A quantum algorithm to verify the Strict Avalanche criterion in Boolean functions (Extended abstract)

C. A. Jothishwaran[1], Vishvendra Singh Poonia[1]
Pantelimon Stănică[3], Sugata Gangopadhyay[2]

[1] Department of Electronics and Communication Engineering
[2] Department of Computer Science and Engineering,
Indian Institute of Technology Roorkee, Roorkee 247667, INDIA
{jc_a, vishvendra}@ece.iitr.ac.in, sugata.gangopadhyay@cs.iitr.ac.in
[3] Department of Applied Mathematics
Naval Postgraduate School, Monterey, CA 93943–5216, USA
pstanica@nps.edu

Abstract. We propose a quantum algorithm that verifies that a given Boolean function, in form of a quantum oracle satisfies the "strict avalanche criterion" (SAC). The complexity of our algorithm is of order $\mathcal{O}(n^2)$ compared to the $\mathcal{O}(n2^n)$ complexity for the classical environment.

Keywords: Boolean functions, Fourier spectrum, strict avalanche criterion, quantum algorithms

1 Introduction

An n-variable Boolean function F is a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. The set of all such functions is denoted by $\mathfrak{B}_n$. We associate to each function $F \in \mathfrak{B}_n$ its character form $f : \mathbb{F}_2^n \to \mathbb{R}$, defined by $f(x) = (-1)^{F(x)}$, for all $x \in \mathbb{F}_2^n$ (we use capitals letters for classical Boolean functions and lower cases for the signatures of such). In this article, abusing notation, we refer to the character forms f as Boolean functions and go to the extent of writing $f \in \mathfrak{B}_n$, whenever $F \in \mathfrak{B}_n$, surely, if there is no danger of confusion. For any $x, y \in \mathbb{F}_2^n$, the inner product is $x \cdot y = \sum_{i=1}^n x_i y_i$, where the sum is over $\mathbb{F}_2$. The (Hamming) weight of a vector $u = (u_1, \ldots, u_n) \in \mathbb{F}_2^n$ is $\mathrm{wt}(u) = \sum_{i=1}^n u_i$, where the sum is over $\mathbb{Z}$. The weight of a Boolean function $F \in \mathfrak{B}_n$, or equivalently $f \in \mathfrak{B}_n$ is the cardinality $\mathrm{wt}(F) = \left|\{x \in \mathbb{F}_2^n : F(x) \neq 0\}\right|$, or equivalently $\mathrm{wt}(f) = \left|\{x \in \mathbb{F}_2^n : f(x) \neq 1\}\right|$. We define the Fourier coefficient of f at $u \in \mathbb{F}_2^n$ by $\widehat{f}(u) = 2^{-n} \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{u \cdot x}$. Recall the well known Parseval's identity $\sum_{u \in \mathbb{F}_2^n} \widehat{f}(u)^2 = 1$. The derivative of $f \in \mathfrak{B}_n$ at $c \in \mathbb{F}_2^n$ is the function $\Delta_c f(x) = f(x)f(x+c)$ (or, equivalently, $\Delta_c F(x) = F(x+a) + F(x)$) for all $x \in \mathbb{F}_2^n$.

2 Strict Avalanche Criterion

A Boolean function $f \in \mathfrak{B}_n$ satisfies the strict avalanche criterion (SAC) if the probability of the function changing its value when a single input value is flipped is exactly 0.5 that is, the derivative $\Delta_c F(x)$ is a balanced function for all $c \in \mathbb{F}_2^n$ such that $\mathrm{wt}(c) = 1$. We refer to Budaghyan [2], Carlet [3], and Cusick and Stănică [4] for detailed discussions on SAC and other cryptographic properties of Boolean functions. The Fourier transform of the function $\widehat{f}(w)$ satisfies the following relation

$$\sum_{w \in \mathbb{F}_2^n} (-1)^{w \cdot c}\, \widehat{f}(w)^2 \,=\, 0, \text{ for all } c \in \mathbb{F}_2^n \text{ such that } \mathrm{wt}(c) = 1. \tag{1}$$

