 1^1 0^{29} \| 0^{32})$.

1. The distinguisher is

   $$\Xi_{DES}^{Differential}(x^*, y^*) = \Xi_{DES}^{Differential}(x, x', y, y')\#2$$
   $$\Rightarrow \ (x \oplus x' = \Delta x \overset{p^*=1}{\longrightarrow} y \oplus y' = \Delta y = (?^4 0^{28} | 0^2 1^1 0^{29})).$$

   For a random permutation,

   $$\pi(x^*, y^*) \Rightarrow (x \oplus x' = \Delta x \overset{p=2^{-28}}{\longrightarrow} y \oplus y' = \Delta y).$$

   Since $|(p^* = 1) - (p = 2^{-28})| > \epsilon$, $\Xi_{DES}^{Differential}$ is useful.
2. Get P_i, P_i' s.t. $P_i \oplus P_i' = \Delta x$ for $i \in \{0, 1, \ldots, 15\}$ and corresponding $C_i = DES_K(P_i)$.
3. Set $h_a = I$, $h_b = \rho_4$ $(\rho_3(.))$, RK_b = round key of ρ_4, $\{RK_b\} = \phi$.

```
Guess $\forall RK_b^j$ for $j \in \{0, 1, \ldots, 2^{28}\}$, and do
(i) $\forall C_i$, compute
  $\tilde{y}_i \leftarrow h_b^{-1}(C_i, RK_b^j)$.
(ii) Compute $A = \mathcal{D}(\Xi, \tilde{y}_i)$ defined by:
  If $\Xi(\tilde{y}_i) \Rightarrow \Delta y$, return $A = 1$.
  Otherwise, return $A = 0$.
(iii) Compute $\mathcal{F}(A, RK_b^j, \{RK_b\})$ defined by:
  If $A == 1$,
    $\{RK_b\} \leftarrow \{RK_b\} + RK_b^j$.
```

4.2 Square Attack on Four-Round AES

We next consider describing Square attack on four-round AES. First, we let the four-round AES encryption of plaintext, P, under key K be denoted by:

$$AES_K(P) = \rho_4(\rho_3(\rho_2(\rho_1(P, K)))). \tag{5}$$

1. The distinguisher is

   $$\Xi_{AES}^{Square}(x^*, y^*) =$$
   $$\Xi_{AES}^{Square}(x_1, x_2, \ldots, x_{256}, y_1, y_2, \ldots, y_{256})\#3$$
   $$\Rightarrow \left(\bigoplus_{x \in x^*} \rho_3(\rho_2(\rho_1(x, K))) = \bigoplus_{y \in y^*} y \overset{p* = 1}{=} 0 \right).$$

   For a random permutation,

   $$\pi(x^*, y^*)\#3 \Rightarrow \left(\bigoplus_{y \in y^*} y \overset{p = 2^{-128}}{=} 0 \right).$$

   Since $|(p^* = 1) - (p = 2^{-128})| > \epsilon$, Ξ_{AES}^{Square} is useful.
2. Get $P_i = x_i = a_1 \| a_2 \| \ldots \| a_{16}$ for $i \in \{0, 1, \ldots, 256\}$ and corresponding $C_i = AES_K(P_i)$, such that $a_{c_1} = i$ for $c_1 \in const$ and that $a_{j \neq c_1} = c_2 \in const$.
3. Set $h_a = I, h_b = \rho_4$, RK_b = round key of ρ_4.

```
Guess $\forall RK_b^j$ for $j \in \{0, 1, \ldots, 2^k\}$ where $k = |RK_b|$ and do
(i) $\forall C_i$, compute
  $\tilde{y}_i \leftarrow h_b^{-1}(C_i, RK_b^j)$.
(ii) Compute $A = \mathcal{D}(\Xi, \tilde{y}_i)$ defined by:
  If $\Xi(\tilde{y}_i) == 0$, return $A = 1$.
  Otherwise, return $A = 0$.
(iii) Compute $\mathcal{F}(A, RK_b^j, \{RK_b\})$ defined by:
  If $A \neq 1$,
    $\{RK_b\} \leftarrow \{RK_b\} - RK_b^j$.
```

5 Comparisons to Related Work

The commutative diagram cryptanalysis [33]—for compactness of description, we shall denote this as CDC—is a "framework for expressing certain kinds of attacks on product ciphers," of which most block ciphers are. Our Xi framework also shares this same purpose. Further, both our frameworks attempt to unify the common features among different block cipher cryptanalysis techniques in order to obtain a general view of these attacks while comparing the power or uniqueness of each individual one. Both frameworks also exploit the nonrandomness within cipher components in order to mount distinguishing attacks on them. However, while CDC focuses on distinguishing attacks, our framework allows for both this and also extension to key-recovery attacks that exploit such distinguishers. In fact, our framework expresses the relationship between the two in an independent way, i.e., a key-recovery attack is viewed as working regardless of the distinguisher exploited and any specific details.

