Testing the Strength of Quantum Random Number Generators

A thesis submitted in partial fulfilment of
the requirements for the degree of

Doctor of Philosophy

by

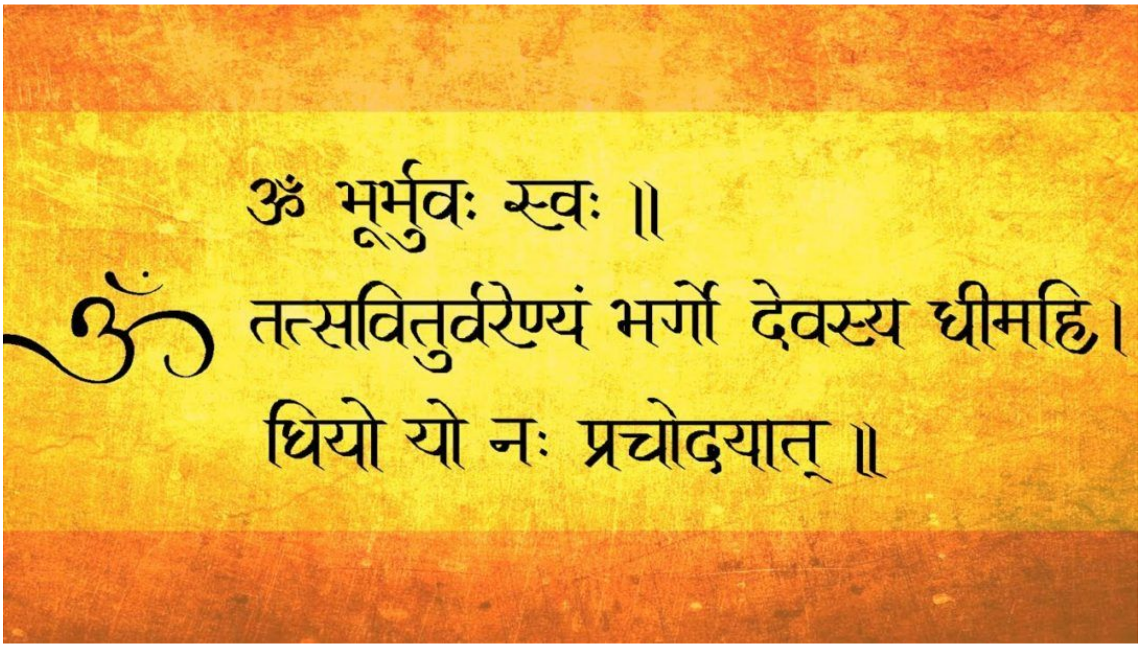
Vardaan Mongia

(Roll No. 19330020)

Under the supervision of

Prof. R. P. Singh
Professor
Atomic, Molecular and Optical Physics Division
Physical Research Laboratory, Ahmedabad, India

DISCIPLINE OF PHYSICS
INDIAN INSTITUTE OF TECHNOLOGY GANDHINAGAR
2024



ॐ भूर्भुवः स्वः ॥
ॐ तत्सवितुर्वरेण्यं भर्गो देवस्य धीमहि।
धियो यो नः प्रचोदयात् ॥

To

my family and friends,

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Vardaan Mongia

18.10.2024

Signature

Name: Vardaan Mongia

(Roll No: 19330020)

Date: 18.10.2024

CERTIFICATE

It is certified that the work contained in the thesis titled **"Testing the Strength of Quantum Random Number Generators"** by Mr. Vardaan Mongia (Roll No. 19330020), has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

18.10.2024

Prof. R. P. Singh
Professor
Atomic, Molecular and Optical Physics Division,
Physical Research Laboratory,
Ahmedabad, India.

(Thesis Supervisor)

Date: 18.10.2024

Thesis Approval

The thesis entitled

Testing the Strength of Quantum Random Number Generators

by

Vardaan Mongia
(Roll No. 19330020)

is approved for the degree of

Doctor of Philosophy

Examiner	Examiner
Supervisor	Director

Date: ____________________

Place: ____________________

Acknowledgments

I never knew I could work at such a high efficiency! My Ph.D. journey started as a pipe dream 10 years ago. Taking my school principal's advice on finding your community in this world and working for it paved the way towards getting a Ph.D. degree. I think I can safely say that the Ph.D. has been a transformative journey in my thinking process and certain healthy lifestyle choices. I feel humbled by the unwavering support of many friends, acquaintances, and mentors who challenged my skewed and somewhat eccentric notions of practicality.

Through this section, I acknowledge the efforts of major contributors towards the completion of the thesis focused on Bayesian sampling methods. I apologise if I missed some names as that is not my intent.

With that said, let's define a set of four people named ***family*** *which includes the members brother, sister-in-law, mother, and father. With their countless sacrifices, the ship has always sailed smoothly from the harbour. They always believed in me, encouraged me to pursue my dreams, and provided me with all kinds of support whenever I needed. I would also take a moment to thank my extended family including Nirmala, Rajiv, Kavita, Dreamsy, and Gurisha. They have always provided me with a fresh perspective on life whenever things looked dull and boring.*

The second set named ***mentors*** *whose members include Prof. R.P. Singh, Prof. Shashi Prabhakar, Prof. Namit Mahajan, Prof. Goutam Samanta, Prof. Debabrata Banerjee who took a head-on challenge to train me from a theoretical person to an experimentalist. It has been a challenging yet rewarding journey into the depths of knowledge and action in the lab by getting a chance to have a mentor support you through thick and thin. Your persistence in your pursuits inspires me every day. I would like to acknowledge their support, guidance, and patience. Your constructive remarks and friendly behavior always kept me motivated for thesis work.*

The third set named ***lab-mates*** *whose members include Pooja, Jayanth, Tanya, JK, Rutuj, Sarika, Ayan, Pravin, Suryansh, Rachita, and Nijil. I thank you for correcting my prejudices and enlightening me by borrowing chapters (and not pages) from your experience from their journey. I also thank visiting students Atharva, Arnav, Dixit, Parvatesh, Preeti for asking probing questions which made me research better and incorporate fresh perspectives from curious minds.*

I would like to express my profound respect to ***PRL*** *for providing me with all the fundamental amenities during my research work. I would also like to thank the PRL academic committee members for their insightful comments and encouragement throughout the research period. I am grateful to the academics and administrative staff members of IIT Gandhinagar for helping me with the registration procedure and considering me to reward me with my Ph.D. degree. What's more, I would take this opportunity to express my sincere gratitude to all my teachers who taught me at any stage of my life. Because of their guidance and constant support, I was able to reach this stage where I could write my doctoral thesis.*

The fourth set named ***JRF 2019*** *brings fond memories shared with Saumya, Rithvik, Santunu, Vineet, Birendra, Kimi, Neha, Bijoy, Saurabh, Sandipan, and Bharti Ganesh. I miss having discussions with you. I feel like I was never alone in this journey; the constant support of being a phone call away made me humble beyond measure. May it be watching a movie together or celebrating being promoted to SRFs that brings fulfillment to my heart.*

The fifth set names ***close acquaintances*** *inspires me to push my intellectual boundaries to new directions. Members of this family include all members from PSL lab members, Sivaraman's lab members, Luminescence all lab members, Meghna, Vaibhav from computer centre, Shivansh, Abhishek, Anindya, Unnati, Naval, and Bharat. Being a first-generation researcher, they have helped me navigate the path in the academic world.*

the sixth set is named ***Thaltej mates*** *whose members include Arup, Aditya, Akanksha, Anshika, Arijit, Goldy, Varsha, Kiran, Aravind, Ankit, Sanjay, Shreya, Rituparna, Naval,* *and Arjun. They provided a safe community to challenge my perceptions of the belief system. Ranging from badminton to table tennis to chess buddies, I have developed friendships for a lifetime.*

***Friends from past academic institutes** includes members, Abhinav, Mansimran, Shiva, Anupam, Shaurya, Namita, Udit, Ishit, Dr. Piyush Goyal, Titiksha, Deepti, Harish, Mithilesh, Shagun, Deepak, Rishabh, Karan Kapoor, Pavish, Aditya, Kanika, Rijul, Anuj Gupta, Sukhdeep Singh. They have provided valuable insights to help me get a Ph.D. at PRL.*

***Friends from chess community** include Vaibhav, Soham, Tarun, Animesh, Suraj, Ankit, Sharad, Narinder. I share countless memories of chess battles and fun exploring different parts of the country. I got the opportunity to express my sincere gratitude toward all of them.*

I would also like to express my deepest gratitude to all the ancestors of physics who left us with something to work on. I want to thank myself for never giving up on any situation. I want to thank myself for always believing in me. Above all, I want to thank God for showering blessings on me at all times.

Vardaan Mongia

Abstract

Random numbers lie in the heart of today's digital world. Random numbers have a plethora of applications in diverse fields. For example, random numbers serve as the basis for Monte Carlo simulations, which aid in modeling the probability of diverse outcomes in a process that cannot be determined easily because of the intervention of random variables. In the field of social sciences, one must have a sound random sampling method to be confident that the study group is a faithful representation of the whole population that one intends to describe. My purpose to study Quantum Random Number Generators (QRNGs) comes from their applications in cryptography. For example, while developing a secure quantum communication link between two parties, Alice and Bob, through the BB84 protocol, Alice needs to develop a random sequence of bits that is encoded in the polarization of a photon, which is then transmitted to Bob. However, random numbers have numerous applications beyond cryptography. For example, they are used in the gaming industry, where different random numbers have been used to design newer frame combinations where a user might play. This lowers the ability of a player to recognize patterns in the game and thus forces the player to rely on intuition. This can assist in psychometric analyses during hiring procedures such as those implemented by giants like Morgan Stanley. Another application of random numbers is in the field of randomized algorithms. Here, random numbers are used to solve particular problems such as sorting books alphabetically in the library, avoiding malicious attacks in page rank algorithms of the Google search results, etc.

This thesis attempts to decipher the advantage of Quantum Random Number Generators (QRNGs) over Pseudo Random Number Generators (PRNGs). With a plethora of applications in cryptography, the question often arises - Is there an advantage in using QRNGs over PRNGs in the cryptographic applications? Is it right to say that the advantage of employing quantum random number generators lies in the quantum unpredictability being better than computational unpredictability? If yes, what would be the defining parameters? If not, then what is the advantage of adding "quantum" resource to randomness? To define quantumness in QRNGs, what exactly is the resource that gives an edge over algorithmically generated bit streams?

The first work in the thesis provides a method to generate QRNGs by using the resource of quantum entanglement. Quantum Entanglement is a stronger resource than randomness and thus, could be used more than just to provide random bit streams. We use it additionally to provide device independence to our system, which aids in circumventing different kinds of attacks possible on the developed QRNG.

Furthermore, we question from the user perspective, given a bitstream, is there a method to see the quantum unpredictability signature in it by using any known methods? This could delineate the smeared boundary between quantum and pseudo-random number generators and thus, their respective application areas. Starting with elementary statistical tests, we have tried to differentiate the pseudo-random numbers from quantum random number generators based on machine learning and algorithmic randomness methods.

Finally, we generate shot noise-based QRNGs with two different sources, namely laser and white light from an LED, and compare their randomness properties with statistical test suites and study their trade-off between security and length of random bit-stream required. In addition to the above studies, we have given one application of PRNGs in the context of quantum key distribution to randomly make a choice of basis for which Alice sends a particular polarization.

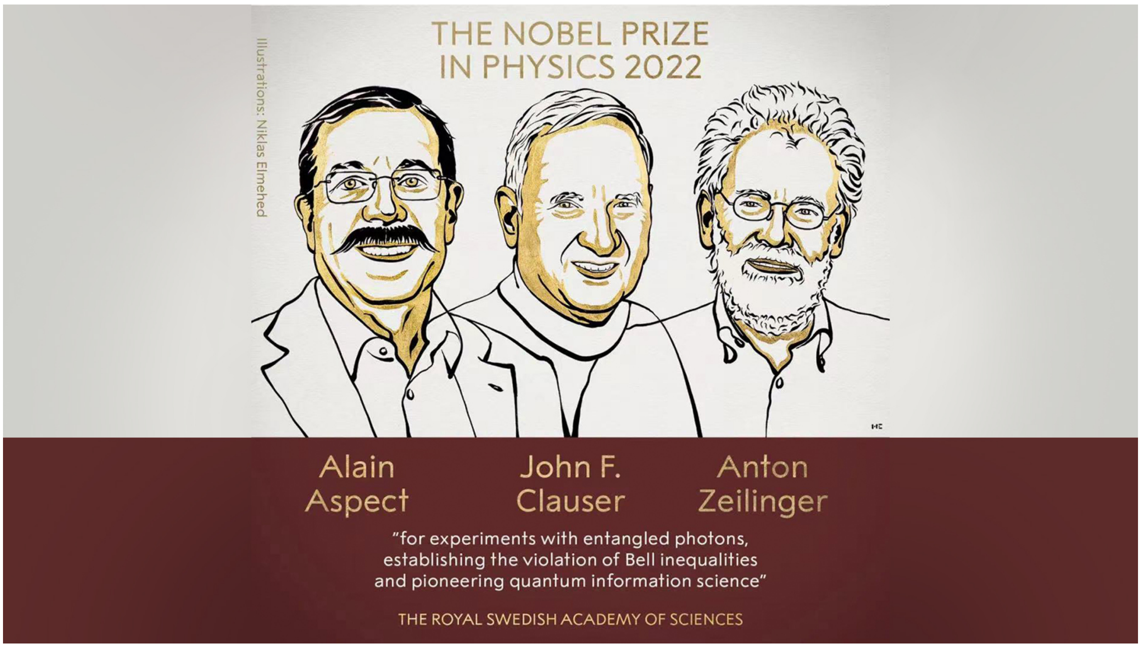
Oftentimes, QRNGs are proposed as the first practical application of quantum technologies. This research addresses the query of whether QRNGs have a place in the market of myriad quantum technologies to follow and are the knight watchman of quantum technologies.

Keywords: Quantum Entanglement, Quantum Random Number Generation, Device-Independent, Shot Noise, LED, True Random Number Generator, Pseudo Random Number Generator

Abbreviations

BS	Beam Splitter
PBS	Polarizing Beam Splitter
HWP	Half-Wave Plate
QWP	Quarter-Wave Plate
PM	Prism Mirror
SPDC	Spontaneous Parametric Down-Conversion
PRNG	Pseudo Random Number Generators
TRNG	True Random Number Generators
CRNG	Chaotic Random Number Generators
QRNG	Quantum Random Number Generator
HOM	Hong-Ou-Mandel
LFSR	Linear Feedback Shift Register
LCG	Linear Congruential Generator
FC	Fiber Coupler
SMF	Single-Mode Fiber
MMF	Multi-Mode Fiber
SPCM	Single Photon Counting Module
SPS	Single Photon Source
WCP	Weak Coherent pulses
DI-QRNG	Device-Independent Quantum Random Number Generator
SI-QRNG	Source-Indepedent Quantum Random Number Generator
MDI-QRNG	Meaurement-Device-Indpendent Quantum Random Number Generator
LASER	Light Amplification by Stimulated Emission of Radiation
LED	Light Emitting Diode

Nobel Prize in Physics 2022



"for experiments with entangled photons, establishing the violation of Bell inequalities and pioneering quantum information science"

ACM A M Turing Award Laureate 2023

Contents

List of Figures

List of Tables

"Quantum mechanics is certainly imposing. But an inner voice tells me that it is not yet the real thing. The theory says a lot, but does not really bring us any closer to the secret of the 'old one.' I, at any rate, am convinced that He does not throw dice."

Einstein, Albert

1

Introduction to Quantum Random Number Generation

Every war or battle ever fought in the world had to trade a lot of secret messages. And behind every encrypted message, there is a use-case of random numbers. If we see today's digital revolution spurred by Google, ranging from search results in everyday use to providing a digital identification mark for a system to fin-tech applications, it requires the use of random numbers. Random numbers are numbers that are hard to calibrate against any parameter as they are uncorrelated and uncertain to the extent possible. However, from an academic point of view, dice as a source of randomness was studied in the past by Francis Galton [4] in 1890 in his Nature article titled, "Dice for Statistical experiments" where he argues why dice are a better choice of randomness than shuffling identically marked balls in a bag or shuffling cards in a deck on empirical grounds.

As random numbers serve the basis of Monte-Carlo simulations [5] as Monte-Carlo simulations can be used to model atmospheric processes, such as light scattering, radiative transfer, and climate change. In the field of social sciences, one must have a sound random sampling method to be confident that the study group is a faithful representation of the whole population that one intends to describe. It is also used in the gaming industry where different random numbers have been used to design newer frame combinations where a user might play. This design including random numbers lowers the ability of a player to recognize patterns in the game and thus, the player mostly relies on intuition. It can assist in psychometric analysis during the hiring procedure by giants like Morgan Stanley [6]. A second application of random numbers includes the field of randomized algorithms. Here random numbers are used to solve particular problems such as sorting books alphabetically in the library, avoiding malicious attacks in page rank algorithms of Google search results, etc. A third application could be zero-knowledge proofs. In a statement, zero-knowledge proof means that you can verify to a prover that you are telling the truth about a statement without revealing the actual proof. Random numbers are an essential part to zero-knowledge proofs and find applications in many areas ranging from blockchain to data privacy [7, 8].

1.1 History of Random Numbers

The field of random number generators has a rich literature in the computer science community with applications ranging from encryption, allocating identity to different systems, and the gaming industry. As proposed by Claude Shannon in 1948 [9] to achieve the security of one time pad, one needs good quality and sufficient length of randomness resource. Since then, "randomness" resource has been a topic of interest. Attempts have been made to generate and use randomness as a resource [10], however, it is tedious to quantify this resource. The famous quote by John von Neumann "Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin." points to the well-known fact that randomness generated from algorithms is not the best bet for cryptographic applications as the uncertainty can be calibrated using advanced algorithms. Quantum randomness comes into the picture when the security/privacy of sensitive information is involved. The word "quantum" signifies the use of principles of quantum mechanics to enhance the present-day security standards of encryption by providing a deeper calibration on the process of generation than the output bit-stream.

Random numbers can be classified on various grounds. One of the classifications is based on the process via which they are generated. This classification is highlighted in Figure 1.1. This classification helps us in understanding where QRNGs fit in with conventional methods such as pseudo-random number generators (PRNGs) or chaotic random number generators (CRNGs) in the domain of random number generators.

Historically, the aim to develop unpredictable random numbers started with Pseudo Random Number Generators (PRNGs). These computer-simulated algorithms generating random numbers are apparently random and deterministic in nature; for example, techniques based on the mid-square method [11], linear congruential method [12]. They are cost-effective, fast, periodical, and seed dependent. Nonetheless, in cryptography, we also want the bits to be independent i.e. they should be both forward and backward secure. Backward security means that an attacker who knows the whole sequence cannot guess the next bit with a probability better than one-half. While forward security means that knowledge of a part of the sequence doesn't allow

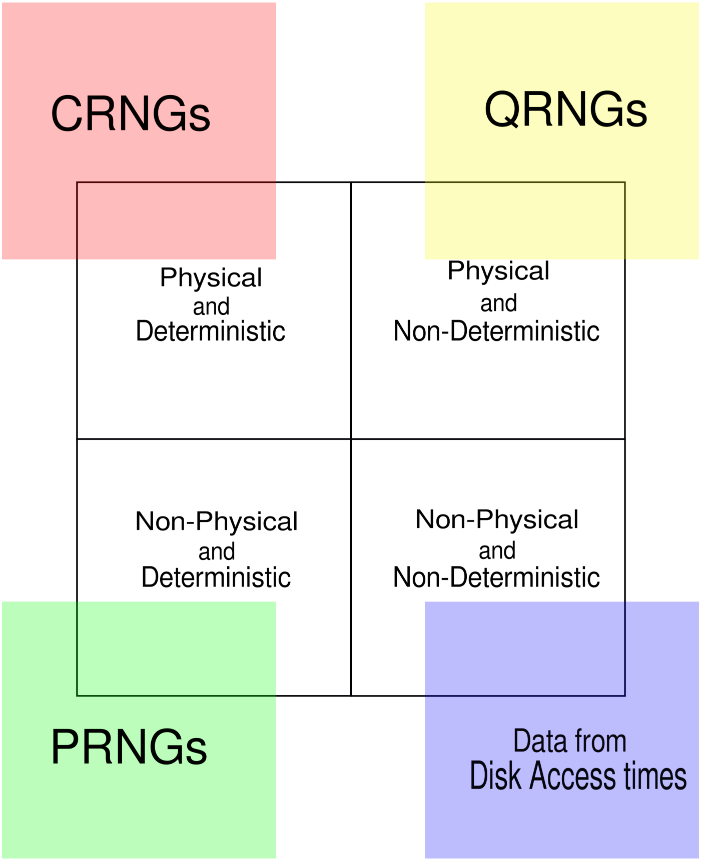


Figure 1.1: Classification of random numbers based on process of generation. CRNGs: Chaotic Random Number Generators, PRNGs: Pseudo Random Number Generators, QRNGS: Quantum Random Number Generators.

an attacker to compute the previous values of the generator with better accuracy than guessing. With this definition in mind, one can get the cryptographically secure pseudo random number generators (CS-PRNGs) which are used in various virtual private network (VPN) services [13] or transport layer security (TLS) of mailing services [14].

Another class of random number generators (RNGs), as shown in 1.1, provides an alternative to unpredictability features. These generators are based on complex chaotic systems. Hence, they follow the name Chaotic Random Number Generators (CRNGs). A chaotic system is one where a slight change in input brings a towering change in output i.e. "The Butterfly Effect" [15]. For illustration, consider the spontaneous chaos in semiconductor superlattices at room temperature [16]. Here, the superlattice is a lattice where different layers have different materials with varied compositions. Noise from one layer of the superlattice adds up to form a chaotic system.

Although this method to generate random numbers fulfills the current needs, the innate process generates randomness at the cost of increasing complexity (with no upper bound) and could easily be superseded by an eavesdropper without getting caught and thus, doesn't fit the requisites for Quantum Key Distribution (QKD) protocols in quantum cryptography. For example, an attacker can access the chaotic system with an initial condition, measure its input, measure the same physical observable (output), and add spurious noise to produce a similar copy of the random data. Since there is no characterization of noise, Eve's probability of hijacking the communication channel and getting unnoticed increases exponentially. It is one reason we move towards quantifiable physical and non-deterministic random number generators.

For traditional RNGs, such as PRNGs/CRNGs as depicted in Figure 1.1, their randomness is advocated either on computational or stochastic complexity. Quantum RNGs provide calibration a step further into the process of generation rather than the output bitstream as in traditional methods. Alternatively, using a non-deterministic and a non-physical random number generator is a viable option for attaining unpredictability but includes human bias as a source of randomness and thus falls short of trusting the source of randomness.

Hence, from the above arguments, it follows that quantum random number generators are the hobson's choice for use of random numbers in the security domain. The "quantum" choice uses fundamental properties of particles to exist in a superposition state. This step is followed by the measurement process, which is a non-unitary process, and one cannot calibrate the process better than probabilities. This uncertainty, or other uncertainties like the Heisenberg uncertainty, are the underlying reason for random number generators based on laws of quantum mechanics. If these uncertainties are calibrated, there are no random numbers. Hence, for security applications, we have moved towards random numbers generating uncertainties which are computational or stochastic to ones which cannot be violated by nature. For example, defining the position of a photon to arbitrary accuracy cannot be achieved, and thus, could be used as a resource for generating quantum randomness. It is worth mentioning that uncertainty pertaining to our ignorance, such as flipping a coin, is not a true source of randomness (CRNGs). However, given the time constraint to calculate the result, they are an optimized solution. The same statement is also true for PRNGs. The uncertainty in PRNGs is attributed to the computational hardness of the algorithm. One such computationally hard problem used in quantum cryptography is factorisation of bi-primes, which forms the basis of the asymmetric key algorithm. However, with the advent of Peter Shor's algorithm, patterns amongst such computationally hard problems have started emerging, buttressed by special properties like superposition [17]. However, with the breakthrough of quantum algorithms, it is only a matter of time that apparently complex systems, providing computational complexity as a source of randomness, are solved.

1.1.1 Days of Recent Past

After the first relevant demonstration of optical QRNGs [18] in 2000, the field of QRNGs has exploded with plenty of research papers in this direction that it becomes difficult to keep track of where the quantum advantage is. Once researchers found out that quantum random number generators have an advantage pertaining to their innate assumptions of quantum mechanics, the initial research exploded in the direction of numerous ways to generate these quantum random numbers ranging from Nitrogen vacancy centres [19] to perovskite light emitting diodes [1]. Today, smaller chip implementations are in progress [20]. However, since the outcome is computational and to date measured using computational methods, there has been a gap in how to test these quantum random number generators. In 2010, after Pironio's work [21], the field got re-directed in the direction of exploiting quantum correlations to evade the need to put trust in the devices used. The Figure 1.2 shows the transition of QRNG techniques from device-dependent scenarios to device-independent schemes.

As a practical trade-off between security and bit-rate, an intermediate branch of semi-device-independent protocols based on Bell's theorem has also emerged. All these protocols use Bell's theorem in one form or the other to guarantee partial or full device independence.

1.2 Necessity of randomness as a resource

In this section, we will give numerous examples which explain why we study and use quantum random numbers instead of their conventional alternatives. We cherry-pick Google's page rank algorithm as our first example as its impact is comparable to the

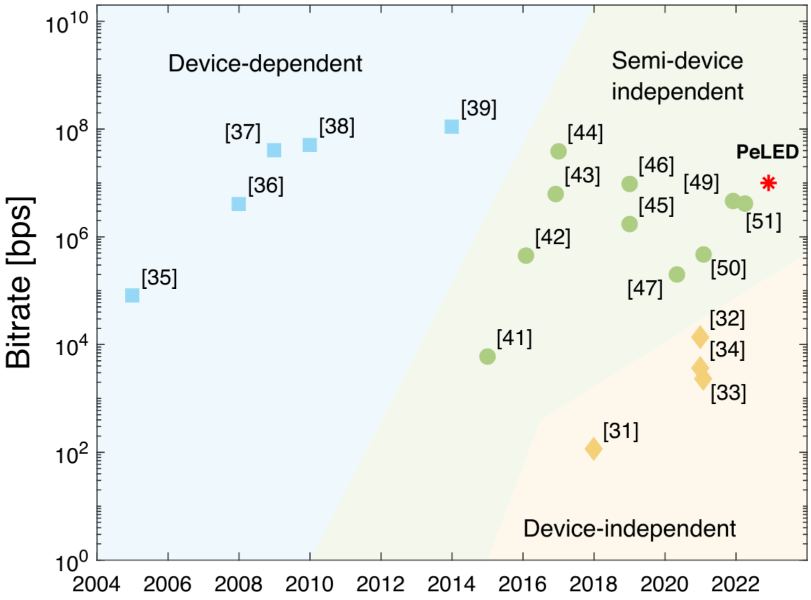


Figure 1.2: Recent shift from device-dependent to device-independent protocols in the QRNG domain. Each numbered data-point in the figure represents a good alternative research method to approach device-independence [1]

.

industrial revolution.

1.2.1 Google's Page Rank Algorithm

Google's Page Rank Algorithm is one of the famous algorithms behind search results and thus, influences everyday life in a plethora of directions. The algorithm classifies webpages based on their popularity such that the most popular results can be shown to the users. Heuristically speaking, the algorithm works as follows: a webpage X is ranked as popular if it has multiple inbound links from independent webpages. In order to calculate the number of incoming and outgoing links, the algorithm operates by going to the source code of a webpage, looking for the outbound links, and jumping to a new webpage. By doing this process iteratively, an accurate picture of the complex network of links can be drawn. However, it may happen that a series of webpages could be pointing to one another in order to try to fool the algorithm by not being truly independent. It helps the webpage gain popularity maliciously. To avoid this trap, one possible solution is that the algorithm jumps **randomly** to a webpage Y (with a finite probability p) which is not necessarily in the list of outbound links of the current webpage. By throwing in randomness into the algorithm, the algorithm decides whether to follow a link or to randomly jump to a website, escaping traps.

