International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Quantum Algorithm to Construct Linear Approximation of an S-Box

Ashwini Kumar Malviya, Namita Tiwari

Abstract*: Linear cryptanalysis, a Known-Plaintext Attack, for symmetric block cipher works by constructing linear approximations of the non-linear components of the cipher. The only component which introduces non-linearity in the symmetric block cipher is an S-box. Using classical computing algorithms, the best known solution to find a linear approximation of a non-linear function, in this case an S-box, requires $O(2^n)$ queries to the S-box and $O(2^{2n+m})$ time-complexity, where n is the input size of the S-box and m is the output size. In this paper, a quantum algorithm is presented which can produce best linear approximations of a non-linear S-box using only $O(2^m)$ queries to S-box with $O(n2^m)$ time-complexity. The proposed algorithm shows a significant improvement over the classical algorithm. Correctness proof of the proposed quantum algorithm is presented along with an example.

***Index Terms*: Linear Cryptanalysis, Linear Approximation, Quantum Computing, Quantum Linear Approximation.**

I. INTRODUCTION

Linear cryptanalysis, proposed by Matsui [1,2], is a cryptanalysis technique used to recover secret key used in a symmetric block cipher. The attack assumes that sufficient amount of Plaintext and corresponding Ciphertext are available to recover some of the key-bits by constructing linear approximations of the non-linear components of the symmetric cipher. Thus, the attack is considered as a Known-Plaintext Attack (KPA). The attack can be turned into a Ciphertext-only attack when a linear approximation excluding Plaintext-bits can be constructed. This depends on the structure of a cipher.

Design philosophy for building a strong symmetric block cipher recommends using highly non-linear components [3]. Whereas, S-box is the only component which introduces non-linearity in symmetric primitives. Thus, linear cryptanalysis targets S-box by constructing a set of linear approximations for it [4–7]. Finding a linear approximation for a given S-box requires $O(2^n)$ queries to it along with $O(2^{2n+m})$ time-complexity to construct a linear approximation equation, where n and m represents the input and output size of an S-box in bits, respectively.

In this paper, a quantum algorithm is proposed which finds the best linear approximation of a non-linear S-box using only $O(2^m)$ queries and in $O(n2^m)$ time-complexity.

The rest of the paper is organized as follows. In Section II, mathematical notations of linear cryptanalysis and quantum computing are presented. Some of the quantum transformations related to the proposed work are also explained. Section III presents the proposed quantum algorithm to generate linear approximations of an S-box (non-linear function) given as a quantum oracle. Complexity and correctness proof of the proposed algorithm along with an example to explain the functioning of the propose quantum algorithm are also discussed here. In Section IV, detailed result analysis is presented compared to the classical algorithm. Section V concludes the paper.

II. PRELIMINARIES

A. Linear Cryptanalysis Notations

Let $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ denotes a non-linear function (S-box) which takes an n-bit input and produces an m-bit output. Thus, there are 2^n possible inputs to the function f. Assume $\boldsymbol{x} = (x_1, x_2, \dots, x_n) \in \mathbb{F}_2^n$ represents the input of f, where each $x_i \in \{0,1\}$ and $1 \leq i \leq n$. Similarly, $\boldsymbol{y} = (y_1, y_2, \dots, y_m) \in \mathbb{F}_2^m$ represents the output of f i.e. $f(\boldsymbol{x}) = \boldsymbol{y}$, where each $y_i \in \{0,1\}$ and $1 \leq i \leq m$.

Let $\boldsymbol{u} \in \mathbb{F}_2^n$ and $\boldsymbol{v} \in \mathbb{F}_2^m$ denote the input mask and output mask of f, respectively. The input mask $\boldsymbol{u}$ is applied on the input $\boldsymbol{x}$ of f as $\boldsymbol{u} \cdot \boldsymbol{x} = \bigoplus_{i=1}^{n} (u_i \wedge x_i) = \bigoplus_{i=1}^{n} (u_i x_i)$. It is referred as dot-product of $\boldsymbol{u}$ and $\boldsymbol{x}$. Similarly, the output mask $\boldsymbol{v}$ when applied on output $\boldsymbol{y}$ gives $\boldsymbol{v} \cdot \boldsymbol{y} = \bigoplus_{i=1}^{m} (v_i y_i)$.