As there are n possible strings c of weight 1, the above expression represents n different relations simultaneously satisfied by $\widehat{f}(w)$. These relations can also be written in terms of w_i, the i^{th} bit of w as follows:

$$\begin{aligned} \sum_{w\in\mathbb{F}_2^n} (-1)^{w_i}\, \widehat{f}(w)^2 &= 0, \text{ for all } i \in \mathbb{Z}^+, 1 \le i \le n, \\ \text{i.e.,} \quad \sum_{w\in\mathbb{F}_2^n\,|\,w_i=0} \widehat{f}(w)^2 &= \sum_{w\in\mathbb{F}_2^n\,|\,w_i=1} \widehat{f}(w)^2 = \frac{1}{2}, \text{ for all } i \in \mathbb{Z}^+, 1 \le i \le n. \end{aligned} \tag{2}$$

The last relation is obtained by using Parseval's identity. Classically verifying SAC for a Boolean function has a time complexity of $\mathcal{O}(n2^n)$.

3 Quantum information: definitions and notation

In this section, we will introduce some notation that we use throughout the paper. For an introduction to quantum computing, we refer to Rieffel and Polak [9], or Nielsen and Chuang [8]. The fundamental unit of quantum information is called a qubit. The states of a qubit is denoted by $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ where $a, b \in \mathbb{C}$ such that $|a|^2 + |b|^2 = 1$. If we measure the qubit $|\psi\rangle$ using the standard basis $\{|0\rangle, |1\rangle\}$ the probabilities of observing $|0\rangle$ and $|1\rangle$ are $|a|^2$ and $|b|^2$, respectively.

In the following, we will use the conventional notation $|a\rangle\,|b\rangle := |a\rangle \otimes |b\rangle$, or $|ab\rangle := |a\rangle \otimes |b\rangle$. A state on n qubits can be represented as a $\mathbb{C}$-linear combination of the vectors of the standard basis $|\psi\rangle = \sum_{x\in\mathbb{F}_2^n} a_x\,|x\rangle$, where $a_x \in \mathbb{C}$, for all $x \in \mathbb{F}_2^n$, and $\sum_{x\in\mathbb{F}_2^n} |a_x|^2 = 1$; the set of vectors $|x\rangle$ forms a basis for the n qubit states and is referred to as the computational basis. Let $|0_n\rangle$ be the quantum state associated with the zero vector in $\mathbb{F}_2^n$. The vectors $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$ define the Hadamard basis for single qubit states.

Any Boolean function $F \in \mathfrak{B}_n$ can be implemented as a bit oracle implementation U_F, so that:

$$|x\rangle\,|\varepsilon\rangle \xrightarrow{U_F} |x\rangle\,|\varepsilon + F(x)\rangle\,. \tag{3}$$

Here, $x \in \mathbb{F}_2^n$ and $\varepsilon \in \mathbb{F}_2$. If the target qubit for U_F is $|-\rangle$, then $|x\rangle\,|-\rangle \xrightarrow{U_F} (-1)^{F(x)}\,|x\rangle\,|-\rangle$. This gives an alternative implementation of a Boolean function oracle known as the phase oracle implementation of the function F.

The number of initial qubits to a quantum oracle of Boolean function in $\mathfrak{B}_n$ is $n+1$, any internal ancillary qubits used are not included in this count. If the i^{th} input qubit acts as the control qubit and the target qubit is the same as the target of the oracle, the gate is represented by CZ^i. In the phase oracle representation, the target qubit is in the $|-\rangle$ state the action of CZ^i is as follows,

$$|x\rangle\,|-\rangle \xrightarrow{CZ^i} |x\rangle \left(\tfrac{|0\rangle-(-1)^{x_i}|1\rangle}{\sqrt{2}}\right). \tag{4}$$

This implies if the $x_i = 1$ the target qubit turns from $|-\rangle$ to $|+\rangle$.