Another seemingly different but actually very similar (i.e., both use the bottom-up approach) property between our two frameworks is that the CDC identifies local properties of round functions and then pieces them together to obtain a global (distinguishing) property of the whole cipher, while our framework identifies a distinguishing property of some middle rounds (the reduced cipher) of the cipher and then covers the remaining outer rounds by guessing round keys in those latter rounds. Another similar property between the two frameworks is that the CDC considers projections of the cipher's input space and how it relates to projections of the output space, while our framework considers a subset of the reduced cipher's input space and how it relates to a subset of the reduced cipher's output space.

The CDC is mainly different from our framework in that it employs commutative diagrams from abstract algebra to concisely express the functional compositions of product ciphers, while our framework uses descriptions that are more algorithmic in nature and, hence, can easily be translated to source code implementations. Further, our framework is more general and can be used to express any cryptanalysis technique with only a change in the description of the exploited distinguisher. Our framework also allows for expressing the slide, amplified boomerang/rectangle, and Square attacks, the tasks of which were left as open problems in the CDC framework [33].

In fact, viewing the distinguishers exploited by these three attacks in our Xi framework context has allowed us to extend the CDC framework such that it is able to successfully express all three attacks. This has been written up separately [28] and further communicated to the author of CDC. We view this as an extremely useful application of our Xi framework.

6 Conclusion and Open Problems

We have presented a general framework that allows for compact description of block cipher cryptanalysis. This allows one to concentrate on the general structure and essence of cryptanalysis, while also making it possible to emphasize specific details of each individual technique. Using this framework, we can also easily compare different distinguishers used and consider fusions of ideas from each to cause natural generalizations. Viewing them in this light also allows highlighting some interesting future work. It remains an open problem to include more attacks under this framework and also to expand this framework to cater to more features.

Acknowledgments

The authors thank the anonymous referees for constructive comments and suggestions that have improved this paper significantly.

References

[1] K. Aoki, "Practical Evaluation of Security against Generalized Interpolation Attack," *IEICE Trans. Fundamentals,* special section on cryptography and information security, vol. E83-A, no. 1, pp. 33-38, 2000.

[2] E. Biham, "New Types of Cryptanalytic Attacks Using Related Keys," *J. Cryptology,* vol. 7, no. 4, pp. 229-246, 1994.

[3] E. Biham, A. Biryukov, and A. Shamir, "Miss in the Middle Attacks on IDEA, Khufu and Khafre," *Proc. Fast Software Encryption '99,* pp. 124-138, 1999.

[4] E. Biham, O. Dunkelman, and N. Keller, "The Rectangle Attack —Rectangling the Serpent," *Proc. Eurocrypt '01,* pp. 340-357, 2001.

[5] E. Biham, O. Dunkelman, and N. Keller, "Related-Key Boomerang and Rectangle Attacks," *Proc. Eurocrypt '05,* 2005.

[6] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-Like Cryptosystems," *J. Cryptology,* vol. 4, no. 1, pp. 3-72, 1991.

[7] A. Biryukov, "Methods of Cryptanalysis," PhD dissertation, Technion, Israel, 1999.

[8] A. Biryukov and A. Shamir, "Structural Cryptanalysis of SASAS," *Proc. Eurocrypt '01,* pp. 394-405, 2001.

[9] A. Biryukov and D. Wagner, "Slide Attacks," *Proc. Fast Software Encryption '99,* pp. 245-259, 1999.

[10] A. Biryukov and D. Wagner, "Advanced Slide Attacks," *Proc. Eurocrypt '00,* pp. 589-606, 2000.

[11] J. Daemen, L. Knudsen, and V. Rijmen, "The Block Cipher SQUARE," *Proc. Fast Software Encryption '97,* pp. 149-165, 1997.

[12] J. Daemen and V. Rijmen, *The Design of Rijndael—AES—The Advanced Encryption Standard.* Springer-Verlag, 2002.

[13] A.C. Doyle, "The Sign of Four," *Lippincott's Magazine,* Feb. 1890.

[14] S. Furuya, "Slide Attacks with a Known-Plaintext Cryptanalysis," *Proc. Int'l Conf. Information Security and Cryptology (ICISC '01),* pp. 214-225, 2002.

[15] T. Jakobsen and L. Knudsen, "The Interpolation Attack on Block Ciphers," *Proc. Fast Software Encryption '97,* pp. 28-40, 1997.

[16] J. Kelsey, T. Kohno, and B. Schneier, "Amplified Boomerang Attacks against Reduced-Round MARS and Serpent," *Proc. Fast Software Encryption '00,* pp. 75-93, 2000.