The linear approximation for f is represented as $(\boldsymbol{u}, \boldsymbol{v}) = \bigoplus_{i=1}^{n} (u_i x_i) \oplus (\bigoplus_{i=1}^{m} (v_i y_i))$. Assuming that there are a and b number of bits set to 1 in $\boldsymbol{u}$ and $\boldsymbol{v}$, respectively, then the linear approximation of f can be written as shown in (1).

$$(\boldsymbol{u}, \boldsymbol{v}) = u_{i_1} x_{i_1} \oplus u_{i_2} x_{i_2} \oplus \dots \oplus u_{i_a} x_{i_a} \oplus v_{j_1} y_{j_1} \oplus v_{j_2} y_{j_2} \oplus \dots \oplus v_{j_b} y_{j_b} \quad (1)$$

i_k in u_{i_k} of (1) denotes bit-position in $\boldsymbol{u}$ where bit is set to 1. Similarly, j_k in v_{j_k} denotes bit-position in $\boldsymbol{v}$ where bit is set to 1. Equation (1) is simplified and written as shown in (2).

$$(\boldsymbol{u}, \boldsymbol{v}) = x_{i_1} \oplus \dots \oplus x_{i_a} \oplus y_{j_1} \oplus \dots \oplus y_{j_b} \quad (2)$$

A linear approximation is considered good for cryptanalysis when (2) becomes 0 for either more than half of the cases or less than half of the cases. In other words,

Manuscript published on November 30, 2019.
* Correspondence Author
Ashwini Kumar Malviya*, Department of Computer Science and Engineering, Maulana Azad National Institute of Technology, Bhopal 462003, India. Email: ashwinixar@gmail.com
Namita Tiwari, Department of Computer Science and Engineering, Maulana Azad National Institute of Technology, Bhopal 462003, India. Email: namita_tiwari21@rediffmail.com

if for exactly half of the input of function f, (2) results in 0 and for remaining half of the input it results in 1 then this linear approximation is considered bad and is not used in linear cryptanalysis. The problem is to find or construct the good linear approximation for a given S-box.

Using classical algorithm, the number of queries to S-box (considered as an oracle) required to find a good linear approximation is $O(2^n)$ along with the time-complexity of $O(2^{2n+m})$ to build Linear Approximation Table [8]. In this paper, it is assumed that the time-complexity includes the complexity of number of queries made to the oracle. Employing Linear Approximation Table several linear approximations can be found by scanning the whole table.

B. Quantum Computing

The paper follows standard notations of quantum computing as presented in [9,10].

Two unitary transformations are employed in the proposed algorithm. These are explained here. The first is Hadamard transformation (H) which works on a qubit as follows:

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{3}$$

$$H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{4}$$

To apply Hadamard transformation on n-qubit system, $H^{\otimes n}$ is used as shown in (5). $H^{\otimes n}$ implies that H is applied on each n qubit individually.

$$H^{\otimes n}|\boldsymbol{x}\rangle = \frac{1}{\sqrt{2^n}} \sum_{\boldsymbol{u}\in\{0,1\}^n} (-1)^{\boldsymbol{x}\cdot\boldsymbol{u}} |\boldsymbol{u}\rangle \tag{5}$$

The second unitary transformation is used for function evaluation. For a function f, U_f denotes the corresponding unitary transformation which works as shown in (6). In (6), q represents a single qubit and $\boldsymbol{x}$ is a n-qubit string. $\boldsymbol{x}$ and q are inputs to U_f which produces the output $\boldsymbol{x}$ and $q \oplus f(\boldsymbol{x})$, where $f(\boldsymbol{x})$ is a single qubit.

$$U_f|\boldsymbol{x}, q\rangle = |\boldsymbol{x}, q \oplus f(\boldsymbol{x})\rangle \tag{6}$$

This unitary transformation can be modified to produce m-qubit output as shown in (7).

$$U_f|\boldsymbol{x}, \boldsymbol{q}\rangle = |\boldsymbol{x}, \boldsymbol{q} \oplus \boldsymbol{y}\rangle \tag{7}$$

where $\boldsymbol{q}$ and $\boldsymbol{y}$ are m-qubit strings and $\boldsymbol{y}$ is output of the given S-box.