1.2.2 Cost of use of poor quality random numbers: Privacy and millions of money, time and energy

Over the span of six years, from 2006 to 2012 there have been many reports of attacks on cryptographic keys generated by weak PRNGs. A major incident happened in 2010 when an attack was carried out on users of Sony's PlayStation 3 (PS3) game console (in which data was stolen for at least 77 million users) pertaining to a flaw in the implementation of the Elliptic curve-based digital signature algorithm by Sony which used the same nonce(numbers repeating only once) multiple times for authentication [22]. The conferences DEFCON and Chaos Communication Congress are events which to date organize events on finding security loopholes in cryptographic areas. It was pointed out by researchers that many of the RSA keys used for encryption over the internet use a poor random number generator and thus, are unfit for use in cryptographic applications. Investigations revealed that the NSA intentionally secretly lowered the security of the world's popular hardware and programming solutions for the purpose of crypto attacks on encrypted content (inclusive of attacks on RNGs):– Dual EC DRBG (Dual Elliptic Curve Deterministic Random Bit Generator) was used, a PRNG created and strongly pushed as a standard by the NSA. Only in 2013 did it turn out that the NSA was the only one to have a backdoor for the generator. Upon disclosure, RSA Security and the US National Institute of Standards and Technology (NIST) instructed not to use the Dual EC DRBG generator [23]. On the same note, Google confirmed that the IBM Java SecureRandom class in Java Cryptography Architecture (JCA) generated cyclic sequences, which compromised application security made for Android to support the electronic currency Bitcoin; the equivalent of a large amount of USD in Bitcoins was stolen. Similarly, it is suspected that the attack on the Tokyo cryptocurrency exchange MtGox, in which more than 800,000 Bitcoins were stolen (which resulted in the declaration of bankruptcy by MtGox [24]) was related to the poor implementation of random number generators.

1.3 Different types of classification of QRNGs

There are numerous platforms to generate QRNGs. For example, using an electron's spin as a qubit and if its superposition state is engineered or a maximally mixed state engineered from an entangled state can be used to generate random numbers. We can use the statistics of nuclear decays as well. Nonetheless, we are mostly interested in QRNGs using optics.

1.3.1 Classification of Optical QRNGs on grounds of assumptions of the devices

Current optical QRNGs are classified into three broad categories based on the type of assumptions involved about the generation source and measurement device, as depicted in the Figure1.3.

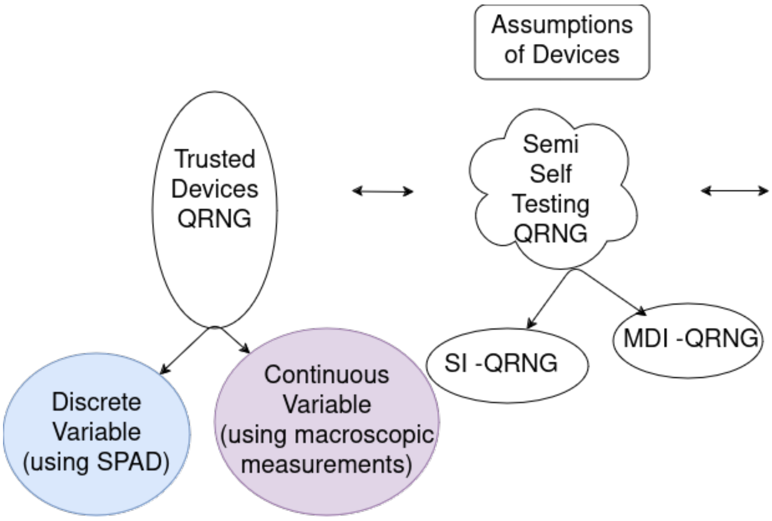


Figure 1.3: Classification of QRNGs on grounds of assumption of devices: Trusted devices and Device QRNGs lie on extreme ends of the optimisation axis defining the trade-off between trust on devices and bit-rate. The color shows pursued works in the thesis.

The first category is built on fully trusted sources and typically can generate randomness at a high speed with the security of these QRNGs being assured by properly modelling the generation devices. The first category of "Trusted Devices" can further be divided based on the type of variable being measured, namely discrete or continuous. Discrete variable optical QRNGs (DV-OQRNG) are based on single photon's properties. For example, the state of polarisation of a photon, its arrival time or its number statistics. Alternatively, continuous variable optical QRNGs (CV-OQRNG) are based on collective properties of the photons, say, phase noise or shot noise of the laser pulse. The fast detection of continuous variables uses a larger Hilbert space compared to discrete variables and is easily expandable with additional components, as per user requirements for specific applications, which makes them cover many of the problems posed by physical, non-deterministic random numbers. Since we measure different properties of photons in discrete and continuous variables, the detection scheme for both discrete and continuous variables must be different. One needs access to good single photon counting modules based on avalanche photodiodes to measure discrete properties. Other expensive but better alternatives include superconducting nanowire single photon detectors (SNSPDs), transition edge sensors (TES) etc. On the other hand, continuous variable measurements are often accompanied by less expensive photodiodes. Trusted device QRNGs imply trust in the good working of devices. However, if the device goes bad or is supplied by a vendor with malicious intent, there is no way to check that. Hence, the need for self-checking QRNGs comes into the picture.

Secondly, the other extreme is Device-independent QRNG, in which verifiable randomness can be generated without trusting the actual implementation. These QRNGs test themselves for any possible set of classical correlations or bad behaviour. For example, constructing a witness operator to check for classical correlations. They provide a higher security compared to fully trusted sources. However, it comes at a cost of very low bit-rate [25]. One can say there is a trade-off between generation speed and verifiability of QRNGs. Fewer the assumptions, higher the security and the quality of QRNG. Device-independent QRNG uses Bell's Inequality for certifying the quantum random number generation process. If the value of the Bell parameter S is greater than 2, it implies the correlations are non-local.

The third category, semi self-testing QRNG, is an intermediate category that provides a trade-off between the trust on the device (or bits tested for quantum origin) and the speed of generated random bits. It is the best practical solution. Source-independent QRNG (SI-QRNG) and measurement device-independent QRNG (MDI-QRNG) techniques come under this category. SI-QRNG completely trusts the source while measurements are not trusted. On the contrary, for MDI-QRNG we don't trust the measurement. The source sends some auxiliary bits to check the measurement device. The rest of the bits are used to generate a sequence of quantum random bit-streams. Both these techniques are used when part of the system (either measurement or generation source) is trusted. Many different protocols [26] have been built up in this direction which alternate between test and generation rounds based on another random number generator.

1.3.2 Classification of Random Numbers on Algorithmic randomness

Random Numbers can also be classified on grounds of whether they are algorithmically constant or random and whether they are Turing-computable or not. A method is called Turing-computable if it can be simulated on a Turing machine. Secondly, methods that are algorithmically constant are the ones that are apparently random amongst statistical measures of randomness; however, their process is algorithmically constant. With the two axes, namely Turing computability and algorithmically random as classifiers, we start with the classification of PRNGs as shown in Figure 1.4. PRNGs such as mid-square method, or LFSR based random numbers fit in the class of being algorithmically constant and Turing-computable. On the other extreme, we have numbers that are not only algorithmically random but also Turing in-computable. One concrete example is entanglement-based QRNGs. It is important

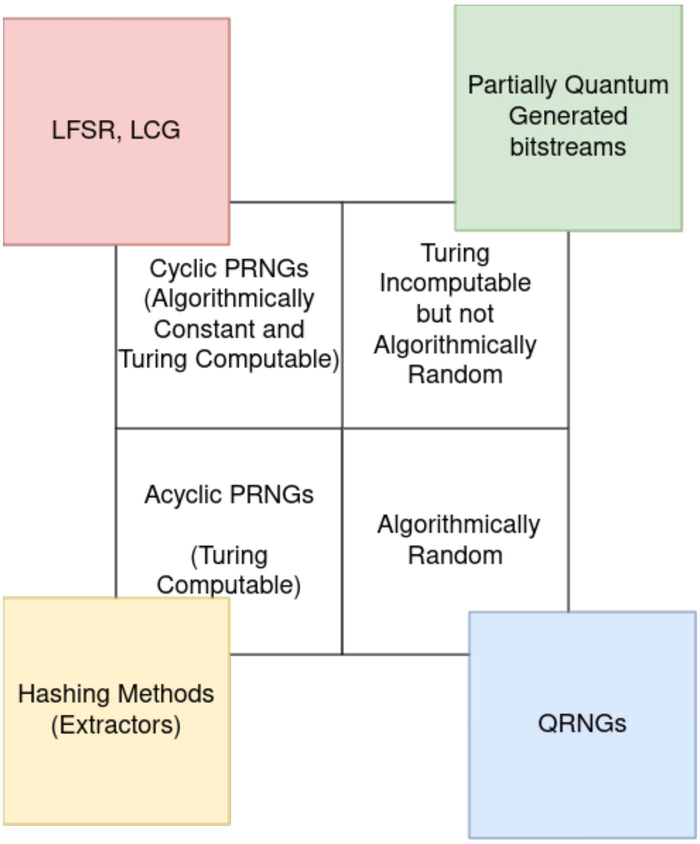

LFSR, LCG
Partially Quantum Generated bitstreams
Cyclic PRNGs (Algorithmically Constant and Turing Computable)
Turing Incomputable but not Algorithmically Random
Acyclic PRNGs
(Turing Computable)
Algorithmically Random
Hashing Methods (Extractors)
QRNGs

Figure 1.4: Classification of random numbers on grounds of algorithmic randomness.

to note that not all QRNGs are algorithmically random and not all PRNGs are algorithmically constant. Consider the following example: A laser source attenuated to a single photon level fires photons, following a Poissonian distribution in time, and registers a click in the detector. This click corresponds to the bit “1” in the bit-stream. The bit-stream “0” can be algorithmically introduced between these sparse ones (owing to the Poissonian distribution) such that the sequence is algorithmically constant and patterns in bit-streams “0” can emerge. However, the overall sequence in such a case is still Turing-incomputable as there is a quantum mechanical origin of bits “1” which cannot be simulated. Hashing methods, however, are Turing-computable but algorithmically random. They can be simulated on a classical computer given enough resources. But are too complex that a simple algorithm can be written to catch these highly a-cyclic random numbers.

1.4 Quantum Random Number Generators

1.4.1 Basic Parameters

QRNGs need to be characterized for two parameters. One is the quality of randomness, while the other is bit-rate. Although PRNGs are pretty fast (in Gbps), they get rejected on the more important parameter, quality of randomness, in the context of cryptographic applications. This quality could be measured on different parameters such as computational unpredictability or no statistical correlations. We evaluate QRNGs with similar parameters, but unpredictability is defined on quantum mechanical parameters. Statistical correlation independence follows the same methodology as for PRNGs. Quantifying bit-rate for QRNGs is straightforward; say for discrete variables, they hold a one-to-one correspondence with the measured results.

Figure 1.5 explains the typical procedure followed for a QRNG development. From a physical process, measurements are made on quantum states and bits are assigned through a particular method convenient to the experimenter's choice as long as the method is unbiased amongst bit assignments of "0" and "1". The raw bit-streams, although giving good performance on unpredictability, lack statistical independence amongst the bits pertaining to experimental imperfections. This is circumvented by post-processing the bit-stream with a hash function as discussed in Chapter 3. The diagram shows two things. One, the advantage of quantum random number generators is typically associated with the calibration of the blue box rather than conventional methods of calibrating the raw output bit-stream. Secondly, the extracted random sequence in the diagram is post-processed using the von Neumann extractor on the raw bit-stream.

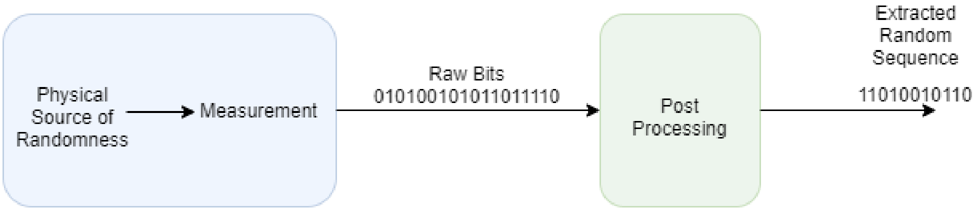


Figure 1.5: Block diagram of a random number generator.

1.4.2 Life Cycle of a quantum random number generator

To describe the quality of a random number generator, understanding how the quality gets affected by different processes in the life cycle of a quantum random number generator is crucial. To address the query, researchers have traced out the journey of a random number generator over different processes involved in the life cycle of the random number generator [27].

Figure 1.6 shows the variation of quality of randomness (measured using computational methods) over the life-span of a typical random number generator. The initial process, depicted by Step 1, denotes the physical process of generation using laws of physics. For example, a single photon source (ideal or approximate) encoded in diagonal polarization ($|D\rangle$) is measured in H-V basis (mutually unbiased basis (MUB) to D-A basis). These measured results are completely based on natural laws. This detection may not be ideal pertaining to the detection scheme, that involves some recovery time and dead time before detecting the next photon. It can be circumvented

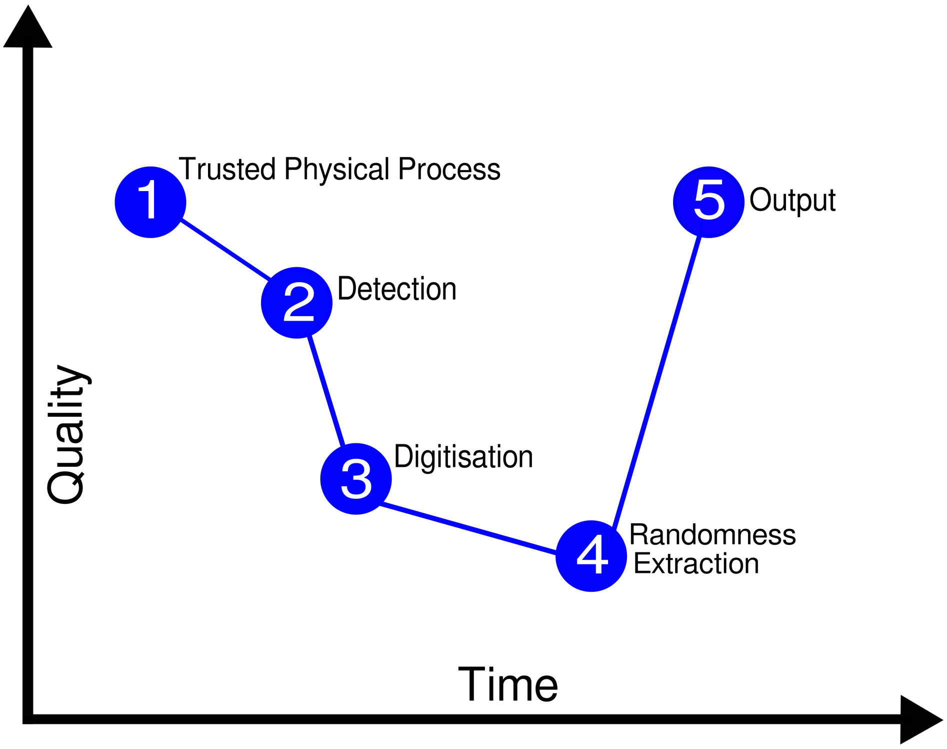


Figure 1.6: Life cycle of a random number generator.

as we don't care about the undetected photons but care that it is a single photon when detected. However, the detectors have a wavelength-specific response. Also, the challenges of on-demand single photon sources add to the deterioration of the quality of random numbers generated since SPADs are click (on/off) detectors. The third step is the digitization process. This process is more relevant for continuous variable random numbers where the assignment of bits is typically based on the resolution of the Analog to Digital Converter (ADC) used. During the digitization process, the quality degrades due to the sampling choice of a particular distribution based on human intervention. To compensate for all these biases introduced over different steps, typically post-processing of the raw bit-stream, namely randomness extraction, is used. The extraction process re-distributes the biases such that the quality of the random number generator increases drastically. The last step is computational while the first three are physical processes. We discuss this abrupt (and apparently ad-hoc) increase in Chapter 5.

1.5 Objective of Thesis

The objective of the thesis is to test the quantum random number generator on different fronts and thus, we take a step towards developing a test-kit for quantum random number generators. This thesis attempts to decipher the domain advantage of Quantum Random Number Generators (QRNGs) over other categories of random number generators (RNGs) as discussed in Figure 1.1. With a plethora of applications in cryptography, the question often arises - Is there an advantage in using QRNGs over PRNGs in the other applications? Is it right to say that the advantage of employing quantum random number generators lies in the quantum unpredictability being better than computational unpredictability? If yes, what would be the defining parameters? If not, then what is the advantage of adding "quantum" resource to randomness? To define quantumness in QRNGs, what exactly is the resource that gives an edge over algorithmically or unpredictability-based generation of bit-streams?

The first work in the thesis provides a method to generate QRNGs by using quantum entanglement as a resource. Quantum entanglement is a stronger resource than randomness and thus could be used for more than just providing random bit-streams. We use it additionally to provide device-independence to our system, which aids in circumventing different kinds of attacks possible on the developed QRNG.

Furthermore, we question from the user perspective, given a bit-stream, is there a method to see the quantum unpredictability signature in it by using any known methods. This could delineate the smeared boundary between quantum and pseudo-random number generators and thus, their respective application areas. Starting with elementary statistical tests, we have tried to differentiate the pseudo-random numbers from quantum random number generators based on machine learning and algorithmic randomness methods.

Finally, we generate optical noise-based QRNGs with two different sources, namely laser and white light from an LED, and compare their randomness properties with statistical test suites and other measures. In addition to the above studies, we have given one application of PRNGs in the context of quantum key distribution to randomly make a choice of basis for which Alice sends a particular polarization.

Often-times, QRNGs are proposed as the first practical application of quantum technologies. This research addresses the query of whether QRNGs have a place in the market of myriad quantum technologies to follow and are the night watchman of quantum technologies.

1.6 Overview of Thesis

The Thesis is divided into seven chapters. The first chapter, i.e., this chapter, introduces the reader to the motivation and the need for quantum random number generators in security applications to preserve the privacy of sensitive information. It further discusses classification related to quantum random number generators in the random number generation domain. Furthermore, we classify quantum random number generators based on assumptions of devices involved. In Figure 1.7, a brief outline of the inter-connectivity of the chapters is drawn out.

Chapter 2 covers the preliminaries needed to understand the theoretical concepts such as spontaneous parametric down conversion(SPDC) generation, develops theo-

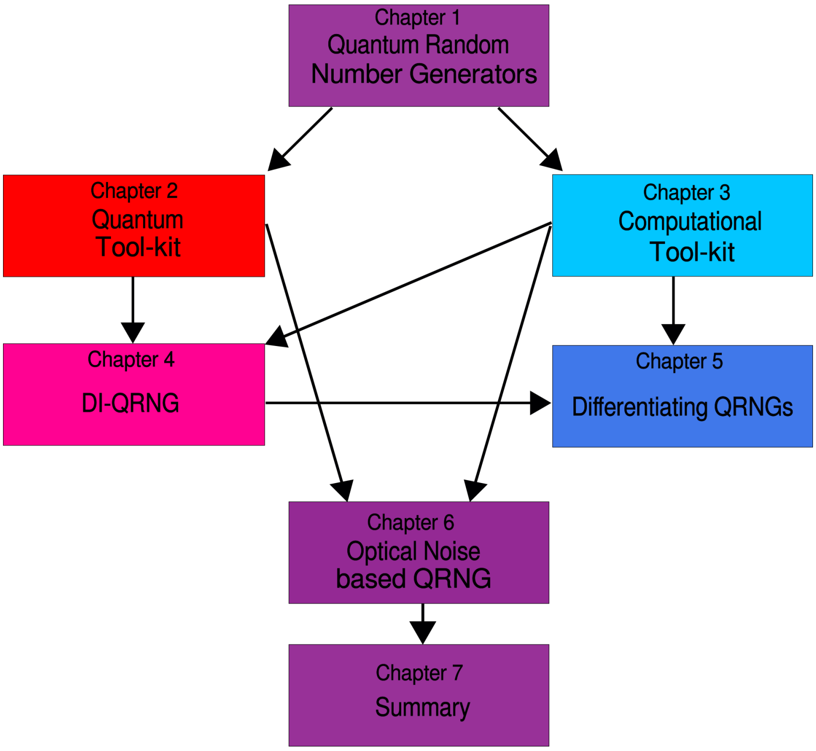


Figure 1.7: Thesis outline and inter-connectivity of the chapters: the figure is color-coded for better understanding (red: quantum tools, blue: computational tools, all other chapters are combination of these colors).

retical understanding of Hong-Ou-Mandel (HOM) effect and quantum entanglement. It includes observation and characterisation techniques for the same. We, then, discuss different kinds of light sources following different statistics. Towards the end, we discuss two illustrative experiments. With the first experiment, we draw a comparison of weak coherent pulse based and SPDC based single photon sources and differentiate them based on information theoretic measures. For the second experiment, we build on the conclusion of the first experiment that relates the amount of single photon nature and randomness. We explore the said relationship further with SPDC based single photon source. We see its variation along three parameters and draw a connection between heralded single photon source and randomness.

Chapter 3 discusses different kinds of conventional random number generators used in cryptographic applications. Specifically, the generators which are hardware-friendly for Internet of Things (IoT) implementations such as Linear Feedback Shift Register (LFSRs) based and Cryptographically Secure-PRNGs like ChaCha20. Furthermore, the chapter discusses post-processing methods involved with random numbers. Finally, on the testing front, different measures of quantifying randomness are introduced. It includes entropy-based metrics in probability space, statistical test suites like NIST, Dieharder, etc.

Chapter 4 discusses the most secure QRNG based on quantum entanglement. Here, we show how quantum correlations, specifically, quantum entanglement, can be used to eradicate trust in the devices. It proves the advantage of using quantum phenomena to generate QRNGs. The entanglement checks are cross-verified using tomography data combined with Bayesian estimation techniques. The use of relative phase between two photons serves as a measure of device independence, which is subdivided into the HOM curve as a certifier on the source front and tomography and estimators versus the Bell parameter on the measurement-device-independent front.

Chapter 5 deciphers the distinction between PRNGs and QRNGs based on a given bit-stream. However, it is observed that there is no distinction between the two methods if only the output file(in the form of bit-stream) is known. This observation has been recently conjectured as a no-go theorem for QRNGs. We strengthen the theorem by cross-verifying the analysis on our set of QRNGs and PRNGs and additionally support the theorem on the machine learning front.

Chapter 6 discusses generation of random numbers from two different sources. Both techniques use the different input sources (LASER and white light). However, detection scheme for both is the same i.e. homodyne detection (using two photodetectors). We compare their results against standard tests of randomness such as NIST and discuss the quantum advantage trade-off with cost for specific applications.

Chapter 7 summarizes the need of going through different routes of testing the random number generators, namely to build a quality test suite for QRNGs. The chapter provides a future outlook on the subject of quantum random number generators.

"The new theory, which connects the wave function with probabilities for one photon, gets over the difficulty by making each photon go partly into each of the two components. Each photon then interferes only with itself. Interference between two different photons never occurs."

Dirac, P.A.M.

2

Developing the quantum tool-kit for quantumness in QRNGs

Quantum optics is the most complete theory which provides experimental verification on many aspects with an increasing generalisation from ray optics to wave optics, to electromagnetic theory of light, to the outermost concentric circle of quantum optics providing an explanation to virtually all optical phenomenon. We begin our understanding of these quantum phenomenon relevant to develop a tool-kit to characterise QRNGs by first diving into different sources of light, followed by generation and manipulation of quantum sources of light.

2.1 Light sources

Until recently, the major light source which was physically realizable was a thermal source. The most widely used light sources were the incandescent bulbs, bright light sources such as mercury lamps. With the development of lasers, a quest for different kinds of light sources began. Broadly, we speak of coherence functions $g^1(x_1, t_1; x_2, t_2)$ and $g^2(x_1, t_1; x_2, t_2)$ to describe different types of light sources. Typically, we define sources at a fixed position i.e. $g^1(t_1; t_2)$ and $g^2(t_1; t_2)$ or $g^1(\tau)$ and $g^2(\tau)$ where $\tau = t_2 - t_1$. The function $g^1(\tau)$ is defined as

$$g^{(1)}(\tau) = \frac{\langle \hat{E}(t)\hat{E}(t+\tau)\rangle}{\langle |\hat{E}(t)|^2 \rangle} \tag{2.1}$$

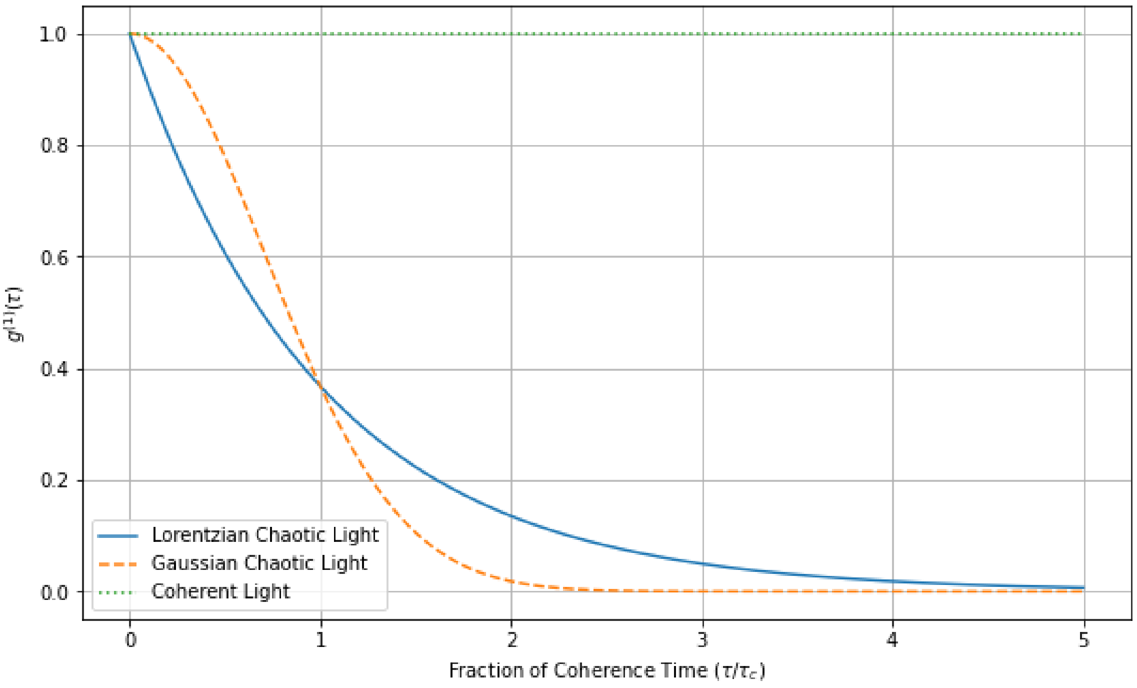


Figure 2.1: Different light sources characterized by field-field correlations as a function of their coherence times

and the function $g^2(\tau)$ is defined as

$$g^{(2)}(\tau) = \frac{\langle \hat{E}^{(-)}(t)\hat{E}^{(-)}(t+\tau)\hat{E}^{(+)}(t+\tau)\hat{E}^{(+)}(t)\rangle}{\langle \hat{E}^{(-)}(t)\hat{E}^{(+)}(t)\rangle \, \langle \hat{E}^{(-)}(t+\tau)\hat{E}^{(+)}(t+\tau)\rangle} \tag{2.2}$$

where $\hat{E}^{(+)}$ and $\hat{E}^{(-)}$ are the positive and negative frequency components of the electric field $\hat{E}$.

More explicitly, they can be written in terms of a and $a^\dagger$ with the relation,

$$\hat{E}^{(+)}(t) = \sum_{\vec{l},\sigma} \sqrt{\frac{\hbar\omega_{\vec{l}}}{2\epsilon L^3}} a_{\vec{l},\sigma} e^{-i(w_{\vec{l}}t - \vec{k}_{\vec{l}}\cdot r)} \vec{e}_{\vec{l},\sigma} \tag{2.3}$$

and

$$\hat{E}^{(-)}(t) = \sum_{\vec{l},\sigma} \sqrt{\frac{\hbar\omega_{\vec{l}}}{2\epsilon L^3}} a^\dagger_{\vec{l},\sigma} e^{+i(w_{\vec{l}}t - \vec{k}_{\vec{l}}\cdot r)} \vec{e}_{\vec{l},\sigma} \tag{2.4}$$

where $\vec{l}$ is the direction of propagation of the field component, σ defines the polarization vector and $\vec{e}_{\vec{l},\sigma}$ is a unit vector orthogonal to $\vec{k}_{\vec{l}}$.