Let $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ be the 2×2 identity matrix, and $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ be the 2×2 Hadamard matrix. The tensor product of matrices is denoted by $\otimes$. The matrix H_n is recursively defined as:

$$\begin{aligned} H_2 &= H \otimes H, \\ H_n &= H \otimes H_{n-1}, \text{ for all } n \ge 3. \end{aligned} \tag{5}$$

Note that, for $x \in \mathbb{F}_2^n$, $H_n\,|x\rangle = 2^{\frac{-n}{2}} \sum_{x'\in\mathbb{F}_2^n} (-1)^{x\cdot x'}\,|x'\rangle$.

4 Quantum Algorithm to verify the strict avalanche criterion

The SAC can be verified for a function $F \in \mathfrak{B}_n$ through the following quantum algorithm. The initial state of the $n+1$ qubits is $|0_n\rangle\,|0\rangle$. In the following expressions, the symbol $(\circ)$ is used to represent matrix multiplication:

$$\begin{aligned}
|0_n\rangle\,|0\rangle \;&\xrightarrow{H_n \otimes (Z \circ H)} 2^{-n/2} \sum_{x \in \mathbb{F}_2^n} |x\rangle\,|-\rangle \\
&\xrightarrow{U_F \otimes I} 2^{-n/2} \sum_{x \in \mathbb{F}_2^n} f(x)\,|x\rangle\,|-\rangle \\
&\xrightarrow{H_n \otimes I} 2^{-n} \sum_{x \in \mathbb{F}_2^n} f(x) \sum_{w \in \mathbb{F}_2^n} (-1)^{x \cdot w}\,|w\rangle\,|-\rangle \\
&\equiv 2^{-n} \sum_{w \in \mathbb{F}_2^n} \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{x \cdot w}\,|w\rangle\,|-\rangle \;=\; \sum_{w \in \mathbb{F}_2^n} \widehat{f}(w)\,|w\rangle\,|-\rangle \qquad (6) \\
&\xrightarrow{CZ^i} \sum_{w \in \mathbb{F}_2^n} \widehat{f}(w)\,|w\rangle \left(\frac{|0\rangle - (-1)^{w_i}\,|1\rangle}{\sqrt{2}} \right) \\
&\equiv \sum_{w \in \mathbb{F}_2^n \,|\, w_i = 1} \widehat{f}(w)\,|w\rangle\,|+\rangle \;+\; \sum_{w \in \mathbb{F}_2^n \,|\, w_i = 0} \widehat{f}(w)\,|w\rangle\,|-\rangle \\
&\xrightarrow{I_n \otimes H} \sum_{w \in \mathbb{F}_2^n \,|\, w_i = 1} \widehat{f}(w)\,|w\rangle\,|0\rangle \;+\; \sum_{w \in \mathbb{F}_2^n \,|\, w_i = 0} \widehat{f}(w)\,|w\rangle\,|1\rangle\,.
\end{aligned}$$

The probability that the outcome of a standard measurement on the target qubit $|\epsilon\rangle$ will yield $|0\rangle$ or $|1\rangle$ is given by:

$$\begin{aligned}
\Pr[\epsilon = 0] \;&=\; \sum_{w \in \mathbb{F}_2^n \,|\, w_i = 1} \widehat{f}(w)^2, \\
\Pr[\epsilon = 1] \;&=\; \sum_{w \in \mathbb{F}_2^n \,|\, w_i = 0} \widehat{f}(w)^2,
\end{aligned} \qquad (7)$$

for a function satisfying SAC these two probabilities are $\frac{1}{2}$ and the expectation value $\langle\epsilon\rangle$ is also $\frac{1}{2}$. it should be noted that the same algorithm must be repeated n times with a different gate

$$CZ^i, \text{ for all } i \in \mathbb{Z}^+ \text{ such that } 1 \le i \le n,$$

and yield the same value for $\langle\epsilon\rangle$

Let $\left|\varepsilon^{(i)}\right\rangle$ $\forall\, i \in \mathbb{Z}^+ : 1 \le i \le n$ denote the state of the target qubit after the CZ^i gate is used in the quantum algorithm. Therefore, the probability distribution of the associated random variable $\varepsilon^{(i)}$ is the same as given in (7). We prove the following theorem.