III. PROPOSED QUANTUM ALGORITHM

The proposed quantum algorithm finds a good linear approximation $(\boldsymbol{u}, \boldsymbol{v})$ of a non-linear function (S-box) given as an oracle. It is assumed that oracle can be queried in superposition by the proposed quantum algorithm.

Instead of taking unitary transformation U_f of S-box f, a function $g_{\boldsymbol{v}}: \mathbb{F}_2^n \to \mathbb{F}_2$ is developed as defined in (8).

$$g_{\boldsymbol{v}}(\boldsymbol{x}) = f(\boldsymbol{x}) \cdot \boldsymbol{v} \tag{8}$$

The function $g_{\boldsymbol{v}}$ is defined for a fix value of $\boldsymbol{v} \in \mathbb{F}_2^m$ and produces a single bit output by first evaluating $f(\boldsymbol{x})$ and then performing dot-product of $f(\boldsymbol{x})$ and $\boldsymbol{v}$. The corresponding unitary transformation $U_{g_{\boldsymbol{v}}}$ of $g_{\boldsymbol{v}}$ is shown in (9).

$$U_{g_{\boldsymbol{v}}}|\boldsymbol{x}, q\rangle = |\boldsymbol{x}, q \oplus g_{\boldsymbol{v}}(\boldsymbol{x})\rangle \tag{9}$$

Equation (9) can be re-written as shown in (10). $U_{g_{\boldsymbol{v}}}$ can be modified to U_g such that it takes $(n+m)$-qubit of input where first n-qubit represents $\boldsymbol{x}$ and remaining last m-qubit represents $\boldsymbol{v}$. Thus, this new unitary transformation can work for different values of $\boldsymbol{v}$. However, without loss of generality, $U_{g_{\boldsymbol{v}}}$ is assumed to design the proposed algorithm.

$$U_{g_{\boldsymbol{v}}}|\boldsymbol{x}, q\rangle = |\boldsymbol{x}, q \oplus (f(\boldsymbol{x}) \cdot \boldsymbol{v})\rangle \tag{10}$$

The precise and mathematical description of the proposed quantum algorithm is as follows.

Step 1. Initialize the quantum registers R_1 and R_2 to $|0\rangle^{\otimes n}$ and $|1\rangle$, respectively. Thus, the quantum state $|0\rangle^{\otimes n}|1\rangle$ is obtained.

Step 2. Apply Hadamard transformation $H^{\otimes n}$ and H on R_1 and R_2, respectively. The following quantum state is obtained:

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{\boldsymbol{x}\in\{0,1\}^n} |\boldsymbol{x}\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \tag{11}$$

Step 3. Apply $U_{g_{\boldsymbol{v}}}$ on R_1 and R_2 to get the following quantum state:

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{\boldsymbol{x}\in\{0,1\}^n} (-1)^{g_{\boldsymbol{v}}(\boldsymbol{x})} |\boldsymbol{x}\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \tag{12}$$

Step 4. Apply Hadamard transformation $H^{\otimes n}$ and H on R_1 and R_2, respectively to get the following:

$$|\psi_3\rangle = \frac{1}{2^n} \sum_{\boldsymbol{x}\in\{0,1\}^n} \sum_{\boldsymbol{u}\in\{0,1\}^n} (-1)^{g_{\boldsymbol{v}}(\boldsymbol{x})\oplus\boldsymbol{x}\cdot\boldsymbol{u}} |\boldsymbol{u}\rangle |1\rangle \tag{13}$$

Step 5. Measure the contents of register R_1 to get the value of $\boldsymbol{u}$.