[17] L. Knudsen, "Truncated and Higher Order Differentials," *Proc. Fast Software Encryption '94,* pp. 196-211, 1995.

[18] L. Knudsen, "DEAL—A 128-Bit Block Cipher," *AES Submission,* 1998, http://www.ii.uib.no/~larsr/papers/deal.ps.

[19] L. Knudsen, "Block Ciphers—A Survey," *Proc. State of the Art in Applied Cryptography,* pp. 18-48, 1998.

[20] L. Knudsen, "Contemporary Block Ciphers," *Proc. Lectures on Data Security, Modern Cryptology Theory, and Practice,* pp. 105-126, 1999.

[21] L. Knudsen and D. Wagner, "Integral Cryptanalysis," *Proc. Fast Software Encryption '02,* pp. 112-127, 2002.

[22] K. Kurosawa, T. Iwata, and V.D. Quang, "Root Finding Interpolation Attack,"303-314, 2001.

[23] S. Lucks, "Saturation Attacks—A Bait for Twofish," *Proc. Fast Software Encryption '01,* pp. 1-15, 2002.

[24] M. Matsui, "Linear Cryptanalysis Method for DES Cipher," *Proc. Eurocrypt '93,* pp. 386-397, 1993.

[25] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography.* CRC Press, 1997.

[26] S. Moriai, T. Shimoyama, and T. Kaneko, "Interpolation Attacks of the Block Cipher: SNAKE," *Proc. Fast Software Encryption '99,* pp. 275-289, 1999.

[27] NIST, "Advanced Encryption Standard (AES)," Federal Information Processing Standard Publication 197 (FIPS PUB 197), US Dept. of Commerce, Nov. 2001.

[28] R.C.-W. Phan, "Expressing the Slide, Amplified Boomerang, Rectangle and Square Attacks under the Commutative Diagram Cryptanalysis Framework," *Computer Standards & Interfaces,* Elsevier, 2006.

[29] R.C.-W. Phan and S. Furuya, "Sliding Properties of the DES Key Schedule and Potential Extensions to the Slide Attacks," *Proc. Int'l Conf. Information Security and Cryptography (ICISC '02)* pp. 138-148, 2003.

[30] M. Rabin, "Probabilistic Algorithms in Finite Fields," *SIAM J. Computing,* vol. 9, no. 2, pp. 273-280, 1980.

[31] B. Schneier, *Applied Cryptography.* Wiley, 1996.

[32] D. Wagner, "The Boomerang Attack," *Proc. Fast Software Encryption '99,* pp. 156-170, 1999.

[33] D. Wagner, "Towards a Unifying View of Block Cipher Cryptanalysis," *Proc. Fast Software Encryption '04,* invited paper, 2004, http://www.cs.berkeley.edu/~daw/talks/FSE04unify.ps.

Raphael C.-W. Phan (M'03) received the BEng (hons) degree in electronics majoring in computer engineering and the MEngSc and PhD degrees from Multimedia University (MMU), Malaysia, in 1999, 2001, and 2005, respectively. He is currently the director of the Information Security Research (iSECURES) Laboratory at the Swinburne University of Technology (Sarawak Campus), Kuching, Malaysia. His research is on cryptography, cryptanalysis, authentication, and key exchange protocols, smart card security, hash functions, and digital watermarking. His work has been published in refereed journals and in internationally refereed cryptology conferences. He is a referee for several IEEE journals on the area of information security. He is general chair of Mycrypt '05 and Asiacrypt '07, program chair of the International Workshop on Information Security & Hiding (ISH '05), and a technical program committee member of Mycrypt '05, the International Conference on Information Security & Cryptology (ICISC '05), International Conference on Applied Cryptography & Network Security (ACNS '06), International Workshop on Security (IWSEC '06), International Workshop on Information Security Applications (WISA '06), International Conference on the Theory & Application of Cryptology & Information Security (ASIACRYPT '06), International Conference on Information Security & Cryptology (ICISC '06), and ACM Symposium on Information, Computer & Communications Security (ASIACCS '07). He is a member of the IEEE.

Mohammed Umar Siddiqi received the BSc Engg and MSc Engg degrees from Aligarh Muslim University (AMU Aligarh) in 1966 and 1971, respectively, and the PhD degree from the Indian Institute of Technology Kanpur (IIT Kanpur) in 1976, all in electrical engineering. He has been in the teaching profession throughout, first at AMU Aligarh, then at IIT Kanpur and Multimedia University, Malaysia. Currently, he is with the Faculty of Engineering, International Islamic University Malaysia. His research interests are in coding and cryptography. he has published more than 100 papers in international journals and conferences.