The coherence functions for different light sources are shown in Figure 2.1 and Figure 2.2, respectively.

2.1.1 Coherent Sources

A source with an almost zero line-width is referred to as a monochromatic source. Typically, $g^{(1)}(\tau) = constant$ for any perfectly monochromatic source. Laser sources,

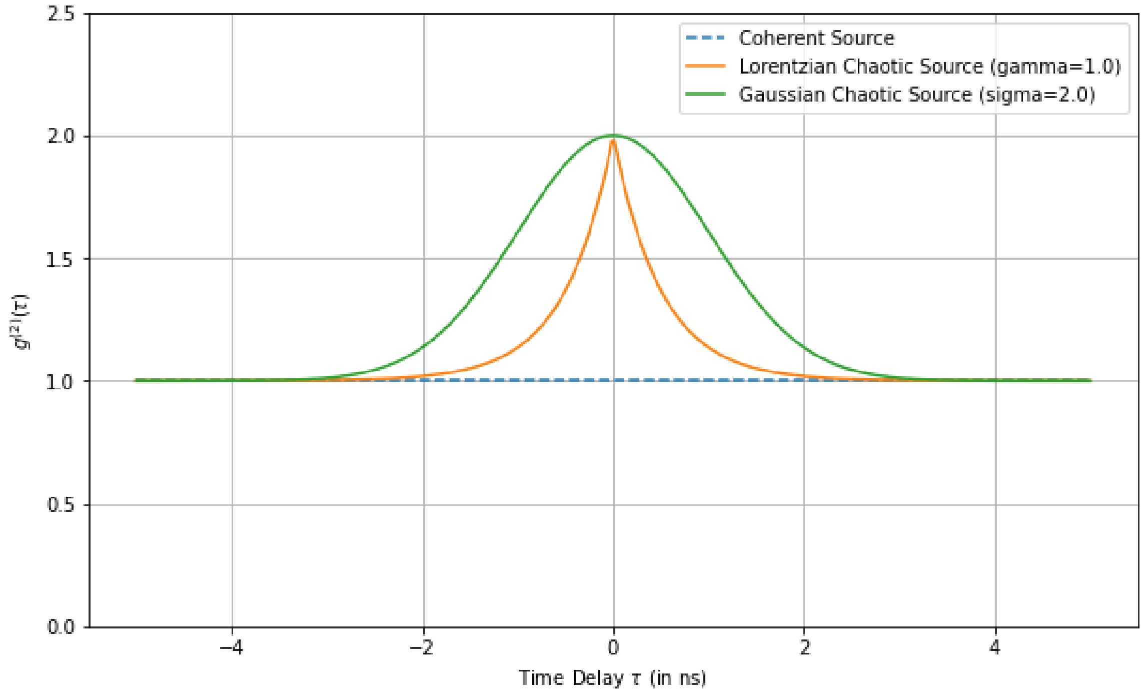


Figure 2.2: Different light sources characterized by Intensity-Intensity correlations as a function of their coherence times. Gamma defines the coherence function of the interference term of Lorentzian chaotic light source and sigma defines the standard deviation of the Gaussian chaotic light source. Values chosen are for representational purposes only.

typically, are monochromatic in nature. A coherent source of light, denoted as $|\alpha\rangle$ is generally referred to as a source with $g^{(2)}(\tau) = 1$.

2.1.2 Thermal Sources

White light originating from a light emitting diode (LED) is a good example of a thermal source. For white light emitted from thermal sources such as LEDs, the photon number follows a Boltzmann distribution, characterized by the equation

$$p(n) = \left(1 - \exp\left(\frac{-\hbar\omega}{kT}\right)\right) \exp\left(\frac{-n\hbar\omega}{kT}\right). \tag{2.5}$$

Here, the system is assumed to be in thermal equilibrium at some temperature T, ω is the angular frequency of the field, and k is the Boltzmann constant. For large mean photon number $\overline{n}$, the distribution becomes

$$p(n) = \frac{\overline{n}^n}{(1+\overline{n})^{1+n}} \tag{2.6}$$

which is a Bose-Einstein distribution with mean value $<I> \propto \overline{n}$. Here, the variance of the photon number is found to be $\Delta n^2 = \overline{n}^2 + \overline{n}$. The Δn^2 term of the variance grows as the mean photon number increases, eventually dominating completely.

2.1.3 Chaotic Sources

For a collision-broadened light, the first-order coherence function is defined as

$$g^{(1)}(\tau) = exp(-iw_0\tau - \frac{|\tau|}{\tau_0}) \tag{2.7}$$

and the second-order coherence function related through the Siegert relation as

$$g^{(2)}(\tau) = 1 + |g^{(1)}(\tau)|^2 \tag{2.8}$$

For $\tau = 0$, the $g^{(2)}(\tau)$ achieves a maximum value of 2.
In general, $g^{(2)}(\tau) < g^{(2)}(0)$. This inequality indicates a photon bunching phenomenon.

2.1.4 Quantum Sources

Quantum sources are sources that show anti-bunching phenomena which can be experimentally verified with $g^{(2)}(0) < 1$ [28]. The quantum source can be a quantum dot or SPDC-based single-photon source. Consider an example of a Fock state with n photons, a quantum state, $|n\rangle$, with

$$g^{(2)}(\tau) = g^{(2)}(0) = 1 - \frac{1}{n} \qquad n \geq 2 \tag{2.9}$$

.

The values of $g^{(2)}(0) < 1$ are an indicator of quantum-mechanical violation of Cauchy inequality [29]. States of such a kind often possess sub-Poissonian statistics unlike their counterparts, i.e. coherent states, which follow Poissonian statistics.

2.2 SPDC Process

Spontaneous Parametric Down Conversion (SPDC) defines the process which involves generating low-energy photons from a parent photon in a spontaneous manner. Spontaneous, meaning that the process of energy transfer of the photon of energy ω_p to vacuum modes is instantaneous. The parametric implication implies that no absorption of energy takes place in the material. The entire energy is transferred to the down-converted photons as depicted in Figure 2.3. In the optical interaction of a parent pump photon with the non-linear crystal, the two daughter photons, namely signal and idler, are generated. This generation process conserves both linear momentum and energy of the system such that

$$\omega_p = \omega_s + \omega_i \tag{2.10}$$

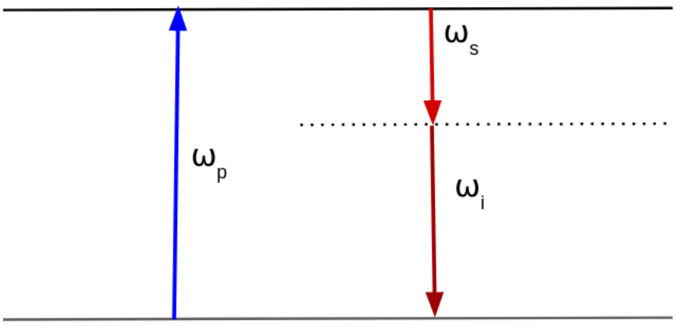


Figure 2.3: Energy Conservation in a parametric process.

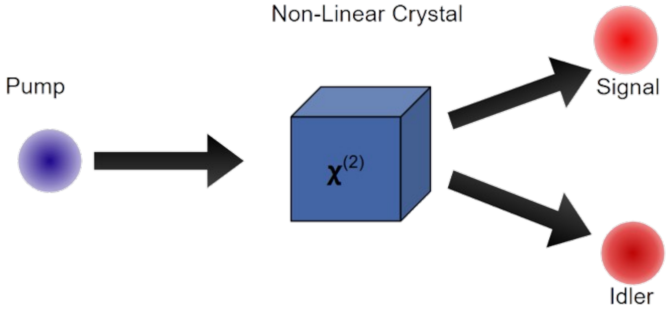


Figure 2.4: Down-conversion of signal and idler photons.

$$\vec{k_p} = \vec{k_i} + \vec{k_s} \tag{2.11}$$

.

The probabilistic down-conversion, as depicted in Figure 2.4 with a finite probability, is dependent on the crystal's non-linear coefficient $\chi^{(2)}$.

2.2.1 Phase-matching conditions

The dispersion relation of a medium is described by the Sellmeier equations, which in their most common form looks as follows:

$$n(\lambda) = A_0 + \frac{A_1\lambda^2}{\lambda^2 - B_1^2} + \frac{A_2\lambda^2}{\lambda^2 - B_2^2} \tag{2.12}$$

where Sellmeier coefficients A and B are determined experimentally.

Solving for the wave equation of a plane wave traveling in a non-linear medium with polarization P leads to Fresnel's equation. If one defines the ordinary and extraordinary direction of propagation with respect to the optic axis of a non-linear biaxial crystal, there are three different cases where the phase matching conditions are satisfied. They are listed as follows:

Type $0 : \mathrm{o} \rightarrow o + o, \quad e \rightarrow e + e,$

Type $\mathrm{I} : \mathrm{e} \rightarrow o + o, \quad o \rightarrow e + e,$

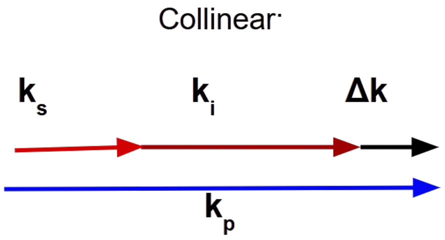


Figure 2.5: Collinear geometry for momentum conservation.

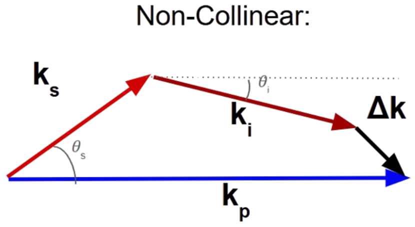


Figure 2.6: Non-collinear geometry for momentum conservation.

$$\text{Type} \quad \text{II : e} \rightarrow o + e, \quad o \rightarrow e + o.$$

When Δk in Figure 2.5 approaches zero, we say that the momentum equations are phase matched. One way to phase match is to cut and place the non-linear crystal such that the pump beam falls on the crystal at the exact angle θ such that it follows the relation

$$\vec{k_p} - \vec{k_s} - \vec{k_i} = 0. \tag{2.13}$$

This technique is referred to as birefringent phase-matching. Only certain crystals for which high birefringence compensates with dispersion precisely can show phase matching [30]. Barium borate (BaB_2O_4, or briefly BBO) and Bismuth borate (BiB_3O_6, or briefly BiBO) serve as good examples. Such crystals are angle tunable to meet the phase matching criteria.

In the case of three-wave mixing, the phase mismatch can be compensated for by an additional term in the equation:

$$\vec{k_p} - \vec{k_s} - \vec{k_i} - \frac{2\pi m}{\Lambda} = 0 \tag{2.14}$$

where m is an odd integer and Λ is the grating period of the crystal used.

This technique using periodically poled crystals is called quasi-phase-matching and is common with crystals like Lithium niobate ($LiNO_3$, or briefly LN) and Potassium titanyl phosphate ($KTiPO_4$, or briefly KTP) crystals. These periodically poled crystals are often temperature-tuned to match the phase-matching criteria.

2.2.2 Hamiltonian

If we define $a_s^\dagger$ and $a_i^\dagger$ as creation operators for signal and idler photons and $\hat{a_p}$ is the pump field, the Hamiltonian for the SPDC process can be written as

$$\hat{H} = \sum_{n=0}^{2} \hbar\omega(\hat{n_i} + \frac{1}{2}) + \hbar g[\hat{a_s}^\dagger \hat{a_i}^\dagger \hat{a_p} + h.c.] \tag{2.15}$$

where $\hat{n_i}$ is the number operator corresponding to the harmonic mode of the pump, signal, and idler photons, and g is a coupling constant directly proportional to $\chi^{(2)}$ of the crystal used.

If we treat pump mode $\hat{a_0}$ classically as a field with complex amplitude $a_0 = v_0 \exp(-i\omega_0 t)$ and use constants of motion for this phenomenon, defined as,

$$[\hat{n_1} + \hat{n_2} + 2\hat{n_0}, \hat{H}] = 0, \tag{2.16}$$

$$[\hat{n_1} - \hat{n_2}, \hat{H}] = 0. \tag{2.17}$$

One can find solutions to the time evolution of creation operators.

2.3 Hong-Ou-Mandel Interference

Optical setups provide a testing ground for assumptions of quantum mechanics. One of the reasons is that the polarization of a photon is directly correlated to the spin of a photon. And as Landau famously said, “There is no classical analogue of the spin of a particle. It is its intrinsic property” [31]. Thus, polarization of a single photon is a quantum property which can be exploited for verifying quantum mechanical assumptions.

Hong-Ou-Mandel (HOM) interference is a two-photon interference phenomenon. It has become a standard technique for testing particles' in-distinguishability [32], forms the basis for linear optical quantum computing [33], measurement-device-independent quantum key distribution protocols [34] and quantum sensing [35]. It is used to generate quantum entanglement, to elaborate the concept of the quantum eraser. It can also serve as a measure to distinguish quantum statistics of identical particles [36]. Consider the input quantum state on the beam splitter as $|\psi_{in}\rangle_{ab}$.

$$\begin{aligned} |\psi_{in}\rangle_{ab} &= \hat{a}_j^\dagger \hat{b}_k^\dagger |0\rangle_{ab} && (2.18) \\ &= |1:j\rangle_a |1:k\rangle_b && (2.19) \end{aligned}$$

where $\hat{a}_j^\dagger$ and $\hat{b}_k^\dagger$ are bosonic creation operators in beam splitter modes, a and b, respectively. The photons can be identified by their respective beam splitter modes in

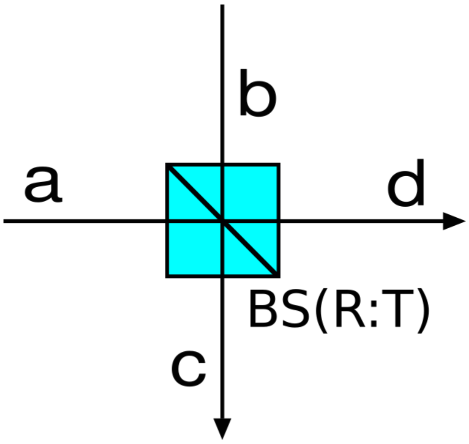


Figure 2.7: Input modes (a, b) and output modes (c, d) of a beam splitter

addition to subscripts j and k which are reserved for their distinguishability in other degrees of freedom, say polarization, arrival time, etc. The output modes c and d can be represented in terms of input modes a and b, as depicted in Figure 2.7 and can be mathematically described as:

$$\hat{c}_j^\dagger = \hat{U}_{\rm BS} \left|\psi^{\rm in}\right\rangle_{ab} = \sqrt{R}\hat{a}_j^\dagger + \sqrt{T}\hat{b}_j^\dagger \tag{2.20}$$

$$\hat{d}_k^\dagger = \hat{U}_{\rm BS} \left|\psi^{\rm in}\right\rangle_{ab} = \sqrt{R}\hat{a}_k^\dagger + \sqrt{T}\hat{b}_k^\dagger \tag{2.21}$$

The evolution of the input state $|\psi_{in}\rangle_{ab}$ under the beam splitter (with reflection and transmission coefficients represented as R and T) operation, $\hat{U}_{BS}$ results in the output state $|\psi_{out}\rangle_{cd}$ which can be written as:

$$\begin{aligned} \left|\psi^{\rm out}\right\rangle_{cd} &= \hat{U}_{\rm BS} \left|\psi^{\rm in}\right\rangle_{ab} \\ &= \left(\hat{c}_j^\dagger \hat{d}_k^\dagger |0\rangle_{ab}\right) \\ &= \left(\sqrt{R}\hat{a}_j^\dagger + \sqrt{T}\hat{b}_j^\dagger\right)\left(\sqrt{T}\hat{a}_k^\dagger - \sqrt{R}\hat{b}_k^\dagger\right)|0\rangle_{ab} \\ &= \left(\sqrt{TR}\hat{a}_j^\dagger\hat{a}_k^\dagger + T\hat{a}_k^\dagger\hat{b}_j^\dagger - R\hat{a}_j^\dagger\hat{b}_k^\dagger - \sqrt{TR}\hat{b}_j^\dagger\hat{b}_k^\dagger\right)|0\rangle_{ab} \end{aligned} \tag{2.22}$$

The above equation in the case of a 50:50 beam splitter can be represented as:

$$|\psi^{\rm out}\rangle_{cd} = \frac{1}{2}(\hat{a}_j^\dagger\hat{a}_k^\dagger + \hat{a}_k^\dagger\hat{b}_j^\dagger - \hat{a}_j^\dagger\hat{b}_k^\dagger - \hat{b}_j^\dagger\hat{b}_k^\dagger)|0\rangle_{ab} \tag{2.23}$$

HOM is bi-photon interference and the easiest observable parameter to detect is coincidence probability. Typically, if two photons are distinguishable in n degrees of freedom for a particular field mode, one needs a unitary matrix of the beam-splitter of order $2^n \times 2^n$. For simplicity, let's restrict to n=1. For the sake of experimental ease, we choose polarization and arrival time as modes of distinguishability, one at a time.

2.3.1 Distinguishability in Polarization

Let's consider the distinguishable parameters j and k in the polarization degree of freedom such that j=V and k=H falling on the input modes a and b on the beam splitter as shown in the Fig 2.8.

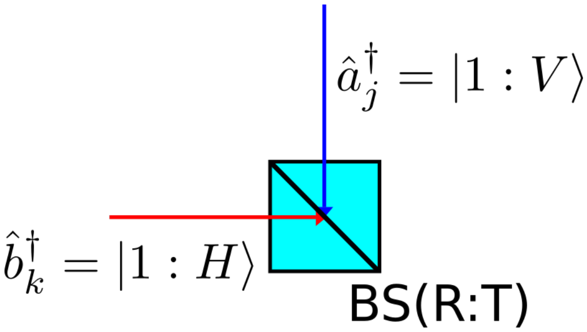

$\hat{a}_j^\dagger = |1:V\rangle$
$\hat{b}_k^\dagger = |1:H\rangle$
BS(R:T)

Figure 2.8: Input modes to a beam splitter with orthogonal polarization

It is assumed that the photons are identical in all other degrees of freedom. Then , the output state in Eq. 2.23 translates to

$$|\psi^{\text{out}}\rangle_{cd} = \frac{1}{2}(\hat{a}_V^\dagger\hat{a}_H^\dagger + \hat{a}_H^\dagger\hat{b}_V^\dagger - \hat{a}_V^\dagger\hat{b}_H^\dagger - \hat{b}_V^\dagger\hat{b}_H^\dagger)|0\rangle_{ab} \tag{2.24}$$

These four possibilities are highlighted in Figure 2.9.

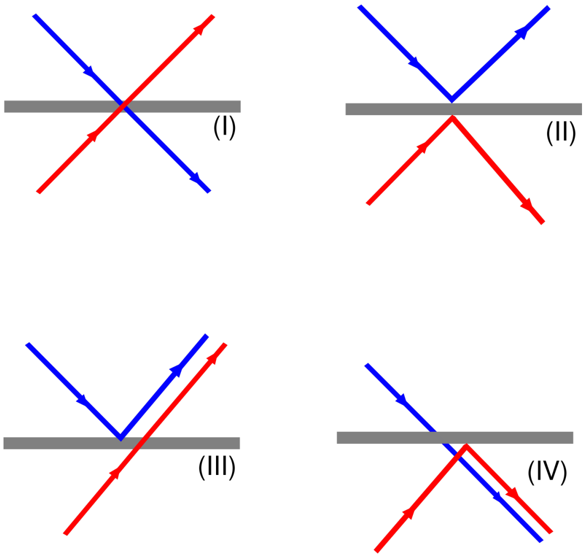


Figure 2.9: Four different possibilities for a typical beam splitter with two otherwise identical photons except polarization. The diagram can be read left to right with incoming photons impinging the beam splitter from left with difference of states in one degree of freedom and leaving from the right.

In the case of identical photons, the input state to the beam splitter becomes indistinguishable w.r.t. polarization as depicted in Figure 2.10.

As both the photons become identical in nature (i.e. j=k=H), the probability amplitude of two modes out of the four possible modes becomes identical and gets

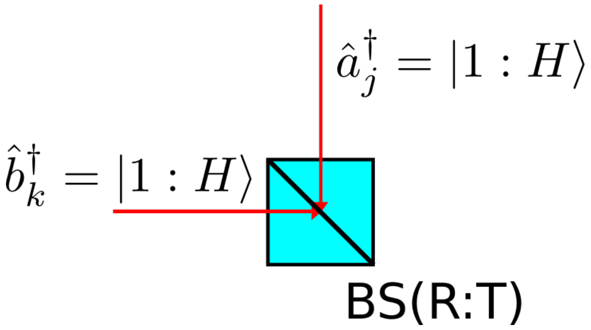

$\hat{a}_j^\dagger = |1:H\rangle$
$\hat{b}_k^\dagger = |1:H\rangle$
BS(R:T)

Figure 2.10: Identical input modes to a beam splitter with same polarization (identical in all degrees of freedom)

cancelled out, leaving only two possible outcomes, which, as shown in Figure 2.11, show the bunching phenomenon. In addition to being a verifier of bi-photon interference, it is also an indicator of the bunching phenomenon amongst photons.

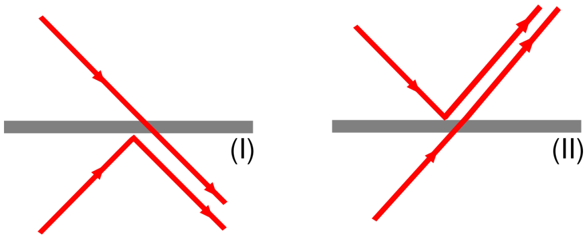


Figure 2.11: Identical Input modes to a Beam splitter with same polarization (identical in all degrees of freedom)

Hong-Ou-Mandel interference can be used to reproduce quantum eraser experiments like Young's double-slit experiment with the difference that here, polarization is chosen as a degree of freedom instead of position.

2.3.2 Arrival Time

Similar to the case of polarization, an analogy can be drawn for the arrival time of photons. Let the two wavepackets containing identical photons (with each wavepacket containing a single photon) arrive at the beam splitter, at times, t_1 and t_2 such that $\tau = t_1 - t_2$.

For such a case, the wave-functions can only be resolved in their arrival times, only if $\tau \leq \tau_{coh}$, and are otherwise indistinguishable. The cases are depicted in Figure 2.12. For such a case, similar to the case of polarization, the four possibilities when the two wave packets are distinguishable reduce to two possibilities with the probability amplitudes getting canceled out from Eq. 2.23.

One can experimentally realize the drop in coincidence counts as a measure of probability amplitude getting cancelled out by measuring the coincident counts as a function of time or path delay. As a sample, the experimental results are shown in Figure 2.13. The visibility V of the HOM curve is analogous to the classical visibility

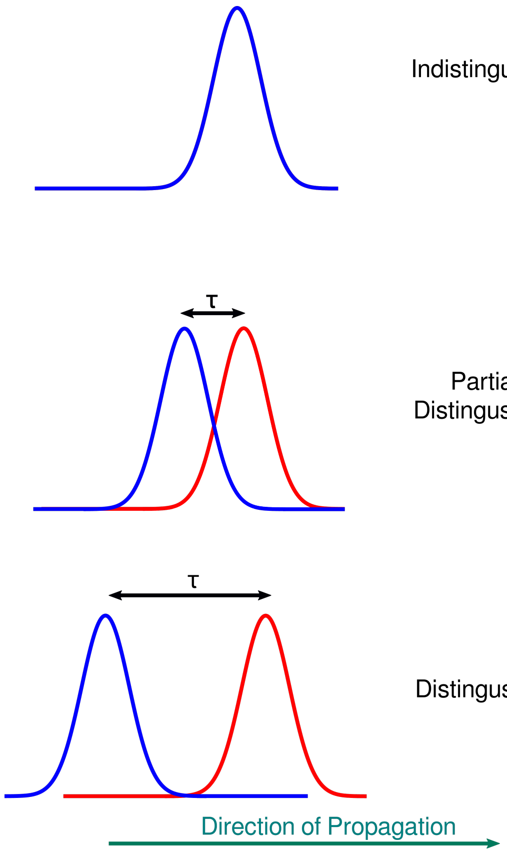


Figure 2.12: Wave-packet overlap (a measure of distinguishable nature) as a function of time delay τ between them

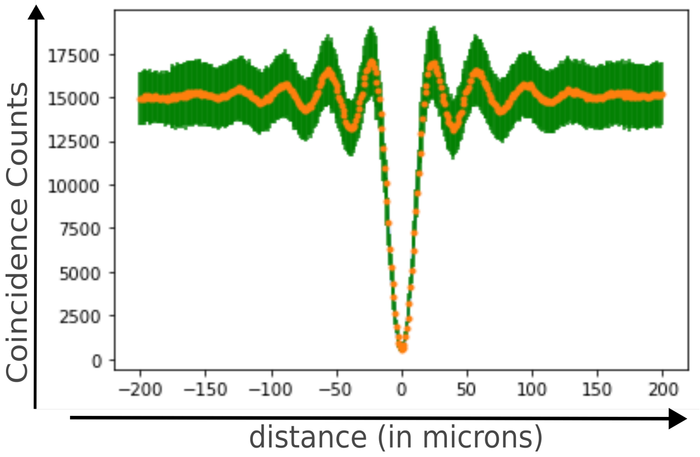


Figure 2.13: HOM curve displaying coincidence counts as a function of relative path delay between photons

and is defined as

$$V = \frac{N_d - N_i}{N_d + N_i} \tag{2.25}$$

where $N_d(N_i)$ are total coincidence counts corresponding to distinguishable and indistinguishable cases.

2.4 Quantum Entanglement

The aforementioned SPDC process can be used to generate quantum entanglement. Typically, the above method is used to generate bi-partite quantum entanglement with polarization degree of freedom. The quantumness in this method is due to the pair of single photons. Using a discrete variable degree of freedom, one can generate bi-partite quantum entanglement. As in classical mechanics or optics, one treats two separate particles in distance as independent; instead, quantum entanglement is a phenomenon which shows the existence of non-local correlations despite the arbitrary distance. Unlike polarization, one can also generate higher dimensional quantum entangled states with orbital angular momentum (OAM) degree of freedom and exploit the dynamics involved with qutrits or qudits in general. The word "Verschränkung" coined by Erwin Schrödinger in 1935 [37] and translated to entanglement describes statistical interactions between two subsystems of a composite system. To classify these interactions, Einstein, Podolsky and Rosen (EPR) through a thought experiment doubted the fundamental assumptions of reality and localism to quantify these interactions. In 1964, John Bell formulated an inequality that must be satisfied by the Local Hidden Variable Theory (LHVM) as an alternate theory to quantum mechanics [38]. The violation of this inequality implies that only quantum theory can explain certain kinds of interaction as proposed by the gedanken EPR experiment.

2.4.1 Generation

There are multiple ways to engineer quantum entangled photons using SPDC photon pairs. Here, we restrict ourselves to the typical case of quantum entangled photons in polarization. In this section, we gently touch upon different methods of generation. As shown in Figure 2.14, four different generation techniques are mentioned. For case (a), the photon pair can be generated using a Type-I crystal where the interference of orthogonally polarized photons from the input ports of the beam splitter generates quantum entangled pairs after post-selection. For case (b), the two orthogonally polarized photon pairs (engineered by SPDC crystals in combination with HWPs) at the two input ports of a polarizing beam splitter can be superposed to generate an entangled state. Superposition can also be achieved by spatial overlap from beam walk-off within birefringent crystals (Figure 2.14(c)). Spatial filtering can be used to remove any residual spatial distinguishability. Although it leads to a loss of photon counts, its single-line alignment makes it a good bet against an ideal entangled photon source. Before moving to case (d), we must understand that SPDC is a broadband process (in wavelength) which allows phase-matching over a broad spectrum of wavelengths. In many source configurations, the generated SPDC output travels through dispersive materials where the photon pairs will pick up a wavelength-dependent phase. Methods of temporal compensation using birefringent crystals such as Yttrium orthovanadate, (YVO_4) can correct for such errors as shown in case (d).