The quantum circuit for $U_{g_{\boldsymbol{v}}}$ and U_g can be designed using $O(n)$ and $O(n+m)$ quantum logic gates, respectively. This is called the quantum circuit complexity. Basically, the quantum circuit complexity to implement a unitary transformation is regarded as the time-complexity to evaluate a function which the unitary transformation represents. The above mentioned algorithm finds the value of $\boldsymbol{u}$ corresponding to a fixed value of $\boldsymbol{v}$ in constant query-complexity (only 1 quantum query to an oracle

is required) and $O(n)$ time-complexity (due to the quantum circuit complexity of U_{g_v}). Thus, a good linear approximation $(\boldsymbol{u}, \boldsymbol{v})$ is obtained for a given non-linear S-box. The proposed algorithm must be repeated for all $\boldsymbol{v} \in \mathbb{F}_2^m$.

Since, there are 2^m possible values of $\boldsymbol{v}$ for which the proposed algorithm must be repeated, the query complexity and the time-complexity becomes $O(2^m)$ and $O(n2^m)$, respectively. Compared to the query and time complexities of the classical algorithm, it is a significant improvement.

Correctness of the proposed quantum algorithm is presented next. Since, the algorithm linearly approximates a non-linear function, the output is generated with a probability which is less than 1. Thus, the proposed algorithm must be repeated for a constant number of times to get the most appropriate value of $\boldsymbol{u}$. Other, values of $\boldsymbol{u}$ with different probability can be measured and stored for constructing a linear trail for use in linear cryptanalysis. Equation (13) is re-written as shown in (14).

$$|\psi_3\rangle = \frac{1}{2^n} \sum_{\boldsymbol{u}\in\{0,1\}^n} \sum_{\boldsymbol{x}\in\{0,1\}^n} (-1)^{f(\boldsymbol{x})\cdot\boldsymbol{v}\oplus\boldsymbol{x}\cdot\boldsymbol{u}} |\boldsymbol{u}\rangle |1\rangle \qquad (14)$$

The relation $(-1)^{f(\boldsymbol{x})\cdot\boldsymbol{v}\oplus\boldsymbol{x}\cdot\boldsymbol{u}}$ evaluates to 1 when $f(\boldsymbol{x})\cdot\boldsymbol{v}\oplus\boldsymbol{x}\cdot\boldsymbol{u}$ is 0, otherwise $(-1)^{f(\boldsymbol{x})\cdot\boldsymbol{v}\oplus\boldsymbol{x}\cdot\boldsymbol{u}} = -1$. Assuming for a fixed $\boldsymbol{u}$, if $(-1)^{f(\boldsymbol{x})\cdot\boldsymbol{v}\oplus\boldsymbol{x}\cdot\boldsymbol{u}}$ is 1 for half of the values of $\boldsymbol{x}$ and -1 for the remaining half then the relation $\sum_{\boldsymbol{x}\in\{0,1\}^n}(-1)^{f(\boldsymbol{x})\cdot\boldsymbol{v}\oplus\boldsymbol{x}\cdot\boldsymbol{u}}$ becomes 0 and thus the probability of measuring such a $\boldsymbol{u}$ also becomes 0. This shows that no bad linear approximation can be generated by the proposed algorithm.