Theorem 1. *Let the probability* $\Pr\left[\varepsilon^{(i)} = 1\right] = p$*, and the result of performing* m *trials of the algorithm* (6) *are represented by the random variables* $\varepsilon_k^{(i)}$*, for all* $k \in \mathbb{Z}^+, 1 \le k \le m$*. Consider the sample mean given by:*

$$X^i \;=\; \frac{1}{m} \sum_{k=1}^{m} \varepsilon_k^{(i)}. \qquad (8)$$

Then for a given margin of error $t > 0$ *we have*

$$\Pr\left[X^i - t \le p_i \le X^i + t\right] \ge 1 - 2\,\exp\!\left(-2mt^2\right).$$

The number of trials (m), required for each iteration in order to estimate p_i is determined by the degree of uncertainty $\delta : \delta \in [0, 1]$ and the margin of error t. We have

$$\Pr\Big[X^i - t \leq p_i \leq X^i + t\Big] \equiv 1 - \delta \geq 1 - 2\,\exp\Big(-2mt^2\Big),$$

which renders $m = \frac{1}{2t^2} \ln\big(\frac{2}{\delta}\big)$.
For an uncertainty of 5% and a margin of error $t = 0.05$ the minimum number of trials comes out to be 738. It can be observed that this value of m is dependent only on δ and t and is independent of the size of the Boolean function n.
Combining this with the number of iterations required to verify the strict avalanche criteria which is n, the overall time complexity of the quantum algorithm to verify the strict avalanche criteria is $\mathcal{O}(n^2)$. This is faster compared to the classical algorithm which has a complexity of $\mathcal{O}(n2^n)$ for large values of the input size n.

5 Alternative quantum algorithms for verifying SAC

The strict avalanche criterion can be verified using quantum algorithms related to other Boolean function properties. One algorithm utilizes the fact that $\Delta_c F(x)$ is balanced for all $c \in \mathbb{F}_2^n$ such that $\mathrm{wt}(c) = 1$. This implies the Fourier coefficient of these derivatives at the point $w = 0_n$ is zero. The character form of the derivative $\Delta_c F(x)$ is defined as $f_c'(x)$, the Fourier coefficient can be represented as $\widehat{g}_c(w)$. Therefore, if f satisfies SAC, then $\widehat{f_c'}(0_n) = 0 \; \forall \; c \in \mathbb{F}_2^n \, : \, \mathrm{wt}(c) = 1$.
If the quantum oracle U_f exists, then the oracle for any of the derivatives $U_{f_c'}$ can be constructed using two such oracles and an additional X gate. The algorithm for verifying SAC is equivalent to Deutsch-Jozsa algorithm [6] on $U_{f_c'}$. The resultant state will not contain the vector $|0_n\rangle$ and therefore measurement of this resultant state will never yield the outcome 0_n. SAC can be verified by repeating this algorithm for each derivative $f_c'(x)$.
The Forrelation problem defined by Aaronson et al. [1] can be used to analyze a number of Boolean function properties. Aaronson et al. [1] also gave a quantum algorithm that evaluates the Forrelation between Boolean functions given their quantum oracles. The $k-$fold Forrelation ($\varPhi$) between k Boolean functions $f_1, f_2 \ldots f_k$ is given by

$$\varPhi_{f_1, f_2 \cdots f_k} = \frac{1}{2^{(k+1)n/2}} \sum_{x_1, x_2 \ldots x_k \in \mathbb{F}_2^n} f_1(x_1)(-1)^{x_1 \cdot x_2} f_2(x_2) \ldots (-1)^{x_{k-1} \cdot x_k} f_k x_k. \qquad (9)$$

The quantum algorithm evaluating this quantity calls each of the Boolean function oracles once.
Datta et al. [5, Lemma 3] used the quantum algorithm for 3-fold Forrelation to verify the $m-$resilience of a general Boolean function $g(x)$. This algorithm evaluates 3-fold Forrelation $\varPhi_{g,h,g}$ where $h(x)$ is a symmetric Boolean function such that $h(x) = 1 \; \forall \; x \in \mathbb{F}_2^n \, : \, wt(x) > m$. If $g(x)$ is $m-$resilient, the final state of the algorithm is always $|0_n\rangle$.
The balanced nature of $f_c'(x)$ also implies that it can be considered to be $0-$resilient. This can be verified setting $g = f_c'$ and using the appropriate $h(x)$ which in this case is just the disjunction between the input variables and the oracle of function can be constructed using X and CX gates.
SAC can therefore be verified by running the algorithm for each of the derivative function. It can be seen that the resultant measurement of the Forrelation based algorithm is the complement of the approach based on the Deutsch-Josza algorithm.