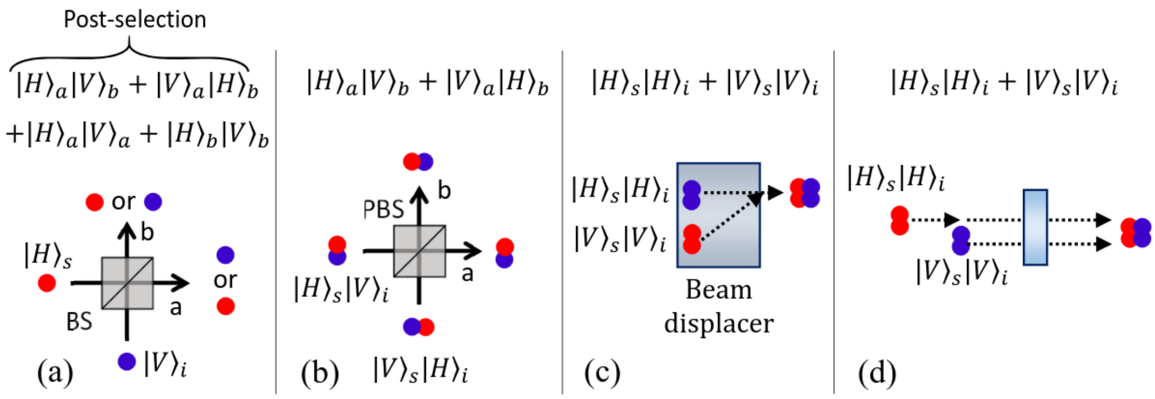

Post-selection
$|H\rangle_a|V\rangle_b + |V\rangle_a|H\rangle_b$
$+|H\rangle_a|V\rangle_a + |H\rangle_b|V\rangle_b$
or
b
$|H\rangle_s$
or
a
BS
(a)
$|V\rangle_i$
$|H\rangle_a|V\rangle_b + |V\rangle_a|H\rangle_b$
PBS
b
a
$|H\rangle_s|V\rangle_i$
(b)
$|V\rangle_s|H\rangle_i$
$|H\rangle_s|H\rangle_i + |V\rangle_s|V\rangle_i$
$|H\rangle_s|H\rangle_i$
$|V\rangle_s|V\rangle_i$
Beam displacer
(c)
$|H\rangle_s|H\rangle_i + |V\rangle_s|V\rangle_i$
$|H\rangle_s|H\rangle_i$
$|V\rangle_s|V\rangle_i$
(d)

Figure 2.14: Four major techniques for generation of polarization entangled photon-pairs using SPDC sources, Image sourced from Ali Anwar et al. [2]

2.4.2 Quantifiers for bipartite system

There are many measures of quantifying quantum entanglement. We discuss the two most fundamental measures on which all other measures are based, namely density matrix based measures and Bell's inequality.

2.4.2.1 Calculating Density Matrix from Quantum State Tomography

As Pauli basis transformations can be experimentally realized using a combination of HWPs and QWPs, we use Pauli basis to represent our density matrix. For a single-qubit, the density matrix can be represented as a function of Stokes parameter, S_i and Pauli basis σ_i such that

$$\hat{\rho} = \frac{1}{2}\sum_{i=0}^{3} S_i\hat{\sigma}_i \tag{2.26}$$

The set of four Stokes' parameters that describe the polarization state of a single qubit can be given a physical meaning as they are represented as $\mathrm{S}_0 = P_{|H\rangle} + P_{|V\rangle}$ $S_1 = P_{|H\rangle} - P_{|V\rangle}$ $\mathrm{S}_2 = P_{|D\rangle} - P_{|A\rangle}$ $S_3 = P_{|R\rangle} - P_{|L\rangle}$

The above argument can be extended from single qubit to two-qubit tomography such that

$$\hat{\rho} = \frac{1}{2^2} \sum_{i,j=0}^{3} S_{ij} \hat{\sigma_i} \otimes \hat{\sigma_j} \tag{2.27}$$

where $S_{ij} = S_i \otimes S_j$ where S_i have been described previously.

Due to experimental imperfections, the obtained density matrices are noisy and thus, constraints of physicality are placed in an ad-hoc manner. Namely, these constraints are

- The density matrix should be normalised i.e. Tr($\hat{\rho}$)=1
- The density should be real i.e. Hermitian or in other words $\hat{\rho}^\dagger = \hat{\rho}$
- The density matrix should be positive semi-definite i.e. all the eigen values should be non-negative.

In Chapter 4, we discuss the procedure to get rid of errors(experimental errors corresponding to a non-physical density matrix) and measures that can be calculated based on the density matrix.

2.4.2.2 Bell Inequality

Typically, Bell's parameter, S, is used as a verifier for quantum entanglement in a bi-partite system defined by the correlation coefficient $E(a, b)$. The goal is to show entanglement has quantum correlations which cannot be attained by change of basis. The vectors a and b are chosen from the basis formed by vectors (a,$a_\perp$) and (b,$b_\perp$) where vectors a and b differ by $\pi/8$ and the relation between all four vectors, namely a, a', b and b', is highlighted in Figure 2.15.

$$S \equiv E(a, b) - E(a, b') - E(a', b) + E(a', b') \tag{2.28}$$

The correlation coefficient can be represented in terms of experimental counts, i.e. coincidence counts N(α, β) where α and β are the angles of the polarizers used in making measurements of the entangled qubits [39]. In general, four different measurements (α, β), $(\alpha_\perp, \beta)$, $(\alpha, \beta_\perp)$, $(\alpha_\perp, \beta_\perp)$ are required for the calculation of $E(\alpha, \beta)$. With the four such correlation coefficients, the number of experimental measurements typically required becomes sixteen.

$$E(\alpha, \beta) = -\frac{N(\alpha, \beta) - N(\alpha_\perp, \beta) - N(\alpha, \beta_\perp) + N(\alpha_\perp, \beta_\perp)}{N(\alpha, \beta) + N(\alpha_\perp, \beta) + N(\alpha, \beta_\perp) + N(\alpha_\perp, \beta_\perp)} \tag{2.29}$$

and

In a typical case where classical correlations are present, S is upper-bounded as

$$S \leq 2. \tag{2.30}$$

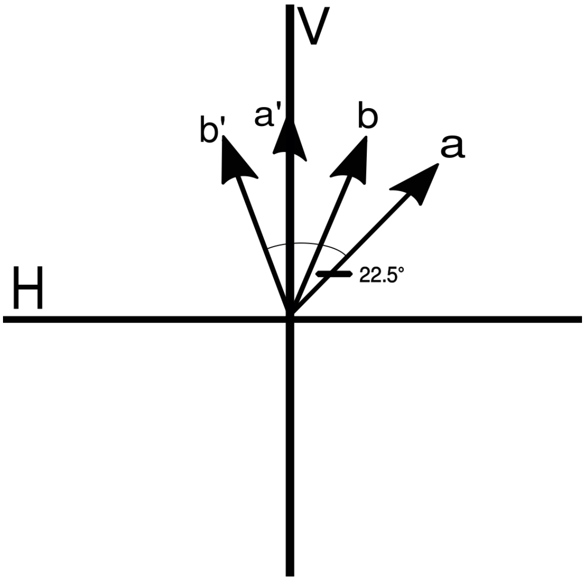


Figure 2.15: Different basis vectors to check the independence of basis property of quantum correlations

However, Bell showed that for any local hidden variable theory to sustain as a fundamental theory, must satisfy these inequalities. However, it was later shown that if the inequality is violated, then the local hidden variable theory is unable to explain these correlations and any theory that could demonstrate the inequality violation is more fundamental in nature. In 1981, building on the work of Clauser [40] with his famous experiment, Alain Aspect [41] demonstrated Bell inequality violation and proved that quantum mechanics is a more fundamental theory and no local hidden variable theory could prove all predictions of physical reality. As there is no other theory which is as verifiable as quantum mechanics, the burden of being the fundamental theory falls on the shoulders of quantum mechanics. Being a probabilistic theory in nature based on Heisenberg's uncertainty relations, it has inherent randomness associated with it. Quantum mechanics has been able to show a Bell violation, with optics as a test-bench, and saturates Tseirlson's bound of $2\sqrt{2}$. One direction in the current literature is to perform loophole-free Bell measurements [42] and the other is to generate random numbers based on quantum contextuality [43].

2.5 Comparing random numbers generated from single photon sources

The aim is to study the dependence of randomness of QRNGs on the single photon nature of the two different sources. One is the weak coherent pulses from a laser diode attenuated to single photon level, characterised by μ, and produces single photons probabilistically. However, it also produces multi-photon events with some small probability. The other source is a heralded single photon source from the Spontaneous

Parametric Down Conversion (SPDC) process generated in a non-linear β-Barium Borate (BBO) Type-1 crystal. Theoretically, it is a better single photon source but is still probabilistic in nature. However, it is also comparatively resource-intensive. Thus, we study the effect of random number generation for these different single photon sources as shown in Figure 2.16.

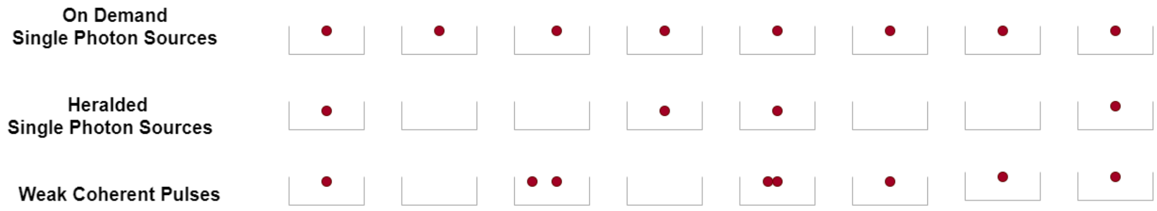


Figure 2.16: Different kinds of single photon sources (a) Ideal (Deterministic) (b) Non-Ideal (Probabilistic) (c) Non-Ideal (Probabilistic) with some multi-photon events

2.5.1 QRNG based on Weak Coherent Pulses

A coherent laser is attenuated to a single-photon level. Once the beam is attenuated to a single-photon level, it falls on a beam splitter. The attenuated beam follows Poissonian statistics. The single photon has a probability to either quantum mechanically tunnel through the separated gap (gap between the two prism mirrors of the beam splitter) to get transmitted or reflected from the prism mirror. For a 50:50 beam splitter, the beam has an equal probability to either get reflected or transmitted. It can be seen as the beam is in superposition of both the states (the two states are both the output ports of the beam splitter). Also, one should note, once the beam is attenuated, the single photon has a probability to lie anywhere within the beam waist (spatially) of the laser beam. (This technique itself could be used as another variant of discrete-variable optical QRNG. It requires more detectors and thus becomes expensive and therefore isn't a wise choice). Without digressing, it works as a simple random number generator depending on whether the single photon is reflected (labeled as "0") or transmitted (labeled as "1"). This method generates a random number bit-string of zeroes and ones. The quantum mechanical process that a single photon cannot be further divided exploits the inherent assumption that the photon being the fundamental particle. The randomness of the beam splitter variant of discrete variable optical QRNGs (DV-OQRNGs) is better compared to those exploiting photon statistics of the single photon or its time of arrival. Primarily, because the process of generating randomness depends on the time window and thus bias gets introduced, unlike the beam-splitter based QRNGs where bias (in the form of deviation from 50:50) can be characterized.

2.5.1.1 Experimental Setup

We first discuss the experiment with weak coherent pulses (WCP) from a laser diode source. The experimental set-up is described in Figure 2.17. Coherent weak laser is attenuated using a variable optical attenuator (VOA).

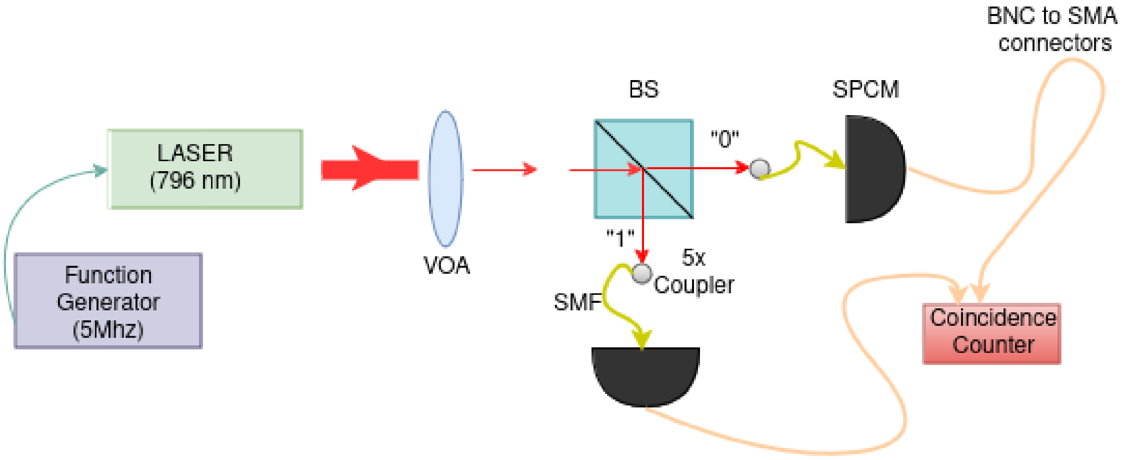


Figure 2.17: Experimental set-up for random number generators using weak coherent pulses; VOA: variable optical attenuator, BS: Beam Splitter

However, there are certain things to be careful about. One, since WCP source has multi-photon events which we must remove before generating random numbers out of it. We should also take into account the background. In our experiment, we used WCP pulses of laser diode at 796 nm with a spectral bandwidth of 10 nm. The laser diode is driven at a repetition rate of 5 MHz. In time domain, it translates to pulses at regular intervals of 200 ns.

2.5.2 QRNG based on heralded single photons from an SPDC source

The pump photons falling on the non-linear β-BBO crystal interact with vacuum fluctuations and get obliterated. This interaction effectuates the generation of two down-converted photons of half the frequency. Single photons are generated from correlated pairs of down-converted photons by conditioning the detection of one of the photons (heralding). Unlike the weak coherent laser, it is a strictly single-photon source. The heralded single photons fall on the 50:50 beam-splitter and thus generate a random number sequence depending on whether the photon is transmitted (labeled as "0") or reflected (labeled as "1") through the beam-splitter. Because we constantly have to condition for the single-photon nature, it experimentally translates to coincidences for the reflected and transmitted part with the heralded arm.

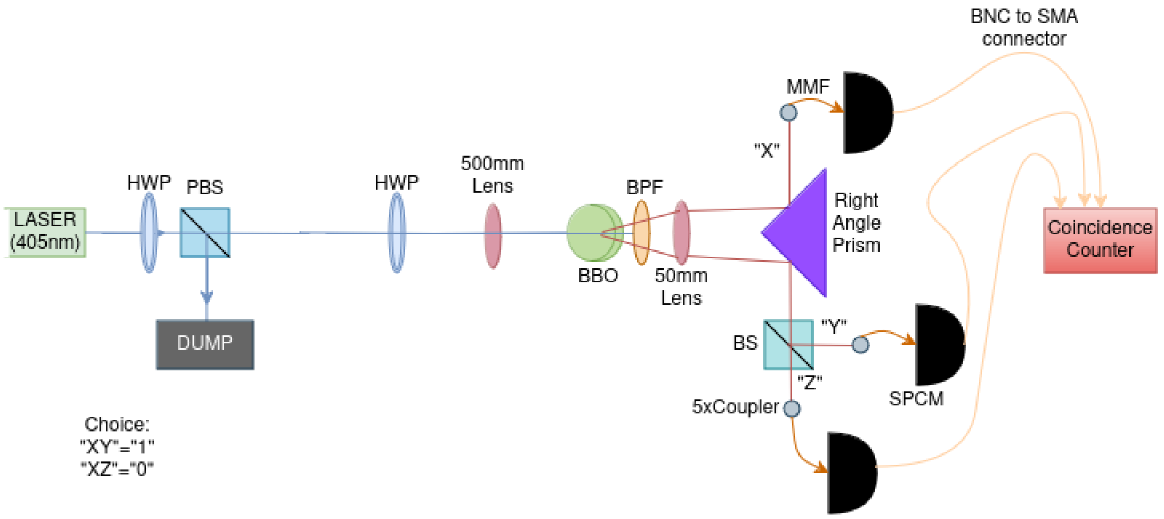


Figure 2.18: Experimental set-up for generating random numbers using heralded single photon sources through SPDC process; HWP: half wave plate, PBS: polarizing beam splitter, SPCM: Single photon counting module, SMF: single-mode fiber, MMF: multi-mode fiber, BPF: band-pass filter, BBO: Barium Borate

2.5.2.1 Experimental Setup

Laser emits a Gaussian beam at a wavelength of 405 nm. Initially, a half wave plate (HWP) and a polarizing beam splitter (PBS) combination are used to have control over the intensity of the beam. The output of the PBS (with an extinction ratio of 1000:1) is horizontally polarized light. Another HWP is used to control the polarization of the beam. A 50 cm lens is used to loosely focus the beam on the non-linear crystal. Loose focusing ensures a large Rayleigh range. The two down-converted photons, namely signal and idler, come out at the diametric ends of the SPDC cone and are collimated using a 5 cm lens. The pump is blocked by a bandpass filter of 10 nm centered at an 810 nm wavelength. The filter fixes the coherence length of the SPDC photons to 32 μm. Both signal and idler are separated by a prism mirror, and the signal is collected in the 5x coupler. It goes through multi-mode fiber into the single photon counting module (SPCM), an avalanche photo-diode (APD) detector working in Geiger mode. It generates a current signal of substantial amplitude that acts as the clock of the ID900 device to generate a time-stamp of the detected photon. The idler falls on the beam splitter, and it has an equal probability of getting reflected or transmitted. The coincidence time window to check for the heralded pairs is fixed at 3.6 ns. Coincidences between XY arms and XZ arms are checked to generate a random bit sequence of zeros and ones. The triple coincidence detection corresponds to multi-photon events from SPDC or accidental background counts. These are removed before generating a random number bit-string. The experimental setup is shown in Figure 2.18.

2.5.3 Comparative analysis of SPDC and WCP based random numbers

We compare both the generation methods, WCP and SPDC based SPS passing through the beam splitters. Intuitively, one knows that SPDC is a better single photon source. Since the same beam splitter is used in both setups, the quality of the single photon source should reflect in the quality of randomness generated. As a measure to calculate the randomness quality, we use entropy as a parameter. It is discussed in greater detail in Chapter 3. We use two different measures, namely Shannon entropy ($H_1(X)$) and min-entropy ($H_\infty(X)$), and calculate their difference. In an ideal case, i.e. for an on-demand single photon source, its difference should follow the equation $(1-1=0)$. For our experimental results, we see that the difference for WCP based random numbers is $H_1(X) - H_\infty(X)$=0.0975 and for SPDC based random numbers is $H_1(X) - H_\infty(X)$=0.05375. The physical implication of the above result is two-fold. Since the difference for SPDC is much closer to zero, it reinforces the fact that SPDC is a better single photon source. Second, it confirms our intuition that the quality of randomness is directly related to the quality of the single photon source. We build on this intuition by calibrating the SPDC source and randomness measures against different parameters.

2.6 Towards a relationship between single photon source and randomness

We executed the setup, as shown in Figure 2.19, considering all the drawbacks encountered in our earlier experiments. We changed the crystal from BBO to pp-KTP as the latter has a higher non-linear coefficient. We checked the variation of single photon nature with time delay (τ), pump power P and OAM order (l).

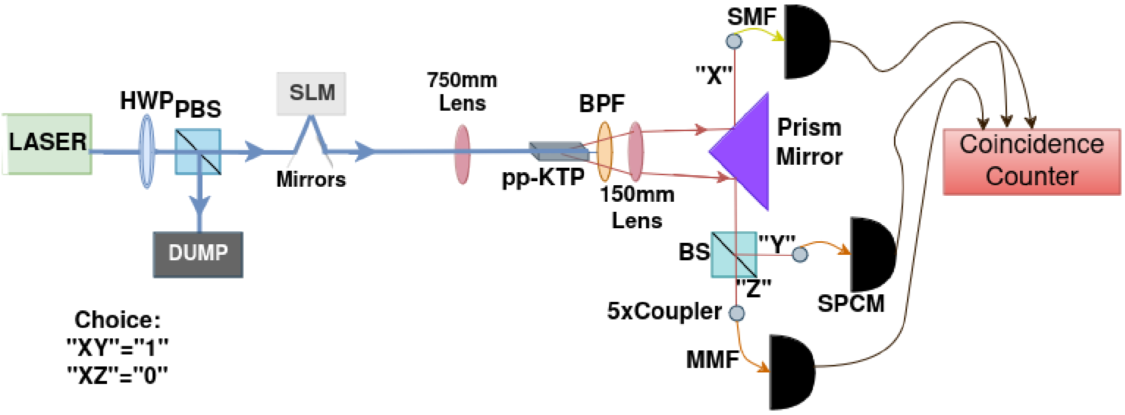


Figure 2.19: Setup for SPDC based QRNG; HWP: half wave plate, PBS: polarizing beam splitter, SLM: spatial light modulator, SPCM: Single photon counting module, SMF: single-mode fiber, MMF: multi-mode fiber, BPF: band-pass filter

2.6.1 Quality of single photons and quality of randomness variation with time delay

We tried to evaluate the relationship between the quality of randomness, $H_\infty(X)$ and the quality of single photon nature, $b = 1 - g^2(0)$. This relationship is investigated by varying the arm length Y with respect to arm length Z as shown in Figure 2.19. The variation of arm length Y is modeled by varying the time delay in the Y arm w.r.t. arm Z before generating the time stamp in the time to digital converter (TDC). There seems to be a strong correlation between the two quantities i.e. $b = 1 - g^2(0)$ and $H_\infty(X)$ as we vary the time delay. The graph in Figure 2.20 depicts the same.

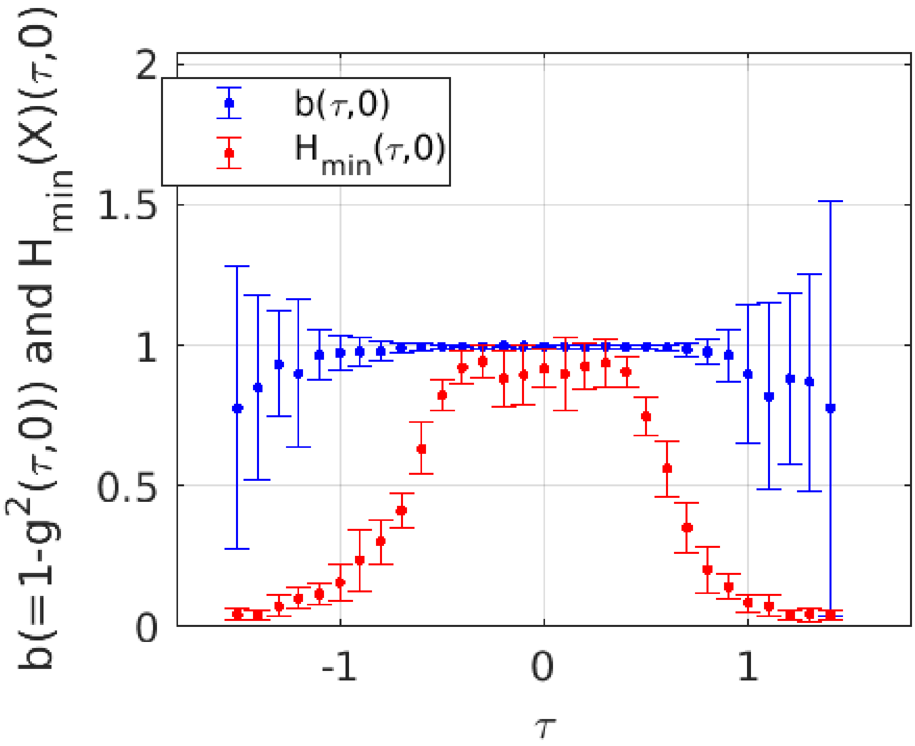


Figure 2.20: Variation of single photon nature parameter $g^2(0)$ and randomness parameter $H_\infty(x)$ with time delay τ (in ns)

2.6.2 Quality of single photons and quality of randomness variation with pump power

Next, we evaluate the relationship between the quality of randomness, $H_\infty(X)$ and quality of single photon nature, $b = 1 - g^2(0)$ by varying the pump power. No clear evidence of variation of both parameters is seen with pump power. Although pertaining to the rise of multi-photon events with a rise in pump power, it is expected that both parameters will decrease with a rise in pump power. However, for this small range of power, the ratio of multi-photon events is negligible compared to single photon events. Hence, both parameters appear independent of power as depicted in the graph in Figure 2.21.

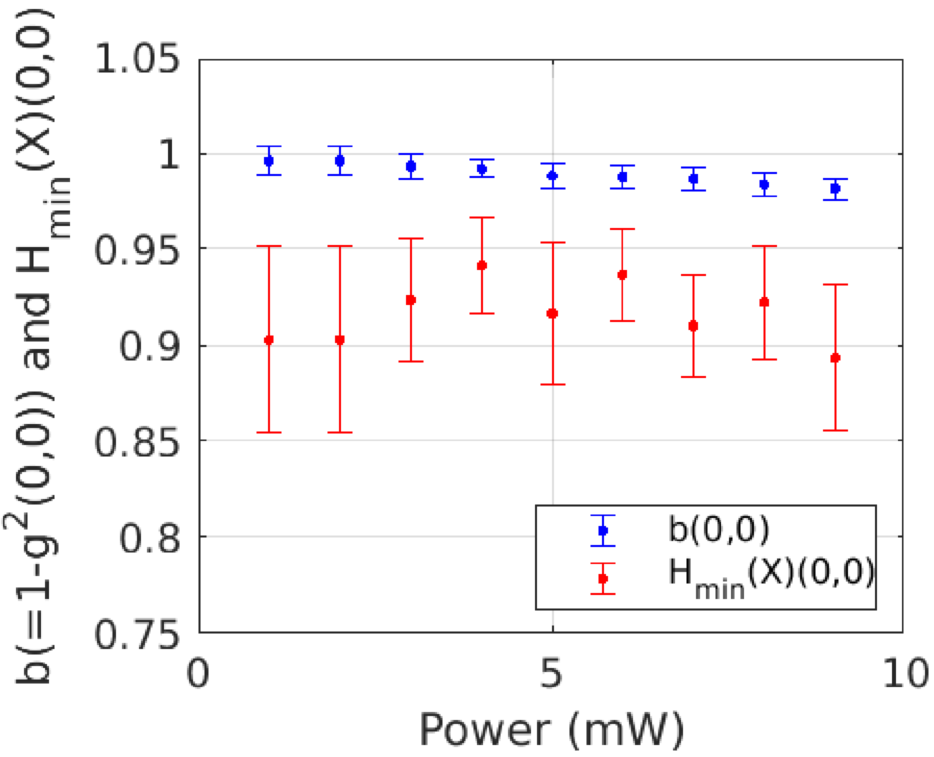


Figure 2.21: Variation of single photon nature parameter $g^2(0)$ and randomness parameter $H_\infty(x)$ with pump power P

2.6.3 Quality of single photons and quality of randomness variation with orbital angular momentum (OAM)

Furthermore, we tried to evaluate the relationship between the quality of randomness, $H_\infty(X)$ and the quality of single photon nature, $b = 1 - g^2(0)$ by varying OAM values. The motive of this graph was to understand how the effect of multi-photon events corrupts the quality of both parameters. The effect of multi-photon events is too low on the quality of random numbers or on the quality of single photon events and thus, is ignorable. For a pump power of 1 mW, the effect is negligible. As the pump power increases, the generation of multi-photon increases. The graph in Figure 2.22 shows negligible variation of both parameters at 10 mW. This variation is small, analogous to the small corruption of single photon nature by multi-photon events of the non-linear process.