Considering a fixed value of $\boldsymbol{u}$, assume $C_0(\boldsymbol{u}) = \#\{\boldsymbol{x}|f(\boldsymbol{x})\cdot\boldsymbol{v}\oplus\boldsymbol{x}\cdot\boldsymbol{u} = \boldsymbol{0}\}$ denotes the number of different values of $\boldsymbol{x}$ for which $f(\boldsymbol{x})\cdot\boldsymbol{v}\oplus\boldsymbol{x}\cdot\boldsymbol{u}$ becomes 0 and consequently in turn $(-1)^{f(\boldsymbol{x})\cdot\boldsymbol{v}\oplus\boldsymbol{x}\cdot\boldsymbol{u}} = 1$. Similarly, $C_1(\boldsymbol{u}) = \#\{\boldsymbol{x}|f(\boldsymbol{x})\cdot\boldsymbol{v}\oplus\boldsymbol{x}\cdot\boldsymbol{u} = \boldsymbol{1}\}$ denotes the number of different values of $\boldsymbol{x}$ for which $f(\boldsymbol{x})\cdot\boldsymbol{v}\oplus\boldsymbol{x}\cdot\boldsymbol{u}$ becomes 1 and thus $(-1)^{f(\boldsymbol{x})\cdot\boldsymbol{v}\oplus\boldsymbol{x}\cdot\boldsymbol{u}} = -1$ holds. Therefore, $C_0(\boldsymbol{u}) + C_1(\boldsymbol{u}) = 2^n$. So, the value of $\boldsymbol{u}$ which can be measured with highest probability has maximum absolute value for the relation shown in (15).

$$|C_0(\boldsymbol{u}) - C_1(\boldsymbol{u})| \qquad (15)$$

After performing measurement, any value of $\boldsymbol{u}$ which is measured has the probability shown in (16).

$$\left(\frac{|C_0(\boldsymbol{u}) - C_1(\boldsymbol{u})|}{2^n}\right)^2 \qquad (16)$$

From (16), it can be inferred that the value of $\boldsymbol{u}$ for which $|C_0(\boldsymbol{u}) - C_1(\boldsymbol{u})|$ is maximum has the highest probability. Hence, when the proposed quantum algorithm is repeated a constant number of times, the value of $\boldsymbol{u}$ that will be measured majority number of times is the part of best linear approximation $(\boldsymbol{u}, \boldsymbol{v})$, for a fixed value of $\boldsymbol{v}$. If $C_0(\boldsymbol{u}) - C_1(\boldsymbol{u})$ is a positive number then $(\boldsymbol{u}, \boldsymbol{v})$ corresponds to the linear approximation, else $(\boldsymbol{u}, \boldsymbol{v})$ corresponds to the affine approximation. In case of affine approximation, apply $(\boldsymbol{u}, \boldsymbol{v})$ as shown in (2) and then XOR the result with 1 to get the linear approximation. This completes the proof of correctness of the proposed quantum algorithm.

An example is presented to explain the working of proposed algorithm. The S-box used in this example is from [8] and is described in Table I. Considered S-box has both input and output size of 4-bits, therefore $n = m = 4$.

For a fixed $\boldsymbol{v} = 0111$, the algorithm finds a value of $\boldsymbol{u}$ which contributes to the best linear/affine approximation $(\boldsymbol{u}, \boldsymbol{v})$ of the given S-box. After applying step 1 and 2 of the proposed algorithm, $|\psi_1\rangle$ is the resultant quantum state represented as follows:

$$|\psi_1\rangle = \frac{1}{\sqrt{2^4}}(|0000\rangle + |0001\rangle + |0010\rangle + |0011\rangle + |0100\rangle + |0101\rangle + |0110\rangle + |0111\rangle + |1000\rangle + |1001\rangle + |1010\rangle + |1011\rangle + |1100\rangle + |1101\rangle + |1110\rangle + |1111\rangle)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Table - I: S-box description

Input[a]	Output[a]	Input[a]	Output[a]
0000	1110	1000	0011
0001	0100	1001	1010
0010	1101	1010	0110
0011	0001	1011	1100
0100	0010	1100	0101
0101	1111	1101	1001
0110	1011	1110	0000
0111	1000	1111	0111

[a] All values are in binary.