5.1 Comparison of the different quantum algorithms

The algorithm introduced here gives a probabilistic verification of SAC that calls the Boolean function oracle U_f only once per instance of the algorithm. The first alternative approach based on Deutsch-Jozsa algorithm requires the oracle of the derivative $U_{f'_c}$ which in turn requires two calls to the oracle U_f. The approach utilising Forrelation calls the derivative oracle twice per instance, this implies U_f is called four times per instance. Therefore (6) requires the fewest number of gates per instance.

The chief distinction between the algorithm introduced here and the alternative quantum algorithms is in how the result is presented or read-out form the resultant state. In case of the alternatives, the result of the algorithm is presented as the probability of the $|0_n\rangle$ state. In the first alternative case this probability is 0 and in the second case, it is 1. The verification of SAC is done by estimating the probability of a particular outcome of the measurement on n qubits.

The algorithm presented in (6) presents the result in target qubit and verification of SAC is equivalent to the estimation of the expectation value of the measurement on a single qubit. This operation is simpler than the readout required for alternatives where n-qubits are measured. the estimation of the expectation value can be performed using the Hoeffding inequality as shown in Theorem 1.

It should be noted that while Forrelation based algorithms can be used to study a wider variety of properties. The algorithm in (6) is an extension of the Deutsch-Jozsa algorithm with a particular focus on verifying SAC and does so through the inclusion of only one additional gate.

6 Conclusion

The quantum algorithm shown here give a probabilistic verification of the strict avalanche criterion. the number of gates required grows linearly with n, the size of the Boolean function and the oracle U_f is called only once per instance. The accuracy of the verification is independent of n and depends on the number of trials performed. The algorithm uses fewer gates and calls to the oracle U_f than the alternate algorithms. It also requires the measurement outcome from only one of the qubits.

Acknowledgement: Research of C. A. Jothishwaran, Vishvendra Singh Poonia, and Sugata Gangopadhyay is a part of the project "Design and Development of Quantum Computing Toolkit and Capacity Building" sponsored by the Ministry of Electronics and Information Technology (MeitY) of the Government of India.

References

1. Aaronson, S., Ambainis, A. : Forrelation: A Problem that Optimally Separates Quantum from Classical Computing, Siam J. Comput., vol. 47, no.3, pp. 982–1038, 2018. In Proceedings of the forty-seventh annual ACM symposium on Theory of Computing (STOC '15). Association for Computing Machinery, New York, NY, USA, pp. 307–316. DOI: https://doi.org/10.1145/2746539.2746547, 2015.
2. Budaghyan, L.: Construction and Analysis of Cryptographic Functions, Springer-Verlag, 2014.
3. C. Carlet, *Boolean Functions for Cryptography and Coding Theory.* Cambridge: Cambridge University Press, Cambridge, 2021.
4. Cusick, T., Stănică, P.: Cryptographic Boolean Functions and Applications, 2nd Edition, Elsevier (2017).
5. Datta, S., Maitra, S., Mukherjee, C.S. : Following Forrelation – Quantum Algorithms in Exploring Boolean Functions' Spectra, https://arxiv.org/abs/2104.12212.
6. Deutsch, D., Jozsa R.: Rapid solution of problems by quantum computation, Proc. R. Soc. Lond. A439553–558 (1992) http://doi.org/10.1098/rspa.1992.0167
7. Hoeffding, W.: Probability inequalities for sums of bounded random variables, J. American Statistical Association 58 (1963), 13–30.
8. Nielsen M., Chuang I.: Quantum Computation and Quantum Information (10th ed.), Cambridge: Cambridge University Press, 2011.
9. Rieffel, E., Polak, W.: Quantum Computing: A Gentle Introduction, The MIT Press, 1st ed. (2011).