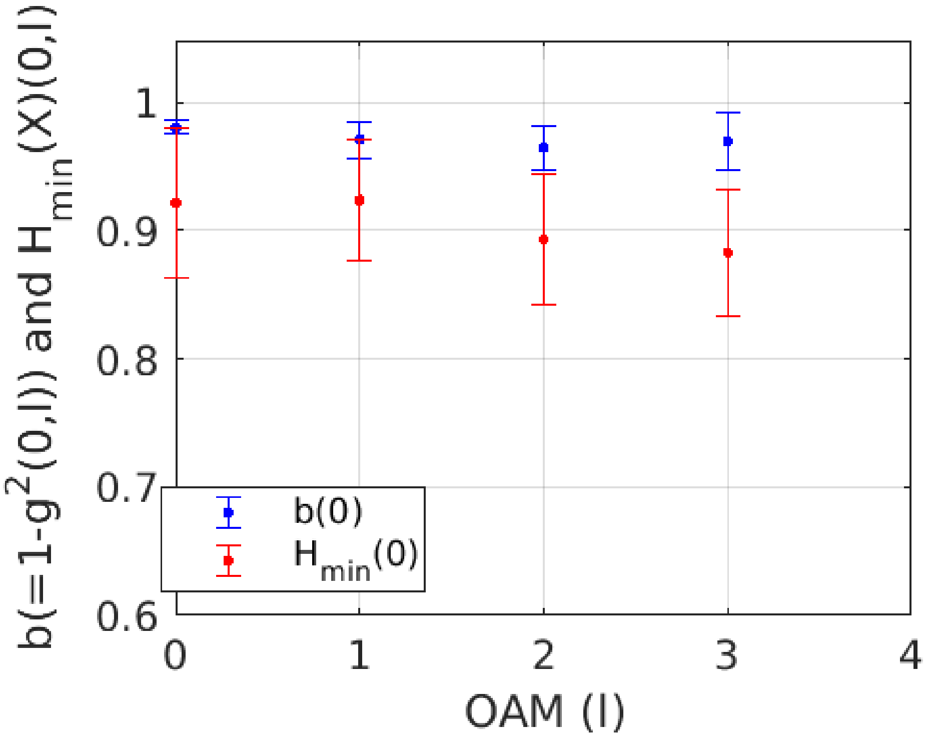


Figure 2.22: Variation of single photon nature parameter $g^2(0)$ and randomness parameter $H_\infty(x)$ with z component of OAM

2.7 Summary

In this chapter, we have built the foundational tool-kit to measure quantum phenomena for QRNGs. It included the generation of quantum entanglement, Hong-Ou-Mandel interference, and quantum state tomography. Next, we have discussed the comparative analysis for weak coherent pulses based and SPDC-based random numbers. It is followed by further exploring the relationship between single photon nature and randomness (with min-entropy as a metric). Both examples cover the discrete variable QRNGs from Figure 1.3 which fall under the category of trusted devices. In Chapter 4, we will discuss the experimental setup for a device-independent QRNG.

"Random numbers should not be generated with a method chosen at random."

Knuth, Donald

3

Computational tool-kit for random number generation

In this chapter we focus on two different computational aspects. One, we discuss the conventional methods, specifically PRNGs and CS-PRNG. Secondly, we discuss computational post-processing methods based on the type of computational sources used in the generation of bit-streams. QRNGs fall under the category of unpredictable sources, which puts constraints on the type of post-processing methods involved to break the biases introduced in the QRNG due to experimental imperfections. To check the independence of statistical correlations, we use NIST-STS as a measure to pick out the weaklings amongst random number generators. We discuss entropy as a measure of randomness(unpredictability). We discuss why min-entropy is a good measure after we see different generalizations of the entropy both in the information and quantum information domains. Towards the end, we discuss one specific PRNG for two reasons. One, to develop an intuition behind the working of the flow of methodology for generation, post-processing, and characterizing PRNGs. This is done to draw a similar analogy for QRNGs. Secondly, to use it for cryptographic applications in quantum cryptography.

3.1 Conventional computational methods: PRNGs

The mathematical definition of PRNG is written as follows:

Programming Language	**Library**	**PRNG type**
C	rand()	Linear Congruential Generator (LCG)
C++	random	Mersenne Twister (MT-19937)
Python	random	Mersenne Twister
Python	secrets	Noise based
Python	numpy.random	PCG-64
Java	base::RNGkind	MT-19937
PHP	rand()	LCG
Ruby	mt rand()	MT-19937
MATLAB	randn	Ziggurat algorithm (based on MT-19937)
R	runif()	MT-19937 and others

Table 3.1: PRNGs used in different libraries with different programming

A deterministic function $G : \{0,1\}^m \rightarrow \{0,1\}^n$ is a (d,ϵ) pseudo-random number generator (PRNG) if

1. m $\leq$ n, and

2. G(U^d) and U^m are (d, ϵ) indistinguishable.

where ϵ is the computationally distinguishability parameter between G(U^d) and U^m.

Since random functions are ubiquitously called from libraries in programming languages which serve plenty of applications ranging from encryption, Monte-Carlo simulation of a physical phenomenon to providing universal unique identifier on the internet, etc. It can be said to be the heart of the internet. Table 3.1 showcases PRNGs used in specific libraries of everyday programming languages.

We discuss two fundamental PRNGs whose computationally complex variants are used in standard libraries. These are Linear Feedback Shift Register based PRNG and Linear Congruential Generator based PRNG.

3.1.1 Linear Feedback Shift Register

Amongst pseudo-random number generators (PRNGs), one of the fast and easy-to-generate known techniques for generating random numbers is linear feedback shift registers (LFSRs). LFSR consists of shift registers whose input bit is a linear combination of its previous bits at each stage. It is typically computed using bit-wise XOR operations and is often defined by a feedback polynomial. LFSR, a type of sequential digital circuit, utilizes clock-driven flip-flops. The electronic circuit comprises a collection of d flip-flops and requires a seed of d-bits. If appropriate taps (XORing position of bits) are chosen, LFSR produces a sequence of bits that appear random with periodicity of $(2d - 1)$ [44]. The operational concept of the LRNG is illustrated in Figure 1. For ease of understanding, we use the convention $\mathcal{L}(d, s)$ to define d-bit LRNG where d is a number of flip-flops and s is a d-bits initial seed value.

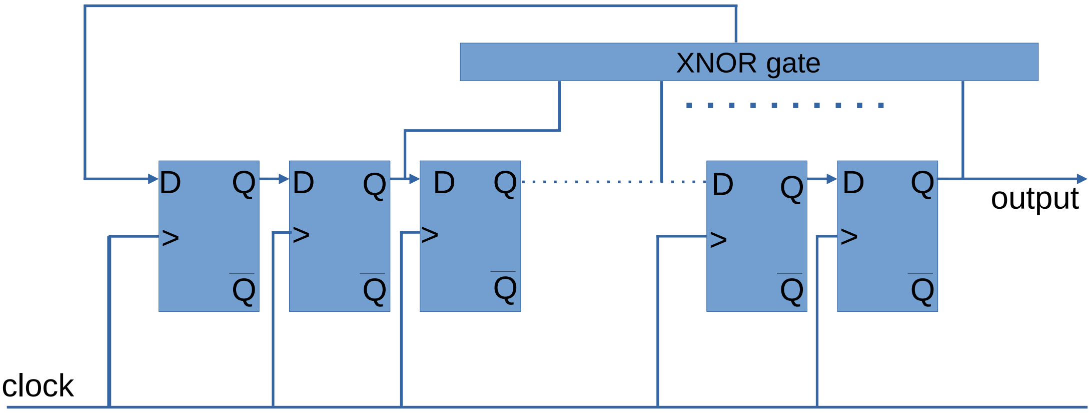


Figure 3.1: Technique of generating LRNG using cascaded flip flops and XNOR.

An LFSR polynomial of $d = 32$ has suitable tap positions for maximum bit-length generation, namely 32,30,26,25. The primitive polynomial (mathematical equivalent of generating maximum length for LFSR) can be written as

$$\mathcal{L}(d) = x^{32} + x^{30} + x^{26} + x^{25} + 1. \tag{3.1}$$

There are two types of LFSR based on the type of XOR operation: Galois field-based and Fibonacci numbers-based. Galois field-based has an easier software implementation, while Fibonacci-based has an easier hardware implementation. This is primarily because Fibonacci-based LFSRs don't require additional 'AND' operations and, thus, can easily be woven into the fabric of field programmable gate array (FPGA). Being FPGA hardware-friendly, they are used in testing for power consumption in integrated circuits. In particular, it is typically used to create test patterns for built-in self-test [45]. On the low power front, LFSR-based random number generators (LRNGs) are widely used in providing encryption to wireless technology such as Bluetooth [46]. Additional benefit lies in the fact that the output is in TTL pulses, which can be easily integrated to any other system further. Such random number generators are easily implementable on hardware like (FPGA) [47] and, thus, could provide easy access to "randomness" resource.

However, a single LFSR is completely insecure as a PRNG since given n consecutive bits of its output it is trivial to compute all subsequent bits. However, by combining several LFSRs with the addition of a non-linear component, it is possible to get secure PRNGs.

3.1.1.1 Stream Ciphers: Adding non-linearities to LFSR process

One approach to building stream ciphers from LFSRs is to run several LFSRs in parallel and combine their output using a non-linear operation. Such ciphers are lightweight and therefore makes them suitable for operating at a high speed in power constrained devices. The Content Scrambling System (CSS) is a system used for protecting movies on DVD disks. This cipher combines two LFSRs using addition over the integers. Another stream cipher, A5/1 is used to encrypt global system for mobile communications (GSM) cell phone traffic is based on combining the outputs of three independent LFSRs [48]. The Bluetooth E_0 stream cipher combines four LFSRs using a 2-bit finite state machine. The SNOW 3G cipher and it's predecessors which are used to encrypt 3GPP cell phone traffic operate a little differently. They run a single LFSR and generate the output from a non-linear operation on it's internal state. All these algorithms have been shown to be insecure and should not be used for cryptographic applications. In simpler words, recovering the plain-text takes very little time than a search on the complete key space for these ciphers. Other stream ciphers like Trivium, Sprout, based on LFSR, are used in common day practice these days.

3.1.2 Linear congruential generator

Linear congruential generator (LCG) is one of the fundamental building blocks of couple of cryptographically secure PRNGS such as Blum-Blum Shub [49], elliptic curve cryptographic techniques [50]. The recurrence relation generating random numbers for LCG is defined as

$$X_{n+1} = (aX_n + c) \mod m \tag{3.2}$$

where X_n represents the sequence of random numbers, and m, a, and c are integer constants which represent the modulus, multiplier, and increment of the generator, respectively. It is used to model stochastic processes via markov chain model and thus finds its applications in simulations.

3.1.3 Chacha20 Cipher

Many cryptographically secure PRNGs (CS-PRNGS) like Fortuna, ISAAC, Chacha20 have replaced their predecessors like Blum-Blum-Shub (based on LCG). However, when it comes to faster implementation, nothing comes closer than the ones built on microprocessor commands such as addition modulo two and exclusive OR operations. For example, Chacha20 is one of the computationally secure PRNGs (CS-PRNGs) built on the stream cipher Salsa20 [51]. Oftentimes, the security of these stream cipher (or derived PRNGs) is based on computational hardness as a resource. The same resource defines the hardware implementation. Hence, there is a trade-off between security of stream cipher or random number built on it and the speed with which they can be run. As per Bernstein [51], ChaCha20 has good properties i.e. it provides higher confidence in cipher security, while being consistently faster than AES on machines without hardware accelerators [52]. Figure 3.2 demonstrates the generation of a 512-bit chunk stream. Firstly, the initial 512-bit state of ChaCha20 must be formed. This state consists of a few fields, namely, constants, key, block count and nonces. Furthermore, the initial state is transformed by 20 quarter round functions of two types: horizontal and diagonal. Each round function converts its 512-bit input into a 512-bit output. Finally, the initial state is added to the output of the last round function [53]. For the summation stage, both operands are seen as arrays of 32-bit words with 16 elements (described as a 4 × 4 matrix).

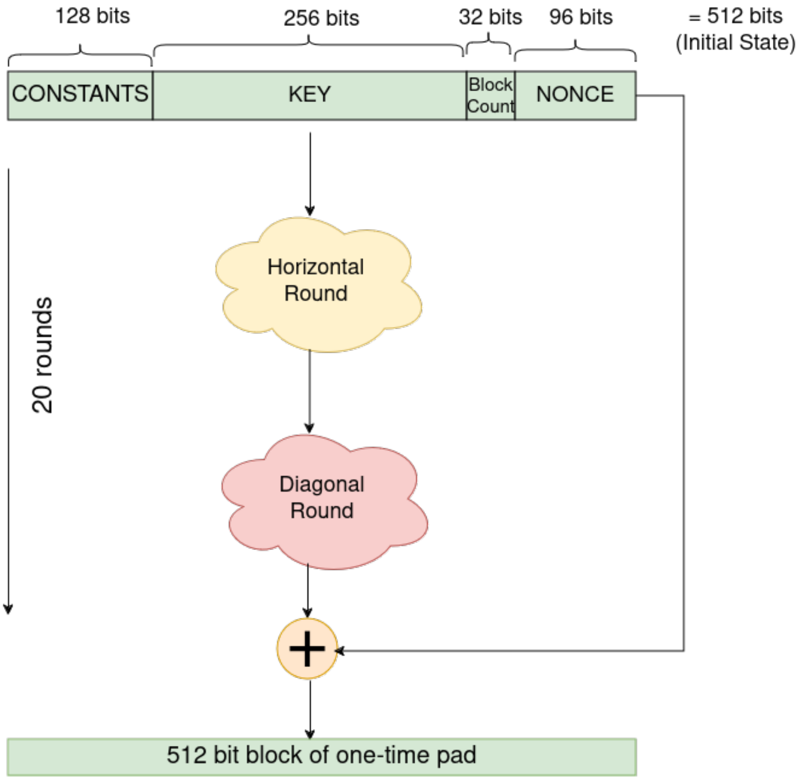


Figure 3.2: Working of ChaCha20 algorithm: 80 quarter rounds of mixing

Almost every stream cipher has two input parameters: a key k and an initialization vector IV . The key is a regular one that is used in every symmetric key cryptography. The IV acts as a randomizer and should take a new value for every encryption session; otherwise, the stream cipher becomes highly deterministic. This ensures that the same input key can be used multiple times to get different output key-streams. The IV can be chosen to be kept open (not secret) and is often referred to as nonces (”number used once”). QRNGs can step into the role of key or nonces if needed.

The components of 512 bit key are depicted in Figure 3.2 and described below:

- **Constants**: As the name suggests, they form the constant part of the key. It is 128 bits (or 16 bytes) in length and is initialized with the ASCII values of the characters of the following string: ‘expand 32-byte k’.

- **Key**: This is the secret input key. It is 256 bits (or 32 bytes) in length in this case (it can be 128 bit as well). Once user gives an input of 256 bit key, this key is expanded using the algorithm which can serve both as a cipher or as a PRNG source as per the needed requirement.

- **Block Count**: A counter that starts from 0 and is incremented to generate the next chunks of OTP. There are 2^{32} unique values of this field, so the maximum length of the entire OTP with the same secret key is 256 GB. It takes up 32 bits(or 4 bytes) in length of the input.

- **Nonce**: They are 96 bits(or 12 bytes) in length. A unique number that can be changed to generate a new OTP with the same key. Using each value of nonce no more than once is crucial for providing a high level of security.

To generate the next 512-bits of the PRNG bit-stream, the block count field of the initial state is incremented and the process repeats. The final result of ChaCha20 is extremely sensitive to changes of the initial state: flipping even a single bit of the input leads to an unpredictable change of the result (like a butterfly effect). Moreover, despite the round functions being reversible, it is impossible to convert the result of ChaCha20 back to the initial state, because of the summation stage. This makes it suitable for stream encryption. Figure 3.3 shows the structure of the quarter round. The input of the quarter round consists of four 32-bit words taken from the input of the round. Column and diagonal rounds supply the input to the quarter rounds depending upon the row or diagonal of the previously defined 4*4 matrix. What makes ChaCha20 simple to implement is that each quarter round is made up of three basic operations which can be run easily on any microprocessor:

1. **Add (A)**: Carry-less addition of two numbers: $(a + b)$ mod 2^{32}

2. **Rotate (R)**: Rotation of a 32 bit number by a fixed number of positions. R - X represents rotation by X number of positions.

3. **XOR (X)**: Exclusive OR between two 32 bit numbers: a $\oplus$ b

Together, all three operations form an ARX cell. The ARX cell is highlighted in Figure 3.3.

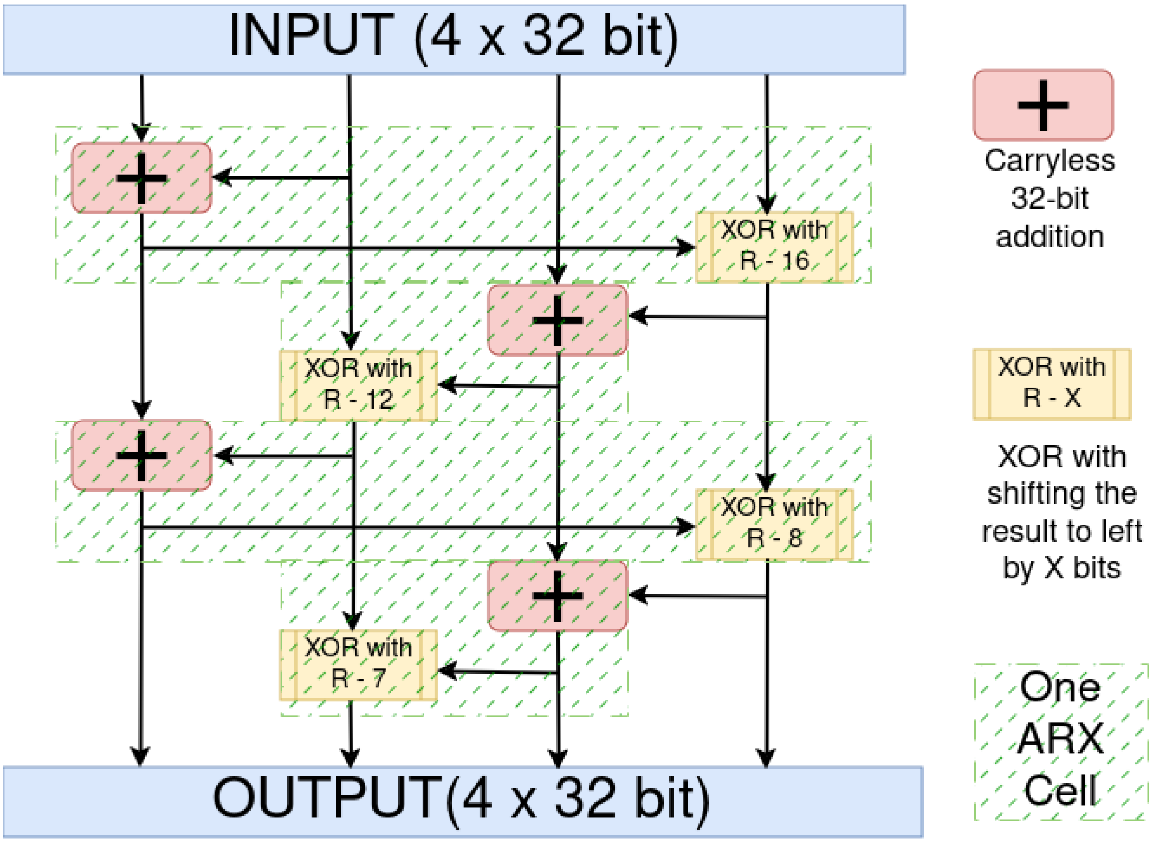


Figure 3.3: Graphical Illustration of quarter round in Chacha20 algorithm

3.2 Post-Processing Methods

3.2.1 Classification of Extractors

Randomness extractors are a class of functions which are used to extract randomness from weak sources of randomness. The output of these extractors is an almost uniform distribution bit-stream. The goal of these extractors is to remove biases from the system. As the name post-processing suggests, the best way to get rid of biases is to redistribute the bias evenly amongst all bits so that all bits are equally likely to have the same bias. From a computer science perspective, they are also used in samplers and expander graphs.

3.2.2 Calibration Metric: Entropy

The entropy of a random variable X is a measure of information gained once we know the random variable X. To quantify it, we define Shannon entropy with a probability distribution $\mathrm{p_x}$ associated with random variable x,

$$H(X) \equiv H(p_1, ..., p_n) \equiv -\sum_{\mathrm{x}} p_{\mathrm{x}} log_2(p_{\mathrm{x}}) \tag{3.3}$$

3.2.2.1 Rényi Entropy

Rényi-entropy is defined as

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \sum_x P_X(x)^\alpha \tag{3.4}$$

for $\alpha \in (0,1) \cup (1,\infty)$. Many useful entropies such as min-entropy and collision entropy can be defined using different α values to quantify processes like information leakage and quality of privacy amplification in Quantum Key Distribution (QKD) respectively. For $\alpha = 1$, the limit yields the Shannon entropy (can be verified using L'Hospital's rule). In other words, Rényi entropies are a generalization of Shannon entropy.

The min-entropy of random variable X is defined as

$$\begin{gathered} H_\infty(X) = -\log p_{\text{guess}}(X) \\ \text{where } p_{\text{guess}}(X) = \max_x P_X(x) \end{gathered} \tag{3.5}$$

Consider the probability distribution

$$P_X(x) = \begin{cases} \frac{1}{2} & \text{for } x = 1 \\ \frac{1}{2(|X|-1)} & \text{else} \end{cases} \tag{3.6}$$

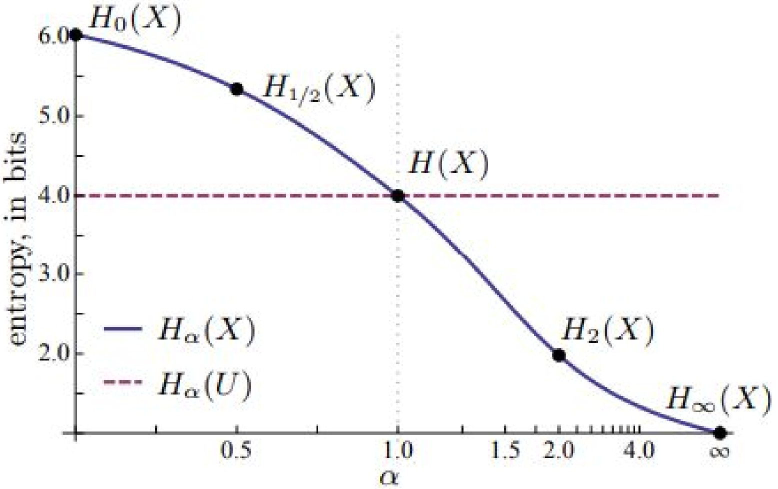


Figure 3.4: Rényi entropies of X with probability distribution as in equation 3.6 with $|X| = 65$ (blue line) compared to a uniform random variable U on 4 bits (red line). [3]

Relative Rényi entropy is a further generalization of Rényi entropy. This new quantity [54] compares two different distributions p and q and is defined as

$$D_\alpha(p_X \| q_X) := \frac{1}{\alpha - 1} \log_2 \left(\sum_{X \in \mathcal{X}} p_X(x)^\alpha q_X(x)^{1-\alpha} \right) \tag{3.7}$$

where $\alpha \in (0,1) \cup (1,\infty)$

Special cases of this quantity cover mutual information and Rényi entropy as shown in the equations below and hence are the most appropriate quantities to study eavesdropping of any quantum channel.

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \sum_x P_X(x)^\alpha = -D_\alpha\left(p_X \| 1\right) \tag{3.8}$$

$$I_\alpha(X,Y) := \inf_{q_Y \in \mathcal{P}(\mathcal{Y})} D_\alpha\left(p_{XY} \| p_X \otimes q_Y\right) \tag{3.9}$$

For $\alpha = 1$, Equations 2 and 3 reduce to

$$H(X) = \sum_x p_x \log p_x = -D\left(p_X \| 1\right) \tag{3.10}$$

$$I(X,Y) := \inf_{q_Y \in \mathcal{P}(\mathcal{Y})} D\left(p_{XY} \| p_X \otimes q_Y\right) \tag{3.11}$$

where $\alpha \in (0,1) \cup (1,\infty)$ and $D\left(p_X \| q_X\right)$ are the relative entropies between two probability distributions.

3.2.2.2 Properties of classical Rényi relative entropy

1. $D_\alpha\left(p_X \| q_X\right) \geq D_\alpha\left(N_{Y|X}\left(p_X\right) \| N_{Y|X}\left(q_X\right)\right)$

   The first property of relative Rényi entropy is that they satisfy data processing inequality. Physically, it means that the entropy of the system $D_\alpha\left(p_X \| q_X\right)$ doesn't increase after passing through a classical channel. Classical Channel, $N_{Y|X}$, is defined as

   $q_Y(y) = \sum_{x \in \mathcal{X}} N_{Y|X}(y \mid x) q_X(x)$

2. $D_\alpha\left(p_{X_1} \otimes p_{X_2} \| q_{X_1} \otimes q_{X_2}\right) = D_\alpha\left(p_{(X_1)} \| q_{(X_1)}\right) + D_\alpha\left(p_{(X_2)} \| q_{(X_2)}\right)$

   The second property defines the additive property of two independent measures in different spaces.

3. For $\alpha > \beta > 0$, $D_\alpha\left(p_X \| q_X\right) \geq D_\beta\left(p_X \| q_X\right)$

   This is also known as the ordering property.

3.2.2.3 Relation between entropy and post-processing

The calculation of entropy is an important measure to keep an eye on how much post-processing is required and how much ϵ security a given system can provide. This ϵ security is comparable for PRNGs and QRNGs given the type of random number generator used. Compared to PRNGs, QRNGs provide security to a much larger depth by calibrating the process of generation rather than the output bit-stream with min-entropy. Typically, the entropy is related to post-processing length, l, and ϵ security by the formula,

$$l = \frac{H_{min} * n}{B} - 2\log_2(\epsilon_{hash}) \tag{3.12}$$

Since all the parameters ranging from initial bit-stream n, ϵ_{hash} security, bits per symbol B, and post-processed length l are intertwined, we can compare the entropies directly to the security random numbers provide. For a fixed post-processing length, bits per symbol and initial bit-stream, one can say that larger the min-entropy, the ϵ_{hash} security is exponentially larger.

For quantum process characterisation, one needs to calculate the quantum equivalent of the min-entropy and since Rényi entropy is a generalisation of min-entropy. Researchers [55, 56, 57], have drawn an analogous formalism for the quantum regime where the random variable X can now be replaced by the density matrix and in the case of diagonalisable matrices, the density matrix formalism reduces to the classical definition of a random variable.

3.2.2.4 Quantum Rényi relative entropy

The quantum generalisation of relative entropy is defined completely analogous to the classical case.

$$D(\rho\|\sigma) := \operatorname{Tr}\left[\rho\left(\log_2\rho - \log_2\sigma\right)\right] \tag{3.13}$$

where ρ and σ are density matrices corresponding to two quantum states. If these quantities are diagonal in a particular basis, they reduce to the classical relative entropy and density matrices reduce to probability distributions. Also, one should note that the quantum relative entropy reduces to special cases of von Neumann entropy and quantum mutual information analogous to the classical case. Also, the quantum channel is defined analogously to the classical channel by replacing density matrices for probabilities.

Further generalization of this quantum relative entropy is quantum Rényi relative entropy. Currently, the research area in this direction is still in progress. It suggests three different generalizations, with all of them satisfying some basic properties. These different generalizations are enumerated below.

1. Petz Rényi relative entropy is defined as

$$D_\alpha(\rho\|\sigma) := \frac{1}{\alpha-1}\log_2\operatorname{Tr}\left[\rho^\alpha\sigma^{1-\alpha}\right] \tag{3.14}$$

   where $\alpha \in (0,1)\cup(1,\infty)$.

2. Sandwiched Rényi relative entropy is defined as

$$\widetilde{D}_\alpha(\rho\|\sigma) := \frac{1}{\alpha-1}\log_2\operatorname{Tr}\left[\left(\sigma^{(1-\alpha)/2\alpha}\rho\sigma^{(1-\alpha)/2\alpha}\right)^\alpha\right] \tag{3.15}$$

where $\alpha \in (0,1) \cup (1,\infty)$.