Applying $U_{g_{v=0111}}$ on $|\psi_1\rangle$ to get $|\psi_2\rangle$ as shown below.

$$|\psi_2\rangle = \frac{1}{\sqrt{2^4}}((-1)^0|0000\rangle + (-1)^1|0001\rangle + (-1)^0|0010\rangle + (-1)^1|0011\rangle + (-1)^1|0100\rangle + (-1)^1|0101\rangle + (-1)^0|0110\rangle + (-1)^0|0111\rangle + (-1)^1|1000\rangle + (-1)^1|1001\rangle + (-1)^0|1010\rangle + (-1)^1|1011\rangle + (-1)^0|1100\rangle + (-1)^1|1101\rangle + (-1)^0|1110\rangle + (-1)^1|1111\rangle)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Performing Hadamard transformation (Step 4 of the algorithm) on $|\psi_2\rangle$ and evaluating $(-1)^{g_{0111}(\boldsymbol{x})\oplus\boldsymbol{x}\cdot\boldsymbol{u}}$ on all $\boldsymbol{x}$ corresponding for every $\boldsymbol{u}$ gives $|\psi_3\rangle$ as follows:

$$|\psi_3\rangle = \frac{1}{2^4}(0|0000\rangle + 12|0001\rangle - 4|0010\rangle + 0|0011\rangle + 0|0100\rangle + 4|0101\rangle + 4|0110\rangle + 0|0111\rangle + 0|1000\rangle - 4|1001\rangle - 4|1010\rangle + 0|1011\rangle + 0|1100\rangle + 4|1101\rangle + 4|1110\rangle + 0|1111\rangle)|1\rangle$$

Measuring quantum register R_1 results in a value of $\boldsymbol{u}$ based on its probability.

Table II shows the probability distribution of measuring different values of $\boldsymbol{u}$ which is immediate from quantum state $|\psi_3\rangle$.

From Table II, it is observed that $\boldsymbol{u} = 0001$ corresponding to $\boldsymbol{v} = 0111$, which has the highest probability, best approximates the S-box and this result coincides with the result presented in [8], thus strengthening the claim of correctness of the proposed work.

Table - II: Probability Distribution of $\boldsymbol{u}$ in $|\psi_3\rangle$

u	Prob.	u	Prob.
0000	0	1000	0
0001	$(12/16)^2$	1001	$(-4/16)^2$
0010	$(-4/16)^2$	1010	$(-4/16)^2$
0011	0	1011	0
0100	0	1100	0
0101	$(4/16)^2$	1101	$(4/16)^2$
0110	$(4/16)^2$	1110	$(4/16)^2$
0111	0	1111	0

IV. RESULT ANALYSIS

As per the results presented in [8], it can be noted that for $\boldsymbol{u} = 0001$ and $\boldsymbol{v} = 0111$, the number of input-output pairs, $(\boldsymbol{x}, \boldsymbol{y})$, that holds as in (1) are 14 out of 16 possible cases. This can be used to denote the extent of linear approximation of the given $(\boldsymbol{u}, \boldsymbol{v})$ as shown below in (17):

$$\frac{L_{\boldsymbol{u},\boldsymbol{v}}}{2^n} = \frac{C_0(\boldsymbol{u})}{2^n} \tag{17}$$

where, $L_{\boldsymbol{u},\boldsymbol{v}}$ denotes the total number of pairs, $(\boldsymbol{x}, \boldsymbol{y})$, for which the mask $(\boldsymbol{u}, \boldsymbol{v})$ holds as per the relation in (1) i.e. $L_{\boldsymbol{u},\boldsymbol{v}} = C_0(\boldsymbol{u})$. 2^n denotes the total number of input-output pairs, $(\boldsymbol{x}, \boldsymbol{y})$, possible for a given S-box of an input size of n-bits. This relation in (17) can very well be used to denote the goodness of a linear approximation. Also, the ideal linear approximation is the one for which (17) becomes 1 i.e. $L_{\boldsymbol{u},\boldsymbol{v}} = 2^n$. Thus, for the results in [8], it is immediate that for $(\boldsymbol{u} = 0001, \boldsymbol{v} = 0111)$, (17) becomes $\frac{14}{16}$.