3. Geometric Rényi relative entropy is defined as

$$\widehat{D}_\alpha(\rho\|\sigma) := \frac{1}{\alpha-1}\log_2 \mathrm{Tr}\left[\sigma\left(\sigma^{-1/2}\rho\sigma^{-1/2}\right)^\alpha\right] \tag{3.16}$$

where $\alpha \in (0,1) \cup (1,\infty)$.

All the entropies defined do satisfy data processing inequalities. In addition to the certification of quantum resources, these entropies can be used to characterize classical and quantum channels.

3.2.3 Types of bit-stream Sources

Based on the quality and method of the bit-stream generated, the need and type of extraction for post-processing are chosen. We classify the bit-stream on the computational front to aptly choose the post-processing method. Hence, the discussion on post-processing follows the said classification of bit-streams. Consider the bit-stream,

$$10101010101010101010 \tag{3.17}$$

Such a bit-stream is predictable deterministically or has no randomness associated with it. On the other end, we have an IID (independent and identically distributed) bit-stream which is independent but biased. However, the bias, say δ, is constant and with simple post-processing methods can be evenly distributed. The second kind of sources is bits that are independent but not identically distributed. In such a case, the bits are biased and the bias is uneven. Parity of block-based extractors can remove these uneven biases at the cost of reducing the length of the bit-stream. Such extractors fall under the category of deterministic extractors which are discussed in the following section. The third kind of sources are Santha-Vazirani sources [58]. These sources are unpredictable in nature and thus, also known as unpredictable in nature. As the name suggests, the bias in these bits is not only even but also dependent on previous bits. This makes the bias unpredictable to track. In these cases, seeded extractors come in handy which help in redistributing this bias evenly amongst all bits such that no bits remain biased greater than ϵ. Such extractors can be used to simulate many problems of larger complexity classes.

3.3 Randomness Extractors

3.3.1 Deterministic Extractors

Deterministic Extractors, C, are those which can map $DE : \{0,1\}^c \rightarrow \{0,1\}^d$ such that for every X $\in$ C, Ext(X) is ϵ close to U_m. In simple terms, extractors that behave deterministically in nature. The output bit-stream can be mapped from the input.

3.3.1.1 Von Neumann extractor

Von Neumann extractor[59] is built to remove systematic unknown biases. It is one of the first and simplest extractors named after its developer. Consider a bit-stream of sequences $X_1, X_2, X_3, X_4, ..., X_n \in [0,1]$ whose bit-stream follows an independent and identically distributed (i.i.d.) condition. However, if the probability of ones is biased by δ, i.e. P$[X_i = 1]$=δ for some unknown δ. To remove such systematic biases, he proposed to go to a higher dimension by pairing the bit-streams in combination of two (dimension d=2) bits, say, 00, 01, 10 and 11. This is followed by a reduction to one dimension by drawing an equivalence between binary bits and equally biased pairs, 01 and 10. The drawback of the extractor is that it puts a very stringent condition on bit-streams, namely i.i.d. and hence is not widely applicable.

3.3.1.1.1 Impossibility of a Universal Deterministic Extractor

: [58] informs us that there does not exist a deterministic extractor that can extract even a single bit from unpredictable sources. This is sad news for the approach of simulating randomized algorithms [60] and protocols with weak sources by deterministic extraction. Historically, this motivation led to the notion of seeded extractors which are described in the next section.

3.3.2 Seeded Extractors

In cases where bias is unpredictable and dependent, i.e. Santha-Vazirani sources [61], one needs an external good source of randomness to extract randomness from the initial poor randomness source (Santha-Vazirani source). That's where seeded extractors fit in. As the name suggests, seeded extractors require an external seed. The quality of this seed together with the construction of the extractor defines the overall quality of the seeded extractor. The goal to use this extractor is to scramble the biases of the initial source such that the bias is evenly redistributed and the source becomes comparatively unbiased. The better the scrambling procedure and the quality of randomness of the seed, the better is the randomness of the output bit-stream. A fun fact is that it can also work for cases where bias is predictable.

Thus, a sequence of complete zeroes can turn into a random number generator after the post-processing via seeded extractor.

3.3.3 Toeplitz Extractor

The Toeplitz hash function comes from a family of two universal hash functions. The input weak random sequence (X) (could be a PRNG or a QRNG) is written as a column vector of length $n \times 1$. The hashed output of the matrix multiplication(M $\times$ X) as shown below is stored in column vector Y of shorter length $l \times 1$ as shown in Equation 3.18 (illustrated in matrix form beneath it).

$$U_{l\times 1} = M_{l\times n} \cdot A_{n\times 1} \tag{3.18}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ . \\ . \\ y_l \end{pmatrix} = \begin{pmatrix} 1 & 0 & . & . & . & . & . & 1 \\ 0 & 1 & . & . & . & . & . & 0 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ 1 & 1 & . & . & . & . & . & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ . \\ . \\ 1 \end{pmatrix}$$

Typically, it is assumed that the seed chosen is uniformly random. If it is, the biasedness is evenly redistributed and thus, the weak source of random number generators becomes a strong one. This process appears like a feedback loop in the sense that we use random bits to generate random bits. However, there is a catch. The length of the seed required in this process is much smaller than the length of random numbers generated. The expenditure comes at the cost of the bit-rate of extractor-applied-QRNG. It gets shortened from length n to l as illustrated in Figure 3.5.

3.4 Testing Mechanisms

We discuss statistical test suites for evaluating the independence of statistical correlations between the bit-streams (a typical measure of randomness). In this regard, we discuss tests described in NIST-STS. There are other good alternatives like Dieharder, ENT, TestU01, but we don't see an advantage of QRNGs (pertaining to experimental imperfections) in that direction and limit ourselves to a typical standard.

3.4.1 NIST-STS

NIST has provided test batteries to check the quality of random numbers. The quality check means that the statistical correlations are evaluated under hypothesis testing

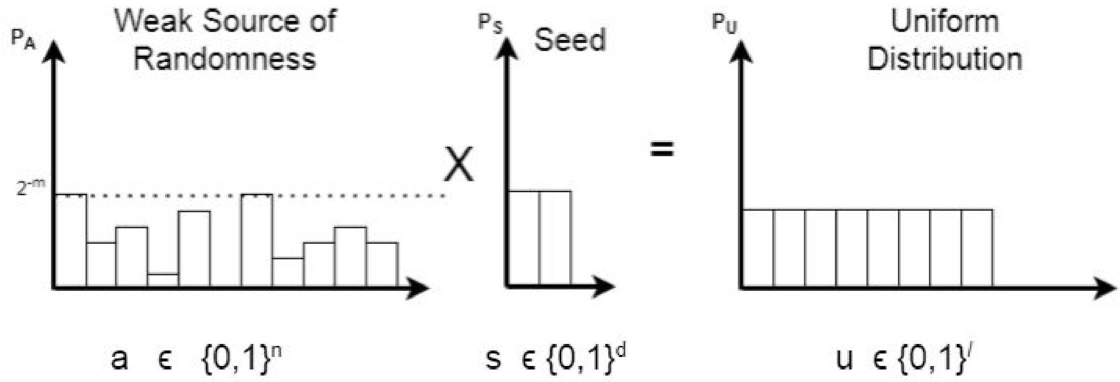


Figure 3.5: Biasedness of a weak source of randomness, generated from a QRNG, redistributed with the aid of another short seed based uniform distribution to generate an almost uniform random number sequence of shorter length. Here, n is the length of input QRNG, d is length of the seed used to generate extractor's input, l is the length of the output (extractor-applied-QRNG) and m is the min-entropy ($H_\infty(A)$) of the source. The y-axes correspond to the probability distributions for different block lengths.

that the said sequence is random. The test suite is a hypothesis testing suite where H_0 is the null hypothesis that the sequence is random while H_a is an alternate hypothesis that the sequence is not random. In the suite, the focus is on Type-I and Type-II errors as depicted below in the Table 3.2 and obtained from [62].

True situation	Accept H_0 (Not-Significant)	Accept H_a (Significant)
H_0 is true	No error	Type-I
H_a is true	Type-II	No error

Table 3.2: Test static p value is equivalent to α where α is a threshold chosen for Type-I errors.

One can assign a threshold α and β for Type-I and Type-II errors, respectively. The test-static p value defines the α value, and α and β are connected. Thus, keeping a check on one is enough, and for NIST-STS, it is α. Lower α values mean higher β values.

3.4.1.1 List of Tests Included

- **Frequency Test**: This test checks for the equi-probability of bits 0 and 1 on the entire set. The test is a derivative of the well-known Central Limit Theorem for the random walk [63] .
- **Block Frequency Test**: The test seeks to detect localized deviations from the ideal frequency of 1's by decomposing the test sequence into a number of non-overlapping subsequences and applying a chi-square test for a homogeneous match of empirical frequencies to the ideal 0.5 [63].

- **Runs' Test**: The non-parametric test looks at "runs" which are defined as substrings of consecutive 1's and consecutive 0's. It considers whether the oscillation in such homogeneous sub-strings changes abruptly (i.e. becomes too fast or too slow).

- **Longest Run of one's in a block**: This test checks for the length of the longest consecutive sub-sequence (run) of ones in a block to evaluate statistical correlations of random bit string.

- **Binary Matrix Rank Test**: The test checks for linear dependency between fixed-length substrings of the original sequence and repeats the process until the entire sequence is verified.

- **Discrete Fourier Transform Test**: This test detects periodic statistical correlation type features in the Fourier domain of the bit series.

- **Non-Overlapping Template Matching Test**: This test checks the sequences which display too many or too few occurrences of a given set of a-periodic patterns. One also needs to ensure the independence of these a-periodic patterns.

- **Overlapping Template Matching Test**: This test rejects sequences which show too many or too few occurrences of m-runs of ones.

- **Maurer's Universal Statistical Test**: This test is designed to detect any one of the very general class of statistical defects that can be modeled by an ergodic stationary source with finite memory. Broadly, it is based on the fact that a universal statistical test can be based on a universal source coding algorithm. A generator should pass the test if and only if its output sequence cannot be compressed significantly.

- **Linear Complexity Test**: To ward off against any combination of LFSR based generating mechanisms or the possibility of using the linear complexity characteristic for testing randomness is based on the Berlekamp-Massey (BM) algorithm, which provides an efficient way to evaluate finite strings.

- **Serial Test**: The (generalized) serial test represents a battery of procedures based on testing the uniformity of distributions of patterns of given lengths.

- **Approximate Entropy Test**: The test is used to approximate entropy characteristics based on repeating patterns in the string.

- **Cumulative Sums Test**:The test is based on the maximum absolute value of the partial sums of the sequence.

- **Random Excursions Test**: This test is based on considering successive sums of the binary bits (plus or minus simple ones) as a one-dimensional random walk. The test detects deviations from the distribution of the number of visits of the random walk to a certain "state," i.e., any integer value.

- **Random Excursions Variant Test**: As the name suggests, this test is a variant of random excursions test based on random walk.

3.4.1.2 NIST Variants

Because tests like non-overlapping template matching test, random excursions and its variant have multiple p-values, it becomes difficult to assign meaning to these hypotheses. To overcome this issue, Sidak proposed a correction factor to give an overall p-value for all the test suites. The correction factor was empirically calculated to be 1.1 × $min(p - value)$ amongst all 15 tests.

3.4.2 Dieharder

Dieharder [64] is a random number test suite built on its predecessor Diehard [65]. Similar to NIST-STS, it checks for the quality of random numbers by checking the bit-stream for statistical correlations routed via chi-square hypothesis testing. With a few exceptions listed below, Dieharder tests are analogous to NIST Statistical Test Suite.The different tests include parking lot test, birthday spacing test, the craps test, minimum distance test, etc.

3.4.3 Algorithmic Methods

Randomness has two different faces. One is statistical randomness, which is outcome-focused. One can look for the unpredictability of the next bit-stream based on the statistical correlations of the previous n bit-streams. The notion of algorithmic randomness is on parallel lines to this direction of inquiry. Algorithmic randomness of a bit-stream focuses on the computability of the algorithm on a universal Turing machine. As per the definition of algorithmic randomness, random numbers are classified into four different categories, as highlighted in the diagram below.

3.4.3.1 Kolmogorov Complexity

For example, the well-known RSA algorithm used in cryptography trades hardness for randomness between computational complexity classes bounded-error probabilistic polynomial time (BPP) and polynomial time (P). Thus, different complexity classes of computation lead to different classes of randomness. We take the general case of NP-hard problem and trade its hardness for randomness to get a general definition of algorithmic randomness. One such measure of randomness is Kolmogorov Complexity. Kolmogorov Complexity is defined as the length of the shortest program to mimic the computation of a given program.

3.4.3.2 Borel Normality

A sequence is said to be Borel Normal, as defined by Emile Borel in 1909, if the occurrence of every digit string of length m in a base r has equal frequency for each m in the sequence [66].

3.5 LFSR based RNGs and their NIST Statistical Testing

In this section, we test the quality of LFSR against NIST Statistical test-suites. It's a zero-sum game between the generation and testing scheme. Here, we show how post-processing using XOR operations can pass the NIST statistical test suites for the quality of random bits generated and, as described in the following section, can be used on FPGA for QKD applications.

3.5.1 Quality check of NIST

We discuss four different variants of LFSR. We discuss how the post-processing methods get complex before they pass NIST test suites if we use simple XOR operations for post-processing. This highlights the need for choosing the appropriate post-processing method for the pseudo random numbers generated.

Variant 1: $\mathcal{L}(d, s)$

Here, the shift register $\mathcal{L}(d, s)$ is used to generate the maximum length bit-sequence $(2^d - 1)$ and is tested for quality against NIST-STS. We observed that only in the case of $d = 128$, 14 out of 15 NIST tests are passed. However, the LCT test fails. All 15 tests failed for smaller d values. The results are described in Table 3.3.

d-bit	NIST-tests	LCT
8	Failed	Failed
16	Failed	Failed
24	Failed	Failed
32	Failed	Failed
64	Failed	Failed
128	Passed	Failed

Table 3.3: Test result for LRNG $\mathcal{L}(d, s)$

Results for the case of $\mathcal{L}(d, s)$ puts faith in the quality of the randomness test suite. The failure $d = 128$ is clear as the bit-stream is generated via the linear operations.

This holds for any d or s value in the LFSR generation scheme.

Variant 2: XOR($\mathcal{L}(d, s_1)$, $\mathcal{L}(d, s_2)$)

We generate two random-bit-streams in this variation by using two different seeds (s value), however, with the same d-bit LFSR. This also doesn't improve the quality of randomness backed by the intuitive understanding of LFSRs and highlighted in Table 3.4. The primary reason for this failure is that the XOR is a linear operation between two same d LFSRs. Finally, LRNG generated with primitive polynomials implies both BM and LFSR are seed-agnostic. Hence, the results are similar to what has been discussed in the previous variant.

d	NIST-tests	LCT
8	Failed	Failed
16	Failed	Failed
24	Failed	Failed
32	Failed	Failed
64	Failed	Failed
128	Passed	Failed

Table 3.4: Test result for LRNG XOR($\mathcal{L}(d, s_1)$, $\mathcal{L}(d, s_2)$).

Variant 3.1: XOR($\mathcal{L}(d_1, s_1)$, $\mathcal{L}(d_2, s_2)$)

For the third variant, we generalise the above XOR operation to consecutive d's (d_1 and d_2 being consecutive). The sequence passes NIST tests, except LCT for the combination of as low as $d_1 = 24$ and $d_2 = 25$, as shown in TABLE 3.5. However, the results for $d_1 = 128$ and $d_2 = 129$ are different. It is based on the fact that LCT couldn't identify it after the XOR operation.

d_1	d_2	NIST-tests	LCT
8	9	Failed	Failed
16	17	Failed	Failed
24	25	Passed	Failed
32	33	Passed	Failed
64	65	Passed	Failed
128	129	Passed	Passed

Table 3.5: Results for LRNG Variant 3.1: XOR($\mathcal{L}(d_1, s_1)$, $\mathcal{L}(d_2, s_2)$)

Variant 3.2: XOR($\mathcal{L}(d_1, s_1)$, $\mathcal{L}(d_2, s_2)$)

This variant differs from the previous variant on the combination of d-values used. In an attempt to find lower d values for hardware implementation, we looked at another variant of XOR($\mathcal{L}(d_1, s_1)$, $\mathcal{L}(d_2, s_2)$), with d_1 and d_2 being consecutive prime numbers. However, this didn't provide any significant advantage compared to Variant 1, with results in TABLE 3.5. Also, the results did improve compared to Variant 3.1. Specifically, for values of $d_1 = 7$ and $d_2 = 11$, the generated bit sequences pass all NIST tests except for the LCT. For XOR($\mathcal{L}(127, s_1)$, $\mathcal{L}(131, s_2)$) passed all the tests and, thus, can be used for suitable applications, as shown in TABLE 3.6.

d_1-bit	d_2-bit	NIST-tests	LCT
3	5	Failed	Failed
5	7	Failed	Failed
7	11	Passed	Failed
11	13	Passed	Failed
...	...	...	...
113	127	Passed	Failed
127	131	Passed	Passed

Table 3.6: Results for LRNG Variant 3.2: XOR($\mathcal{L}(d_1, s_1)$, $\mathcal{L}(d_2, s_2)$)

As evident from the preceding four variations of LRNGs, it is clear that XOR($\mathcal{L}(128, s_1)$, $\mathcal{L}(129, s_2)$) and XOR($\mathcal{L}(127, s_1)$, $\mathcal{L}(131, s_2)$) pass all the 15 tests (14 NIST-tests + LCT). To further investigate, we took 100 sets of random sequences for these two cases and found that XOR($\mathcal{L}(128, s_1)$, $\mathcal{L}(129, s_2)$) fails 1.53% of the individual tests, while XOR($\mathcal{L}(127, s_1)$, $\mathcal{L}(131, s_2)$) fails 1.40% of any individual tests. With all the consistency checks and good results, we implement both XOR($\mathcal{L}(128, s_1)$, $\mathcal{L}(129, s_2)$) and XOR($\mathcal{L}(127, s_1)$, $\mathcal{L}(131, s_2)$) on FPGA/hardware to be used in our QKD source.

3.6 FPGA implementation of PRNGs

Utilizing hardware like FPGA is necessary for QKD experiments, primarily because of the generation of voltage pulses to control/trigger the operation of any circuit/hardware. For this purpose, we prefer Arty 7 FPGA Evaluation Kit, Artix 7 35T FPGA [67] from AMD to implement the proposed LRNG method. The design of this board is specifically focused on providing a remarkably flexible Micro Blaze Soft Processing System. The board incorporates an on-chip analog-to-digital converter and provides ample I/O options for generating and acquiring voltage signals. Consequently, there is no need for an additional daughter board to facilitate interfacing with real-time signals. In addition to its specifications, this board boasts a compact form factor, affordability, and suitability for our operations. Additionally, the board is low cost ($159), having a 100 MHz clock rate.

We have chosen Very high-speed integrated circuit hardware description language (VHDL) codes for our implementation. One can use Verilog or another high-level language like C or Python to program an FPGA. The generated random voltage pulses are time-tagged, and subsequently, bits are tested against NIST-STS. The resulting outcomes are presented in the Table 3.7. The VHDL code implemented on FPGA can be obtained from the Ref. [68].

3.6.1 LFSR based PRNGs

The clock rate for random voltage generation is set to 1 MHz for hardware implementation. To test the random pulses generated from the FPGA board for XOR($\mathcal{L}(128, s_1)$, $\mathcal{L}(129, s_2)$) and XOR($\mathcal{L}(127, s_1)$, $\mathcal{L}(131, s_2)$), we connect both the ports (clock and random signals) to the IDQuantique ID900 time-tagger. It records the arrival time of these pulses and records it as time stamps. The recorded time stamps by the time tagger are processed in Matlab to obtain a binary bit-stream. The data consists of 55 M random bits, where each set is recorded for 55 s. The TABLE 3.7 showcases the results from our two proposed XOR methods. The results obtained are in good agreement with the numerically obtained data. In all of these cases, all 15 NIST tests were successfully passed.

Modified LFSR	NIST-tests	LCT
XOR($\mathcal{L}(128, s_1)$, $\mathcal{L}(129, s_2)$)	Passed	Passed
XOR($\mathcal{L}(127, s_1)$, $\mathcal{L}(131, s_2)$)	Passed	Passed

Table 3.7: Test results for bits obtained from Arty 7

Throughout the hardware implementation, the selection of an optimal clock rate proved to be a crucial parameter due to the limitations in synthesizing certain code configurations on the FPGA board. We systematically generated random numbers for a 1 MHz clock rate using FPGA. The device consumes 64 mW of power for XOR ($\mathcal{L}(128, s_1)$, $\mathcal{L}(129, s_2)$) and 63 mW for XOR($\mathcal{L}(127, s_1)$, $\mathcal{L}(131, s_2)$).

3.6.2 LFSRs in QKD

The VHDL code for generating random numbers via shift register is fed to the Arty 7 FPGA board. The setup for generating time stamps and corresponding voltage pulses is shown in Figure 3.6. Along with random voltage pulses, we also generate clock pulses for referencing to assign random pulses as bits 0 and 1. The FPGA board's output is derived from its I/O ports, featuring Pmod connectors in this setup.

Typically, the clock and random voltage signal produced by the FPGA board must be fed into the time tagger module to generate random bits. However, since the time tagger module is equipped with an SMA connector, using a Pmod to SMA converter shown in Figure 3.6 becomes imperative to establish the necessary connection. Furthermore, in the QKD setup, the laser diode driver circuit also incorporates an SMA connector. Consequently, the utilization of a Pmod to SMA converter is also crucial to seamlessly integrate the FPGA board's signals with the laser diode driver circuit.

The timing waveform in Figure 3.6 illustrates the waveform of the random voltage pulses and reference clock pulses. We utilize a time tagger (IDQuantique ID900) to track the timestamps of the clock and random pulses accurately.

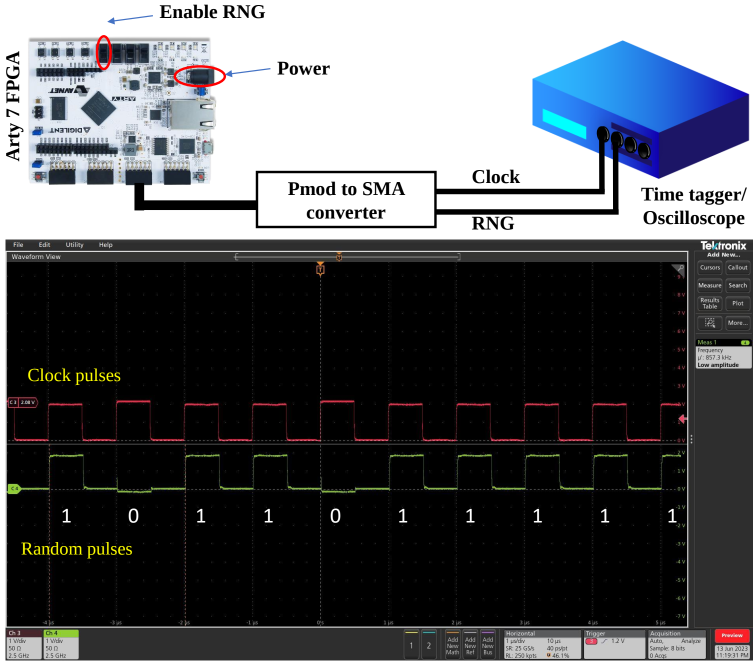


Figure 3.6: The timing waveform of random voltage pulses and clock pulses was generated from the FPGA board and obtained using an oscilloscope. When the RNG and clock pulses are active simultaneously, it indicates a bit value '1'. Conversely, when only the clock pulse is present without any accompanying random pulse, it represents a bit value '0'.

The primary application for LFSR in QKD applications comes into the picture for P&M protocols. Our application is mainly dedicated to the BB84 protocol implementation, which is one of the well-studied P&M and popular QKD protocols. The operational functioning behind P&M-based QKD protocols is to use a particular property of photons (say, polarisation/phase) to generate quantum states in mutually unbiased bases, send them to another party who measures them, and form a key. Say Alice chooses a polarization degree of freedom to encode and share the key with Bob for communication. Alice needs to prepare her states which are indistinguishable (arbitrary) in four different polarisation quantum states, namely horizontal $|H\rangle$, vertical $|V\rangle$, diagonal $|D\rangle$, anti-diagonal $|A\rangle$. To prepare this signal, Alice uses four different laser diodes (one for each quantum state) and combines them. However, while combining, we need to ensure that laser circuits for all four laser diodes fire randomly so that the signal is unbiased to all four quantum states. This requirement is a pre-requisite for the unconditional security of any P&M-based QKD protocol.

To meet the specified requirements, we have developed a laser driver circuit that utilizes random voltage pulses generated from an FPGA. Additionally, we have incorporated a 1×4 demultiplexer in the FPGA, which is controlled by randomly generated select lines derived from LFSR-generated random bits. This demultiplexer ensures that only one output port of the FPGA is enabled at any given time, thereby allowing for activating a single laser diode at a time. By implementing this demultiplexing scheme, we can generate four random polarization states in Figure 3.7. Figure 3.7 (top) presents the schematic diagram illustrating the demultiplexing scheme. The random voltage pulses generated from Arty 7 for driving the laser diode driver circuit are shown in Figure 3.7 (bottom). The optical setup of the QKD transmitter is shown in Figure 3.8. Here, we have used a laser diode driver circuit (marked in red circle), integrated with other optical components of the BB84 setup [69]. Information bits are encoded in polarization and sent to Bob via the quantum channel.

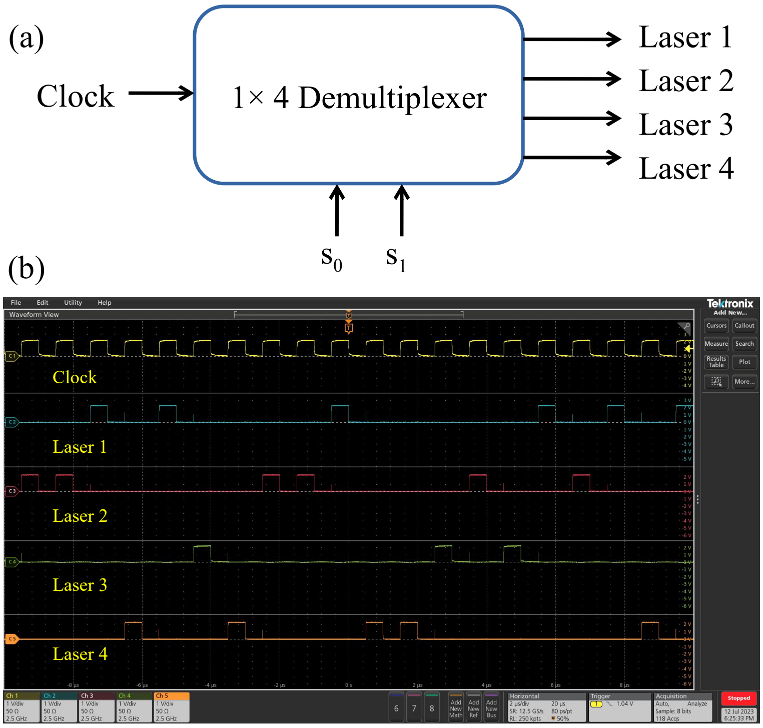


Figure 3.7: a) Schematic of generating four TTL pulses using 1×4 demultiplexer using the random bits generated from LFSR as a select line s_0 and s_1. The select line enables one laser at a time and performs the demultiplexing. b) Four colours representing four random TTL pulses generated from FPGA to drive four corresponding laser diodes, each diode corresponds to polarisation states ($|H\rangle$, $|V\rangle$, $|D\rangle$, $|A\rangle$).