However, from quantum state $|\psi_3\rangle$, it can be noted that for $\boldsymbol{u} = 0001$, the corresponding coefficient is $\frac{12}{16}$ which does not coincide with the result presented in [8]. The reason for this is that in the presented quantum algorithm, when squared, this coefficient denotes the probability to measure a particular $\boldsymbol{u}$ and does not denotes the goodness of the linear approximation. Moreover, the coefficient is derived by taking the ratio of subtraction between $C_0(\boldsymbol{u})$ and $C_1(\boldsymbol{u})$ with 2^n. Thus, (18) shows the relation for deriving the coefficient for a given $\boldsymbol{u}$. From (18), it can be seen that the number of pairs, denoted as $C_1(\boldsymbol{u})$, which cannot be linearly approximated with $(\boldsymbol{u}, \boldsymbol{v})$ reduces the coefficient of $\boldsymbol{u}$ in quantum state $|\psi_3\rangle$ by twice which reduces the overall probability to measure the desired value of $\boldsymbol{u}$. Therefore, it becomes necessary to repeat the proposed quantum algorithm for a constant number of times and then take the value of $\boldsymbol{u}$ which is measured majority of the times. This approach produces the desired value of $\boldsymbol{u}$.

$$\frac{|C_0(\boldsymbol{u}) - C_1(\boldsymbol{u})|}{2^n} = \frac{L_{\boldsymbol{u},\boldsymbol{v}}}{2^n} - \frac{C_1(\boldsymbol{u})}{2^n} \tag{18}$$

V. CONCLUSION

A quantum algorithm to linearly approximate a non-linear function was proposed which has a significant improvement over the classical solution in terms of query and time complexity. The proposed quantum algorithm finds the best linear approximation using only $O(2^m)$ queries and in $O(n2^m)$ time-complexity. This work extends the cryptanalysis toolbox and may be used for designing better symmetric primitives. It is also noticed that the proposed algorithm is similar in design when compared to the Bernstein-Vazirani algorithm which is used for finding linear structure of a given Boolean function. Since, the proposed quantum algorithm finds only one solution, future research may include investigations about constructing several linear approximations and concentrate on attaining better speedup.

REFERENCES

1. M. Matsui, A. Yamagishi, "A New Method for Known Plaintext Attack of FEAL Cipher," *EUROCRYPT 1992*, Hungary, 1992, pp. 81–91.
2. M. Matsui, "Linear Cryptanalysis Method for DES Cipher," *EUROCRYPT 1993*, Norway, 1993, pp. 386–397.
3. E. W. Tischhauser, "Mathematical aspects of symmetric-key cryptography," Ph.D. Dissertation, Katholieke Universiteit Leuven, Belgium, 2012.
4. J. Y. Cho, "Linear Cryptanalysis of Reduced-Round PRESENT," *The RSA Conference 2010*, San Francisco, USA, 2010, pp. 302–317.
5. W. Wu, "Differential-Linear Cryptanalysis of Camellia," *Progress on Cryptography. The International Series in Engineering and Computer Science*, vol 769. Springer, Boston, MA, USA, 2004, pp. 173–180.
6. J. Ren, S. Chen, "Cryptanalysis of Reduced-Round SPECK," *IEEE Access*, vol. 7, 2019, pp. 63045–63056.
7. W. Yi and S. Chen, "Multidimensional zero-correlation linear cryptanalysis of the block cipher KASUMI," *IET Information Security*, vol. 10, no. 4, 2016, pp. 215–221.
8. H. Heys, "A tutorial on linear and differential cryptanalysis," *Cryptologia*, vol. 26, 2002, pp. 189–221.
9. M. A. Nielsen and I. L. Chuang. (2010) Quantum Computation and Quantum Information (10th Anniversary ed.), Cambridge University Press, New York, USA.
10. B. Valiron, "Qauntum Computation: a Tutorial," New Generation Computing, vol. 30, 2012, pp. 271–296.