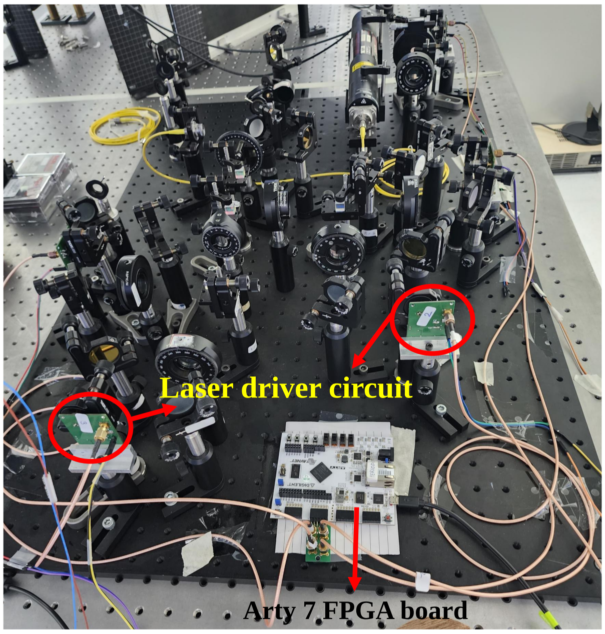


Figure 3.8: The optical setup for BB84 QKD setup which includes a Arty 7 FPGA board to generate RNG, which drives the laser diode driver circuit for QKD experiments.

3.7 Summary

In this chapter, we have discussed the generation of PRNGs, CS-PRNGs, and based on the type of bit-streams, the post-processing methods. Specifically, we saw one concrete example of the post-processing method using XOR operations that influence the NIST test results. We saw its specific implementation on an FPGA board which was pursued in collaboration with hardware experts.

"Quantum phenomena do not occur in Hilbert space. They occur in a laboratory"

Peres, Asher

4 Investigating Device-Independent QRNG

4.1 Introduction

The requirement of random numbers in cryptography is typically fulfilled by the use of algorithmically generated random bit-streams, also referred to as pseudo random number generators (PRNGs). PRNGs have many applications ranging from digital signatures to IoT applications [70]. On the other front, military grade applications require highly secure random number generators. The quality of randomness in PRNGs relies on their computational complexity and their length [71]. However, they can turn obsolete in cryptographic applications pertaining to advanced algorithms which can dig up the low computational resolution making the apparently random bit-streams non-random. Quantum random number generators (QRNGs) provide an alternative based on their quantum unpredictability (arising from the laws of quantum mechanics) [18]. Various discrete [72] and continuous [73] degrees of freedom are useful in QRNGs for their cryptographic applications.

On the testing front, as earlier discussed in Chapter 3, NIST Statistical Test Suite [74], Dieharder [75], ENT [76] have been used to prove randomness of QRNGs. But this is a half-baked truth. NIST Statistical Test Suite [74], as the name suggests, is used for testing statistical properties like uniformity amongst PRNGs and thus, can strictly be used to check for statistical properties of QRNGs. Nonetheless, for the quantum case, one has to use quantum correlation validators like quantum entanglement measures [77]. For example, given n bits, these quantum correlations forbid one to predict (n+1)th bit. There is no common consensus on using a single quantum certification for all quantum random number generators and hence, different works in the literature use different techniques for certification [78]. By certification, we understand that a given inequality can certify that the bits generated have a quantum correlation. This work uses quantum measures like CHSH inequality violation [39] for the same.

On the device-independence front, treating the process of generation and measurement as a black box requires a lot of independent checks and, thus, resources. Our technique, being device-independent, assumes no constraints imposed by the devices. Such a technique can easily be integrated into faster device-independent protocols [79, 80] by using an efficient non-linear crystal.

The chapter studies quantum correlations in QRNGs to exploit the resource of quantum correlations for device-independence of QRNGs. Initially, we discuss the theoretical framework required to understand the physical process of generation and certification of random number generation. This is followed by the experimental setup on how to generate random numbers from bipartite quantum entanglement. In the results section, the random bits, generated quantum mechanically in a device-independent manner, are tested for their performance against NIST Statistical test suites.

4.2 Theoretical Aspects

Before discussing the experimental setup, we need to understand the theoretical framework which circumscribes the quantum random number generation and testing process. Firstly, we describe the generation of the QRNG process which is a typical entanglement-based QRNG. Secondly, to test that our technique is device-independent, we take the aid of quantum correlations between the two qubits. Device-independence, as the name suggests, implies no faith to be put in the devices. To develop faith on the source front, a HOM-type setup suffices, while to keep a check on measurement devices, two independent measures based on Bell parameter violation are used. In simple words, the relative phase information between two photons ensures device-independence. This relative phase information is manifested in HOM to keep a check on devices on the source, while quantum entanglement keeps a check on measurement devices. Thirdly, we discuss the typical method (NIST Statistical Test Suite) of checking the uniformity of QRNGs.

4.2.1 Generating random numbers from quantum entanglement

The use of quantum entanglement for the generation of quantum random numbers requires the ability to produce high-quality entangled states. For this, we first use a HOM interference experimental setup (needed to prove source-independence). To achieve the pair of entangled photons, once at the dip point of the HOM interference curve, we make the photons distinguishable in one degree of freedom, say polarization analogous to the quantum eraser experiment as described in [28]. This is done by rotating one of the half waveplates by 45°. Theoretically, the output state of the HOM interferometer (represented by point A in Figure 4.2) is given by

$$|\psi_{AB}\rangle = (|2H\rangle_a |0\rangle_b + |0\rangle_a |2H\rangle_b)/\sqrt{2}$$

Typically for a general θ rotation of the half-wave plate (HWP) with respect to $|H\rangle$ in $H-V$ basis, the state can be written as

$$\begin{aligned} |\psi(2\theta)\rangle &= \frac{\iota}{\sqrt{2}} cos\theta(|2H\rangle_a |0\rangle_b + |0\rangle_a |2H\rangle_b) \\ &\quad + \frac{1}{2} sin\theta(|H\rangle_a |V\rangle_b - |V\rangle_a |H\rangle_b) \\ &\quad + \frac{\iota}{2} sin\theta(|H,V\rangle_a |0\rangle_b - |0\rangle_a |H,V\rangle_b) \end{aligned} \tag{4.1}$$

If the polarisation of one of the photons is rotated by $2\theta = 90°$, the state transforms to

$$\begin{aligned} |\psi(90)\rangle &= \frac{1}{2}(|H\rangle_a |V\rangle_b - |V\rangle_a |H\rangle_b) \\ &\quad + \frac{\iota}{2}(|HV\rangle_a |0\rangle_b + |0\rangle_a |HV\rangle_b) \end{aligned} \tag{4.2}$$

Because our detectors are click detectors, the post-selected state reduces to

$$|\psi_{AB}\rangle = \frac{1}{\sqrt{2}}(|H\rangle_a|V\rangle_b - |V\rangle_a|H\rangle_b) \tag{4.3}$$

Furthermore, we generate random numbers through a bipartite entangled state. This is achieved by tracing out one of the qubits of a bipartite quantum entangled pair; the other qubit has an equal probability of being either H or V polarized (represented by values p_0 and p_1). Mathematically, the probabilities after projective measurements i.e. $M_0 = |H\rangle\langle H|$ and $M_1 = |V\rangle\langle V|$ [81] can be calculated as

$$p_0 = \mathrm{Tr}\left((M_0 \otimes I)(\rho_{AB})\right) = \frac{1}{2}$$

and

$$p_1 = \mathrm{Tr}\left((M_1 \otimes I)(\rho_{AB})\right) = \frac{1}{2}$$

.

Here, ρ_{AB} is the bipartite density matrix of the entangled state, and “0” and “1” refer to binary bits, decided by the experimentalist. This equi-probable distribution is the basis of generating random numbers from a bipartite entangled state. We employ

HOM curve as a certification technique where its visibility serves as a measure of source-independence. Once you know that the visibility is good enough, we lose the assumption of putting our faith in the devices on the source front.

4.2.2 NIST Statistical Test Suite

The NIST Statistical Test Suite is a hypothesis-based suite that checks whether a given sequence is random. The test-static p gives the confidence level in the null hypothesis, stating that a said sequence is random. The test suite entails 15 tests with numerous sub-tests. These tests check for patterns in the bit-stream in long as well as short range. The tests include auto-correlation, compression, and spectral frequency based tests as discussed in Chapter 3. The cut-off value for accounting for false positives in this hypothesis is set by choosing a p-value, and it can vary in the range (0.1, 0.001) as referred to in the manual [74]. For our case, we use the default value set at 0.01. Before testing against the NIST test suite, the data is post-processed using the Toeplitz hash function. The advantage of using the Toeplitz hash function is that it scrambles the "quantum information" evenly amongst all bits. This is ensured by the 2-universality of the Toeplitz hash function [82].

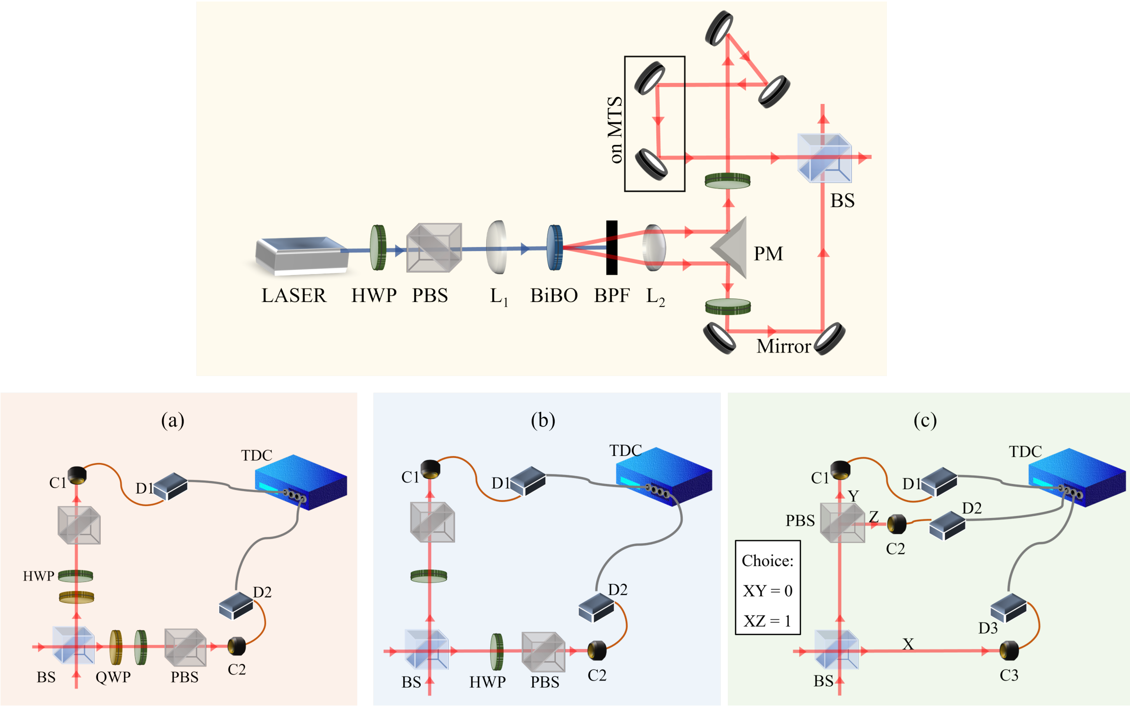


Figure 4.1: Experimental Setup for entanglement based random numbers: (a) certification via density matrix (b) certification via direct CHSH Bell parameter (c) generation of random numbers. L_1, L_2: lens, PM: prism mirror, MTS: motorized translation stage, BS: beam splitter, PBS: polarising beam splitter, QWP: quarter wave plate, HWP: half wave plate, BPF: bandpass filter, $C1, C2, C3$: Coupler, D1, D2, D3: detectors, TDC: time to digital converter, "X", "Y" and "Z" denote path of the photons to different detectors

4.2.3 Theoretical framework to study Quantum Entanglement

Several authors [83] have provided methods to quantify "entanglement" as a resource with the aid of different measures [84, 85, 77, 86]. One of the well-studied experimental entanglement measures is CHSH inequality [87]. The CHSH inequality measures correlations between two qubits using different measurement bases. For an entangled pair A and B, we will label the bases for the qubit A to be a, a', and for qubit B to be b, b' respectively. This allows us to calculate the CHSH parameter S,

$$S = E(a,b) - E\left(a,b'\right) + E\left(a',b\right) + E\left(a',b'\right), \tag{4.4}$$

where, E(a,b), represents the expectation value of the product of measurements in a-b basis. The same holds true for other combinations of a'-b' basis. For CHSH parameter $S \geq 2$, the measured state shows quantum entanglement.

Alternative to this measure, quantum state tomography (QST) estimates the density matrix, ρ describing a given state from projective measurements. Pertaining to statistical noise in such experiments, Bayesian and Maximum likelihood methods are used in finding the estimate ρ_{est} closest to the true density matrix ρ from the space of physical density matrices $\mathbf{P}$ [88]. The premise to choose CHSH inequality is that it is well studied, while the premise for density matrix choice is that it contains complete information about a state.

Consider a 2-qubit state characterized by its density matrix ρ, which can be written as follows:

$$\rho = \frac{1}{2^2} \sum_{i_1,i_2=0}^{3} u_{i_1,i_2} \hat{\sigma}_{i_1} \otimes \hat{\sigma}_{i_2}, \tag{4.5}$$

where u_{i_1,i_2} are real numbers and $\hat{\sigma}_i$ are the Pauli matrices. The state ρ can be obtained from 4^2 (for a 2-qubit system and 4 refers to Stokes parameters [89] type measurements) optimal measurements [90]. To assign a density matrix to a physical state, (i) ρ must be normalized (Tr$\rho = 1$), (ii) Hermitian ($\rho^\dagger = \rho$), and (iii) positive semi-definite ($\langle\psi|\rho|\psi\rangle \geq 0$) for all unit-norm states $|\psi\rangle$.

On the initial investigation of the outcomes of projection of state ρ with a sufficient number of repetitive measurements, we employ least-squares (LS) inversion to estimate ρ_{LS} of state ρ; from the projection probabilities approximated from measured outcome frequencies. This method is limited by the contribution of statistical noise and may return the state ρ_{LS} to be nonphysical ($\rho_{LS} \notin \mathbf{P}$), as it cannot guarantee positive semi-definiteness.

Nevertheless, one can implement the approach of estimating ρ_{est} from set $\mathbf{S}$ using a point estimator such as Maximum likelihood estimator (MLE) [90] or an uncertainty region estimator like Bayesian QST [91]. The need to implement two different approaches of estimation is twofold. One, it keeps a check on statistical noise i.e. in a device-independent setting. Secondly, it draws a comparison between two different approaches of estimation.

Bayesian quantum state estimation

Consider that any state (ρ') in $\mathbf{S}$ is parameterized by a vector x, ensuring that any value within x's support yields a physical matrix. According to Bayes' theorem, the posterior probability distribution of x, given experiment results D [91], follows:

$$\pi(\mathrm{x}) = \frac{1}{\mathrm{Z}}\mathrm{L_D}(\mathrm{x})\pi_0(\mathrm{x}), \tag{4.6}$$

where $L_D(\mathrm{x})$ represents the probability of obtaining the observed outcomes given state ρ_{LS}, $\pi_0(\mathrm{x})$ denotes the prior distribution (reflecting beliefs about ρ before the experiment), and Z is a normalizing constant ensuring $\int d\mathrm{x}\, \pi(\mathrm{x}) = 1$.

Access to $\pi(\mathrm{x})$ enables computation of the expectation value of any function ϕ of ρ':

$$\langle \phi(\rho') \rangle = \int d\mathrm{x}\, \pi(\mathrm{x})\, \phi(\rho'(\mathrm{x})), \tag{4.7}$$

facilitating determination of the mean and standard deviation of any quantity of interest.

However, the numerical computation of the integral resembling in Eq. (4.7) proves to be quite challenging. Hence, we have used the method of obtaining R samples $\{x^{(1)}, x^{(2)}, \ldots, x^{(R)}\}$,to approximate the Eq. (4.7) to:

$$\langle \phi(\rho') \rangle \approx \frac{1}{R}\sum_{r=1}^{R} \phi(\rho'(x^{(r)})), \tag{4.8}$$

as described in [91]. Lukens parameterizes the weights from Gamma distributed random variables for the prior for $\pi_0(\mathrm{x})$ and uses Monte-Carlo sampling methods for sampling of R which together influence the outcome $\langle \phi(\rho') \rangle$. As sampling (performed via the help of random numbers) is an integral component in the estimation of the density matrix, the density matrix calculations form a feedback loop with the process of random number generation. Thus, to avoid detection side attacks in a device-independent setting and also draw a comparison, another estimator like maximum likelihood is required.

Maximum likelihood quantum state estimation

Maximum likelihood estimation (MLE) identifies the density matrix most likely to have produced the observed data D:

$$\rho_{MLE} = \mathrm{argmax}_{\mathbf{P}}\ \ (L_D(\rho')),\ \text{where } \rho' \in \mathbf{P} \tag{4.9}$$

where $L_D \propto P(D|\rho')$ represents the probability of obtaining the observed outcomes D given state ρ', as defined by a specific model [90]. Proper parameterization of ρ' ensures that the estimate (ρ_{MLE}) is a physical state. This method has become the prevailing approach to Quantum State Tomography (QST) due to its easiness. However, ρ_{MLE} is a point estimate, lacking quantification of result uncertainty.

Once the density matrix is calculated, we evaluate CHSH Bell parameter S, as an upper bound, given by the formula,

$$S_{max} = 2\sqrt{c_{11}^2 + c_{22}^2} \tag{4.10}$$

where c_{11} and c_{22} represent the largest eigenvalues of C^TC and C^T is the transpose of C. The correlation matrix C with elements c_{ij} is related to the density matrix ρ for a 2-qubit system by the relation

$$\rho = \frac{1}{4}\sum_{i,j=0}^{3} c_{ij}\hat{\sigma}_i^1 \otimes \hat{\sigma}_j^2 \tag{4.11}$$

as mentioned in [92].

4.3 Experimental Setup

The laser (TOPTICA-TopMode 405) with a center wavelength of 405 nm and a bandwidth of 0.01 nm emits a Gaussian beam. The beam passes through a half-wave plate (HWP) followed by a polarizing beam splitter (PBS). This combination gives control over the intensity of the beam. A 50 cm lens (L_1) is used to focus the beam on a Type-I Bismuth Borate (BiBO) crystal of length 5 mm. The pump power before the nonlinear crystal is 5 mW. Two correlated degenerate spontaneous parametric down-converted photons are generated in the crystal in a non-collinear geometry assisted via angle tuning. The photon pairs are collimated by a 10 cm lens (L_2) and separated by a prism mirror (PM) to interfere in a Mach-Zehnder-like setup as shown in Figure 4.1. In one of the paths, a motorized translation stage MTS25-Z8, with a resolution of 29 nm, is added to compensate for the extra delay, if any. The two output ports of the beam splitter are initially used to show a HOM dip [93] as shown in Figure 4.2.

Once the two polarisation entangled qubits are generated as described in the theoretical section, one can simply detect one of the qubits in the detector following path "X". The other qubit is measured in H-V polarisation basis (path "Y" and "Z"). Measuring this other qubit in coincidence with the photon in path "X" generates random numbers as it gets detected either in the H polarisation (bit 0) or V polarisation (bit 1) at a particular instant with a 50:50 probability as shown in Figure 4.1(c). The detectors (SPCM-800-14FC with a dark count of 100 counts per second) are identical in all three arms. To see measurement-device-independence on the detector side, the diagram of the setup is slightly modified after the beam splitter to take projective measurements using quarter wave plate (QWP), HWP and a PBS combination as shown in Figure 4.1 (a). On the contrary, to show the source-independence, HOM curve is used as a certification technique. Its visibility serves as a parameter for source-independence. For example, with good dip visibility, your need to trust that the laser is behaving well evaporates. Thus, a high visibility of the HOM curve is desired. In our case, we achieved a visibility of 97%. Since no phase information is required to generate bit-streams, random bit-streams generated from maximally mixed state of a bipartite system and single photon source are identical. The extra phase information in the entangled case helps in providing security against attacks on the source. After calibration of data points A and B (as highlighted in Figure 4.2) using the HOM curve, we generate entangled states at these data points.

NIST Statistical Test Suite	***p*-value(Dataset A)**	***p*-value(Dataset B)**
Approximate Entropy	0.985	0.546
Block Frequency	0.380	0.129
Cumulative Sums	0.973	0.557
FFT	0.979	0.973
Frequency	0.840	0.465
Linear Complexity	0.840	0.965
Longest Runs	0.060	0.966
Non Overlapping Template Matching	0.069	0.325
Overlapping Template Matching	0.721	0.590
Random Excursions	0.843	0.383
Random Excursions Variant	0.435	0.621
Rank	0.993	0.084
Run's	0.858	0.325
Serial	0.403	0.356
Universal	0.285	0.210

Table 4.1: p-value for different test suites: Dataset A corresponds to the entangled state generated at Setting A, and Dataset B corresponds to the entangled state generated at Setting B, respectively, as highlighted in Figure 4.2. Both post-processed datasets are 1.2 M long.

4.4 Results

In this study, we have generated two datasets of 4.5 million (M) bit-streams from the experimental setup shown in Figure 4.1. The data is recorded for two different cases: Dataset A is recorded at the point of highest visibility of the HOM curve of Figure 4.2, which represents a maximally entangled state, and Dataset B is recorded 700 nm away from the dip point, which degrades the quality of the above entangled quantum state. After post-processing using the Toeplitz hash function, the length of random numbers is reduced from 4.5 M to 1.2 M.

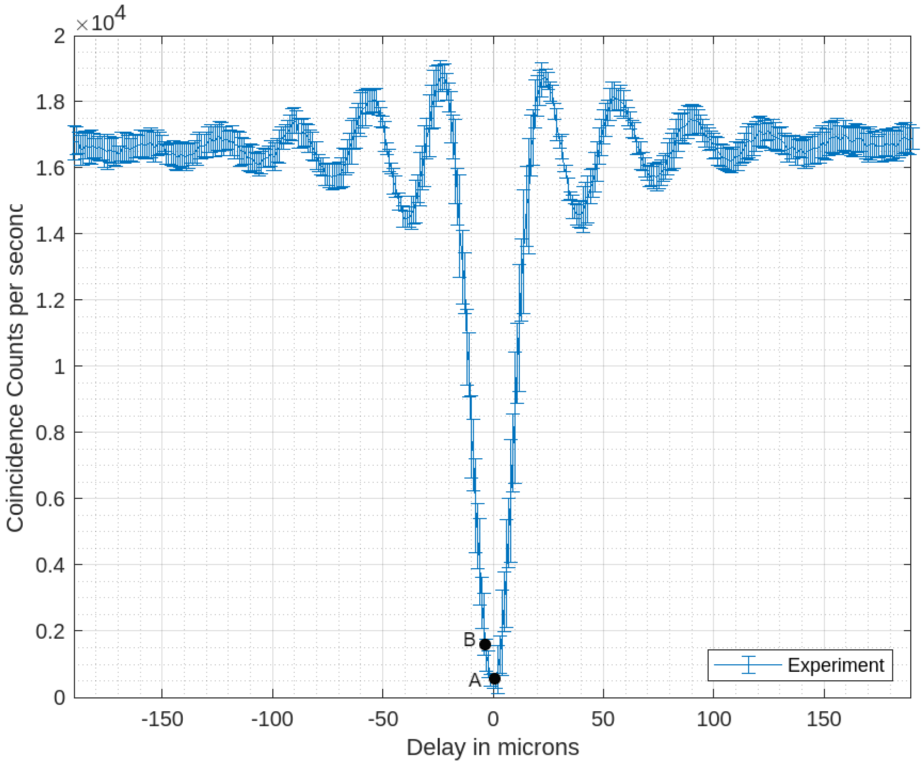


Figure 4.2: Hong Ou Mandel (HOM) curve serves as a source-independence certification for a polarisation entangled state at points A and B

The datasets are tested against NIST-STS for quality of statistical randomness, and the results are highlighted in Table 4.1. Every test in Table 4.1 shows a test-static $p \geq 0.01$, suggesting that the generated dataset is statistically random. For example, the frequency shows the faith in our null hypothesis (sequence is random) with a confidence or p-value of 0.84. A similar statement could be made for other tests. Although the two datasets show a different quantum signature, the same cannot be said about their randomness property with statistical measures. This could be because of multiple reasons. For one, the post-processing of QRNGs is computational and dominates its influence over the quantum nature of the dataset. Secondly, the measure of randomness for the case of QRNGs is, typically, described as the measure of randomness (computational) appended to the quantum verifier implying that randomness and quantumness are two independent parameters.

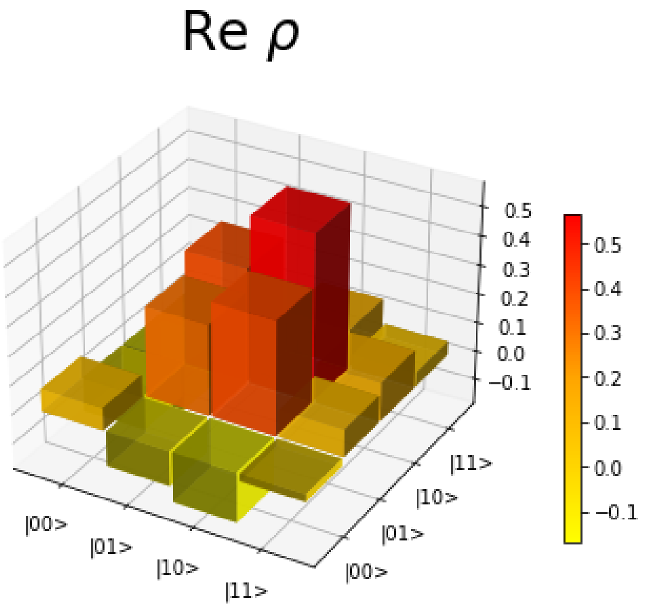


Figure 4.3: Real part of the density matrix: Projection of an entangled Bell state (Eq. 4.3) at point A in Figure 4.2 in H-V basis. The magnitude of each projection is color-coded.

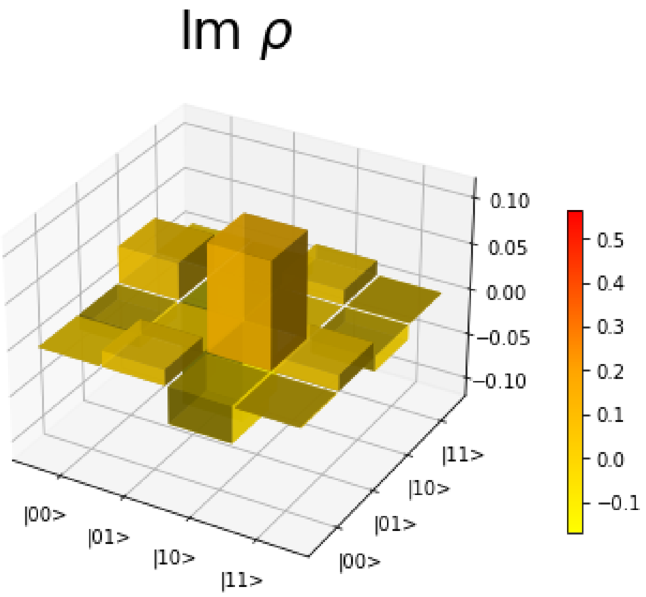


Figure 4.4: Imaginary part of the density matrix: Projection of an entangled Bell state (Eq. 4.3) at point A in Figure 4.2 in H-V basis. The magnitude of each projection is color-coded.

Sr. No.	S	S(ρ_{MLE})	S($\rho_{Bayesian}$)
Dataset A	2.78 ± 0.03	2.65	2.81 ± 0.02
Dataset B	2.51 ± 0.02	2.40	2.47 ± 0.01

Table 4.2: CHSH Bell parameter S value obtained via different methods is shown. Column 2 shows direct S measurement; Column 3 denotes S obtained from ρ post-processed with MLE; Column 4 denotes S obtained from ρ post-processed with Bayesian estimation. $|S| \geq 2$ indicates quantum behaviour of the device.

The dataset A and B are obtained from two different quantum states, therefore having different "quantum information". Here, we have used the CHSH inequality (S) as one such quantifier of entanglement, as given in Eq. (4.4). The CHSH violation is measured from the direct observation as discussed in Ref. [39] and alternatively by estimating the density matrix using Bayesian and MLE quantum state tomographic techniques as in Ref.[90, 91].

For the estimation of S, the direct experimental measurements are taken using the method outlined in Ref. [39], and results are shown in column 2 of Table 4.2. The S for the settings in dataset A and dataset B is well above the classical limit ($|S_{classical}| = 2$), indicating the quantum nature of random bits in both datasets. However, to eliminate the possibility of assumptions based on device imperfections (say, detector efficiency), we calculate **P** using Eq. (4.10). The estimated density matrices (ρ^A_{est}, ρ^B_{est}) are the density matrices of the entangled photon states used to generate the datasets A and B, respectively.

To estimate the density matrix for each dataset using MLE, experimental measurements were performed as in Ref. [90]. Column 3 of Table 4.2 provides the estimated value of S(ρ_{MLE}) using Eq. (4.10). For the settings in dataset A , we estimate $S(\rho^A_{MLE}) = 2.65$ and for dataset B, $S(\rho^B_{MLE}) = 2.40$. The estimated real and imaginary parts of ρ^A_{MLE} are shown in Figs. 4.3, 4.4.

The results in Table 4.1 indicate a high confidence that both the datasets have statistically independent random bit-streams. Additionally, the results shown in Table 4.2 prove the quantum correlation. The table, although indicating quantum signature, shows some inconsistency amongst different values. This discrepancy arises due to two reasons. One is that Bayesian estimation is better than MLE in cases where the right prior is known (Dataset A in our case). Secondly, the effect of sampling can bias the Bayesian estimation (Dataset B in our case). For Dataset B, right prior is not known. This requires further investigation. However, the result provided in this study positively concludes that the QRNG developed using polarization-entangled photon pairs is statistically random and secure against source and detector side attacks as indicated by HOM dip visibility and CHSH value respectively. A contrasting difference between our technique and randomness expansion protocols is that we have performed sequential measurements bunched together for a quantum correlation quantifier, while the semi-device-independent protocols sparse them in generation and test rounds with some bias picked from a quantum source.

One open problem with random numbers is to investigate whether unpredictability and statistical properties are inter-related. To address this query, one can use a metric from information theory, specifically min-entropy [94], defined as:

$$H_{\infty}(X) = -log_2(p_{max})$$

where p_{max} is the probability of maximum occurrence of random variable X to quantify the said relationship. It is observed that min-entropy for Dataset A is $H_{\infty}(X) = 0.999735$ and for Dataset B is $H_{\infty}(X) = 0.999038$. This decrease shows a relation between unpredictability and statistical properties. As there is a decrease in statistical correlations (refer to p-values in Datasets A and B of the frequency test), there is a similar decrease in unpredictability. We performed scrambling operations at different lengths for the above datasets and found no significant change over length. One possible explanation is that the scrambling process is classical, and thus, it is challenging to see quantum unpredictability signature in highly classically processed data. This is supported by the fact that the p-values of the Frequency test and Linear Complexity test are equal for Dataset A. These observations require further investigation of the parameters of the algorithms used in the test suite to understand the difference between computational and quantum unpredictability which could be a possible direction of future study.

4.5 Summary

In this article, we have provided a sanity check of the quantum source (HOM dip) and investigated the working of a fully device-independent quantum random number generator. One can put a constraint on resources and translate the fully device-independent scheme to measurement device-independent or source-independent quantum random number generator protocols with automation of HWPs and calculation of smooth conditional min-entropy (based on the density matrix). Other verifiers like entanglement measure/witness can be used to define new protocols in the semi-device-independent regime or for higher dimensions. Also, it is theoretically proven that the visibility of the HOM curve is equal to the purity of input photons [95]. This provides a correlation between closeness to the dip point and the amount of CHSH Bell violation as indicated by Table 4.2. Physically, it means that relative phase information between two photons is used to prove device-independence and thus, preventing attacks on QRNGs from Eve. Here, device-independence checks can be relaxed to cases where HOM serves as a check for source-independence or CHSH Bell parameter serves as a security check for measurement device-independence. Sending the qubit which is being traced out here, to another party, post-processing to consider loss over the channel and together with the addition of local operations using classical communication (LOCC) can convert this random number generation scheme to a device-independent quantum key distribution scheme.

"Artificial intelligence is not a substitute for human intelligence; it is a tool to amplify human creativity and ingenuity."

Li, Fei-Fei

5

Strengthening the no-go theorem for QRNGs

5.1 Introduction

Let's consider a toy example to develop an understanding of how statistical testing works, and what are their constraints. Consider that we are in Las Vegas playing the roulette table. To simplify the explanation, let's imagine that only red and black outcomes are possible. In this case, the probability of a red or a black outcome are both 1/2, and the probability that a specific combination of m outcomes appears is $1/2^m$. Let's further assume that 19 reds have appeared in a row in last 19 rounds respectively. Can we predict the next bet? Many among the hoi-polloi would bet in favour of the belief that "the probability that the next is red is very low because there have been already 19 reds consecutively", and thus, place the next bet on blacks. The wheel spins again, but, unfortunately, the outcome turns out to be red again. Furious, and with a feeling of having been tricked, we decide to go to the director of the Casino to make a complaint. The director, however, who is a quantum physicist, kindly tells us that it is simply impossible that we have been tricked because the outcome of the roulette is based on sampling a quantum process. We might have a hard time believing it, but he continues, "Well, the roulette table has been here for 2 years now, and it generates a new outcome every half second. This means that more than 130 million values have been generated already. It's true that the sequence that has just occurred is unlikely and not typical (the probability of this happening is $\frac{1}{2^{20}} \approx 0.95 \times 10^{-6}$). However, too much data has been generated during these two years, the 20-red-long sequence should have happened approximately 120 times already". To prove himself, he goes to the records, processes the data, and detects that such a sequence has actually occurred 117 times. Thus, probabilistically, it was a perfectly random sequence. In general, a statistical test "looks" at sequences, and if they are outside of what we might call, loosely speaking, typical, it concludes that the sequence is not random. This brings some limitations of course, since perfect random number generators, like the one in the example above, also generate "atypical" sequences. The most relevant implication of this fact is that perfect random number generators are expected to fail statistical tests with non-zero probability when, by chance, they produce "non-typical" sequences. Nowadays, statistical testing of random number generators is typically carried out via batteries of statistical tests. The most commonly used batteries of statistical tests today are the NIST- Statistical Test Suite [62], the Test U01 by [96], and Dieharder suite [97]. Statistical testing is a useful tool when it comes to the detection of patterns. Basically, it can be used to prove lack of randomness, however not as a tool to prove the existence of randomness. In the next section, we discuss the second approach to testing random number generators, which involves more advanced techniques like machine learning models or measures of algorithmic randomness. Obviously, this approach has it's pros and cons. For example, this technique is universal and independent of type of method used in generation process. However, it still tries to calibrate random numbers based on the output bit-stream rather than the generation process. Other alternatives, like modelling a QRNG source and calculating min-entropy, typically are used to calibrate the process rather than output. Both the techniques lie at different points on the optimisation axis between universality and robustness of the QRNG model.

The predictable behaviour of PRNGs is vulnerable to advanced algorithmic attacks [98] and deciphering using machine learning algorithms. Machine Learning (ML) is a field of studying statistical models to be used by computer systems to perform tasks without explicit instructions. These models are successfully employed in diverse fields like pattern recognition [99] and recommendation algorithms [100] in everyday life. They serve both as a consumer and generator of random numbers.

Previously, recurrent neural networks (RNNs) have been used to predict the bit-stream of PRNG algorithms [101]. Here, we use a well-studied time series model, namely long-short term memory (LSTM) model, to predict the next bit from the previously known bit-streams. We compare its performance against NIST statistical test suites (NIST-STS). On the generation front, due to the predictability of PRNGs, quantum random number generators (QRNGs) provide an alternative in the field of cryptography as a resource for "better" randomness.

"Better" randomness of QRNGs stands on the surface that natural laws of physics forbid anyone to discern the next random bit generated. Quantum randomness is global randomness as it is based on a more broader definition based on assumptions of quantum mechanics and addresses processes rather than algorithmic information theory, which relies on the notion of individual random sequences using a self-delimiting universal computer. As per Renner's definition [102], the quantum advantage is shown by appending a quantum quantifier of trace-distance. Proving the source of generation has quantum origin implies that QRNGs being better than PRNGs is a statement that needs some experimental verification. Although a clear advantage can be seen for quantum correlations where the correlations being quantum add privacy to these bits. However, admittedly, QRNG raw bit-streams are not good on statistical properties (built on the definition of uniformity than unpredictability) and are often post-processed using hash functions [103]. Once the process of generation is complete and the bit-stream is ready for use in real-world applications, is there a technique to distinguish the source of origin (Quantum or Pseudo) based on the given bit-stream? The question is of paramount importance in providing access to randomness beacon as a service. How does the user verify the bit-stream he received has quantum properties? Some authors have shown that there is a direct anti-correlation between quantumness and randomness. In this article, we try to distinguish them on machine learning grounds. Without question, QRNGs provide an additional calibration on the process rather than the output bit-stream. But does this nature get reflected in the bit-stream generated? If true, randomness resource as a beacon service is a success and we have solved the age-old encryption problem with QRNGs. Otherwise, where does the user see the advantage of QRNGs? Is it just in using quantum correlations to prove device-independence? With this motivation in mind, we discuss the possible differentiating factors with four different computational tools.

In other words, from a user perspective that given a random bit-stream, it is impossible to decide whether the generation process is classical or quantum if the measures involved are Turing-computable with practical feasibility as a constraint. Researchers have shown that QRNGs are in-computable [104].

In this data driven approach, we draw a comparison between pseudo random number generators and quantum random number generators and provide experimental evidence of the no-go theorem on the machine learning front and two measures of algorithmic randomness. The choice of PRNG and QRNG chosen are ones based on current security standards. To be specific, ChaCha20, a computationally secure PRNG (CS-PRNG), and quantum entanglement based QRNG are chosen to differentiate between PRNGs and QRNGs. The testing fronts are similar to those proposed by the authors recently and independently claiming in-distinguishability [105]. In addition to a similar methodology followed from their work for in-house generated QRNG and simulated CS-PRNG, we have appended the analysis on the machine learning front. In the next section, we discuss the machine learning model and other measures of randomness on the algorithmic front.

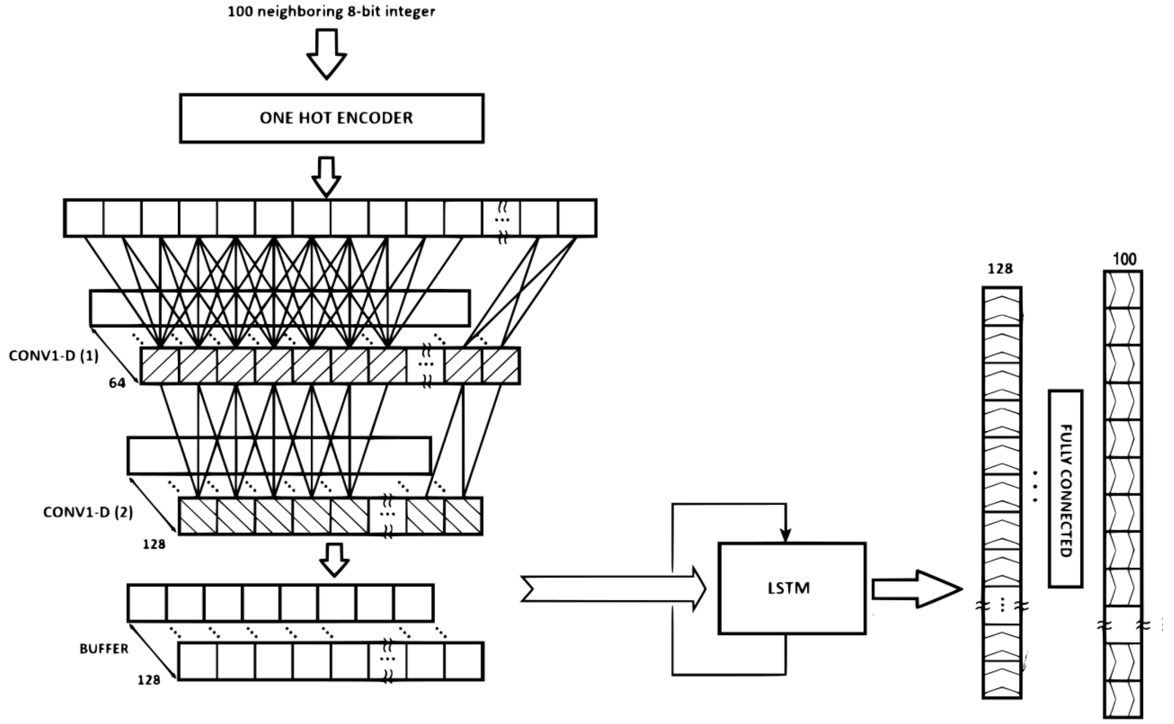


Figure 5.1: Complete working of the machine learning model

5.2 Machine Learning Model

To capture dependencies amongst the bits, we assume there are N features that correspond to a single bit-stream generation. Hence, we use convolutional methods to extract these N features that could correspond to the bit-stream generation. The dependencies between these features are further calculated by the LSTM model as described in Figure 5.1. Finally, predictions are made based on learning the features which provide an accuracy to our model. Comparing this prediction probability with guessing probability captures the advantage of our machine learning model.

5.2.1 Extraction of features via CNN

Convolutional Neural Network [106] is a specialized type of neural network designed for processing structured grid data, such as images. The key components of CNNs include:

- Convolutional Layers: These layers apply convolution operations to the input data, extracting features by sliding filters (kernels) over the input.

- Activation Functions: After convolution, an activation function (like ReLU or sigmoid) introduces non-linearity to the model.

- Pooling Layers: These layers reduce the dimensionality of the feature maps, retaining essential information while discarding less important details.

In an attempt to explain CNN to a quantum optics experimentalist [107] , we draw out an analogy. To paint a picture, one can visualise the convolutional layers being analogous to optical filters used in the experiments, feature extraction being analogous to choosing the degree of freedom of photons to study, non-linear interactions of the state being analogous to non-linear activations, pooling being analogous to the focus of study, say fock states. This is one reason I believe that quantum random number generators are quite useful for optical monte-carlo simulations.

5.2.2 LSTM Model

As the name suggests, LSTM looks out for both long-term and short-term temporal dependencies using laws of differentiation. This is a better choice of temporal modeling compared to conventional recurrent neural networks (RNNs) solving the vanishing gradient problem with an extra cell for long-term context. LSTM is the basic building block of its model. The cells concatenated horizontally form the information highway for context where the output of one cell is input to the next cell. In this section, we describe its working in detail which is also illustrated in Figure 5.2.

5.2.2.1 Working of the LSTM cell

Since we are interested in catching temporal dependencies, we need to understand how the context (previous bits in our case) based prediction happens. For contextual information to be available at hand, we look into how this context is stored in the long and short-term memories of the LSTM cell. Long term stores context for longer temporal correlations between the bit-stream while the short term stores context for the last few recurring bits. However, both long and short-term cells influence each other. All this working of the time-series forecasting model can be understood in terms of the simplest recurring unit of LSTM, the LSTM cell. Succinctly, it is an information highway for context (systematic biases of the optical equipment used reflected in previous bits) to find long-term and short-term dependencies of the n^{th} bit-stream on previous bit-streams. Now we focus on how the long-term context is stored in the memory of the LSTM cell. The cell state C_{t-1}, in Figure 5.2, represents the information stored in the long-term memory. The hidden state h_{t-1} depicts the information stored in the short-term memory. For the next cell with new input information as x_t, a non-linear activation of sigmoid function is applied to the cell state based on input hidden state h_{t-1}. This activation is zero if x_t provides no new information compared to h_{t-1} and one otherwise. Thus, this activation updates the cell state, C_{t-1} providing the amount of relevance of the new information x_t and h_{t-1}. Once the long-term memory, the cell state C_{t-1} is updated, the cell state is transferred onto the next step. Here, the input to the cell state is defined pertaining to the relevance of the new input information, x_t. After the input has been updated, the next part of the LSTM cell, the output

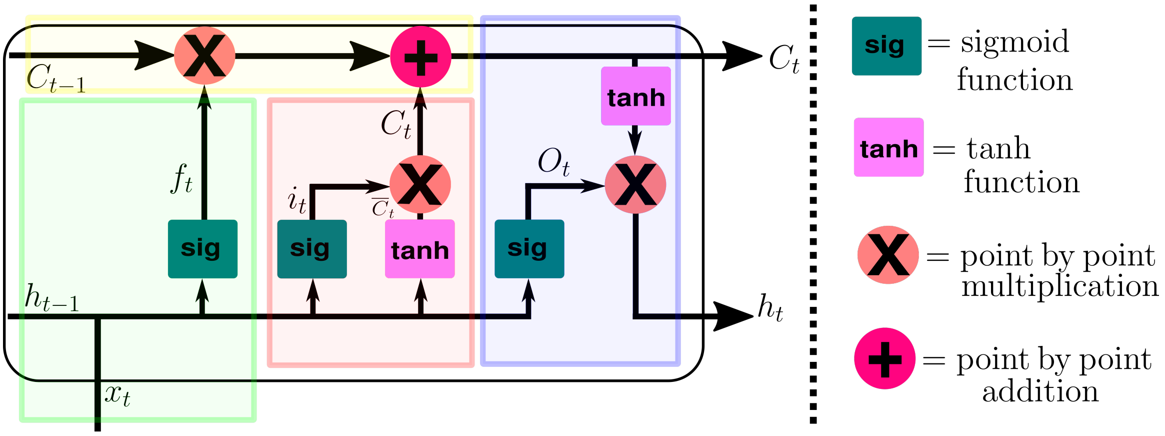


Figure 5.2: Detailed working of an LSTM cell: Green box represents the forget gate, red box represents the input gate, blue ox represents output gate, yellow box represents the cell state: operations performed on the input state are labelled on the right side of the dashed line.

gate, does two things. One, it generates the output for the next bit prediction, y_t. Secondly, it updates the short-term memory h_t based on the h_{t-1}, x_t and C_t through appropriate activations which serve as input to the next recurring gate.

5.3 Measures of Algorithmic Randomness

An algorithmically random string is one not producible from a description significantly shorter than itself, when a universal computer is used as the decoding apparatus. As per Calude, testing the randomness property is computationally hard. Chaitin's definition requires that there should be no algorithmic way to recognize which strings are random. A string is random if it cannot be algorithmically compressed. Furthermore, all incompressible strings do possess all conceivable effectively testable properties of stochasticity.

5.3.1 Kolmogorov Complexity

Kolmogorov complexity is a measure of computability. It is a theoretical concept in computer science based on Turing computability. The Kolmogorov complexity K(x) of a string x is defined as

$$K(x) = min \ \ (|p| : U(p) = x) \tag{5.1}$$

where U is a universal Turing machine, p is a program, and $|p|$ is the length of the program. As an example, consider the bit-stream, “0101010101010101” . The typical and shortest method to describe the text, excluding common overheads, can be easily verified as “01” ×8. Practically, the Kolmogorov complexity can be approximated by using compression algorithms. This approximation also limits our case of deciphering

QRNGs from PRNGs and reduces the efficacy of all algorithms to efficiently calculable measures as stated in the no-go theorem [105]. Amongst the LZ-compression family [108], we specifically focus on the predecessor of all these algorithms, the LZ-76 algorithm. A string with high Kolmogorov complexity is considered random or incompressible, while a string with low complexity can be described succinctly.

5.3.1.1 Working of LZ-76 algorithm

Lempel-Ziv (LZ) compression technique was first developed in 1976 as a lossless data compression technique. The length of the output produced by LZ-76 can be seen as an upper bound for the Kolmogorov complexity of the input string [109]. Specifically, if x is compressed using LZ-76 to produce a string y, then:

$$K(x) \leq |y| + O(\log(y)) \tag{5.2}$$

This means that the compressed representation y gives us insight into the complexity of x. The core idea behind it is to replace repeated occurrences of bit-streams with references to a single copy, thereby reducing the overall size of the bit-stream. In essence, LZ-76 compresses the bit-stream by identifying and encoding sequences of bits that appear multiple times. The algorithm dynamically builds a dictionary as it processes the input bit-stream, allowing it to adapt to varying bit-stream patterns.

The compression process in LZ-76 can be broken down into several steps:

- The algorithm reads the input bit-stream one character at a time while maintaining a sliding window containing previously processed characters of the bit-stream.
- For each new character read, it searches within the sliding window for the longest match (the longest sequence that matches the current position). If a match is found, it generates a keyword in the dynamic dictionary representing this match.
- If no match exists, a keyword representing the new pattern in the bit-stream is created. If a match is found, a reference keyword is generated that indicates both the distance to the start of the match and its length.
- The number of unique elements in the dictionary classifies as our definition of Kolmogorov complexity.

5.3.2 Borel Normality

Borel normality is a concept in probability that provides a way to assess the randomness of a binary sequence (or bit-stream) [110]. A binary sequence is considered Borel normal if, in its infinite expansion, every finite sequence of bits appears with the expected frequency. A binary sequence $x = x_1, x_2, x_3, ..$ is said to be Borel normal if for every finite binary string s of length n, the proportion of occurrences of s in the first N bits of the sequence approaches 2^n as N tends to infinity. Mathematically, this can be expressed as,

$$\lim_{n\to\infty} |\frac{count(s, x_1, x_2, , x_N)}{N} - \frac{1}{2^n}| \geq \epsilon \tag{5.3}$$

where $count(s, x_1, x_2, .., x_N)$ is the number of times the string s appears in the first N bits and $\epsilon > 0$.

5.4 Experiment

We first choose a known bad PRNG (in terms of both unpredictability and statistical correlations), namely, linear congruential generator (LCG) to test the quality of our model. The parameter m defined in the recurrence relation of LCG is varied for five different values [111]. Once this preliminary test is done, we develop faith in our model that it is working correctly and the generated results are not ad-hoc. Furthermore, we post-process our bad PRNG with a Toeplitz hash function to break the statistical correlations by re-distributing the unpredictable bias of the said LCG. Following this, we test our post-processed PRNG with machine learning analysis for unpredictability and NIST-STS for statistical correlations. This provides a measure to check on the properties of the post-processing method. With these initial consistency checks on our machine learning model, we perform the machine learning analysis on the PRNG and QRNG generated. Specifically, just using the bit-stream from the QRNG generated in the previous chapter provides us an entanglement-based QRNG excluding device-independence. Dataset A and Dataset B correspond to different Bell parameter values, namely S= 2.78 and 2.47. For the choice of a good PRNG, we use the CS-PRNG ChaCha20, previously discussed, which provides an unpredictability advantage over typical PRNGs like LCG. Both the QRNG and CS-PRNG are tested initially with NIST Test suites to check for statistical correlations. This is done both before and after post-processing for QRNGs, while CS-PRNGs pass the NIST without the need for post-processing. This is followed by machine learning analysis to decipher the two using measures of unpredictability. Since there could be biases pertaining to the model being tailored to the random number generator used, we also cross-verify it using measures of algorithmic randomness, namely Kolmogorov complexity and Borel normality. To check the measure of quality of quantifiers like Kolmogorov complexity and Borel normality, we provide a reference with the known bad PRNG, LCG.

5.5 Results and discussion

In this section, we discuss four methods to quantify randomness. The three random numbers used are LCG as a PRNG, entanglement-based QRNG, and ChaCha20 as a CS-PRNG. To verify the method of measuring randomness, we first verify our four methods using LCG. This is followed by actual testing of a good QRNG, namely entanglement-based QRNG and ChaCha20, a CS-PRNG.

5.5.1 NIST Results

Coming to testing measures, the preliminary need for random number generators to be useful is that they should be free from statistical correlations, which we verify using the NIST statistical test suite. In case the random number generator fails these tests, we post-process these bit-streams with the Toeplitz hash function to extract randomness out of them, making them free of statistical correlations, which can be verified using the NIST statistical test suite again.

5.5.1.1 Un-processed

The results for un-processed PRNG (LCG), QRNG (Entanglement based) and CS-PRNG are highlighted in Table 5.1. The length of the initial un-processed bit-stream is chosen to be 5 million (M). With this length, it is seen that LCG and Entanglement based QRNG both perform poorly on the independence of statistical correlations. However, ChaCha20 is free from statistical correlations. Being a CS-PRNG, it is expected of it. The same is expected for LCG pertaining to known correlations (based on the Markov chain model) and QRNG pertaining to experimental imperfections which skew the bit-stream towards having statistical correlations.

5.5.1.2 Post-processed

To remove the biases reflected in statistical correlations, we post-process the bit-stream and the results are highlighted in Table 5.2. The results for ChaCha20 being post-processed and verified with NIST are not discussed on the premise of not being relevant. One can clearly see that the NIST statistical test suite is passed for both LCG based PRNG and entanglement based QRNG. In other words, one is unable to decipher the kind of random number generator tested using NIST statistical test suites. This is the first evidence in verifying the claim made by the no-go theorem about the impossibility to distinguish between pseudo and quantum randomness.

Tests included	QRNG (Dataset A)	QRNG (Dataset B)	PRNG (m=24)	PRNG (m=26)	PRNG (m=28)	PRNG (m=30)	PRNG (m=32)	ChaCha20
Approximate Entropy	0	0	0	0	0	0	0	0.92
0 Block Frequency	0	0	1	1	1	1	0	0.24
Cumulative Sums	0	0	0	1	0	0	0	0.68
FFT	0	0.139	0	0	0	0	0	0.24
Frequency	0	0	0	0.9	0	0	0	0.59
Linear Complexity	0.686	0.377	0.013	0.709	0	0.011	0.064	0.13
Longest Runs	0	0	0	0	0	0	0	0.79
Non Overlapping Template Matching (KS Test)	0	0	0	0	0	0	0	0.04
Overlapping Template Matching	0	0	0	0	0	0	0	0.05
Random Excursions (KS Test)	0	0	0	0.958	0	0	0	0.04
Random Excursions Variant (KS Test)	0	0	0	0.220	0	0	0	0.11
Rank	0.062	0.816	0	0.607	0	0.429	0.143	0.20
Run's	0	0	0	0.093	.033	0	0	0.37
Serial	0	0	0	0	0	0	0	0.25
Universal	0	0	0	0	0	0	0	0.45

Table 5.1: NIST Test results for pre-processed bit-stream of length 5 M; Only ChaCha20 passes all 15 tests

Tests included	QRNG (Dataset A)	QRNG (Dataset B)	PRNG (m=24)	PRNG (m=26)	PRNG (m=28)	PRNG (m=30)	PRNG (m=32)
Approximate Entropy	0.985	0.546	0.765	0.149	0.114	0.068	0.569
Block Frequency	0.380	0.129	0.727	0.249	0.144	0.553	0.708
Cumulative Sums	0.973	0.557	0.494	0.01	0.695	0.557	0.092
FFT	0.979	0.973	0.598	0.508	0.119	0.575	0.627
Frequency	0.840	0.465	0.738	0.01	0.623	0.881	0.101
Linear Complexity	0.840	0.965	0.066	0.348	0.077	0.749	0.554
Longest Runs	0.060	0.966	0.882	0.005	0.260	0.585	0.078
Non Overlapping Template Matching	0.069	0.325	0.139	0.189	0.098	0.101	0.298
Overlapping Template Matching	0.721	0.590	0.851	0.098	0.719	0.262	0.642
Random Excursions	0.843	0.383	N.A.	N.A.	0.516	0.641	0.166
Random Excursions Variant	0.435	0.621	N.A.	N.A.	0.941	0.285	0.326
Rank	0.993	0.084	0.310	0.092	0.661	0.453	0.497
Run's	0.858	0.325	0.890	0.912	0.253	0.212	0.674
Serial	0.403	0.356	0.640	0.194	0.033	0.066	0.880
Universal	0.285	0.210	0.239	0.447	0.849	0.479	0.756

Table 5.2: NIST Test results for post-processed bit-stream of length 1.2 M

5.5.2 Machine Learning Results

As a second measure to quantify randomness, we use a hybrid of space-time machine learning model to evaluate the unpredictability of the bit-stream. The total data bit-stream (of 5 million (M)) is divided into two parts. 4 M for training the model and 1 M for making predictions whose results are highlighted in Table 5.3.

5.5.2.1 un-processed

With varying m values of LCG, one can see that the model is able to predict the bit-stream 8 out of 10 times with more than 50% accuracy. This follows from the recursive relation used in the definition of LCG. Since our model is able to catch these recursive properties, this verifies our model and develops our faith in the model being used. The 50% accuracy is quite large compared to simple guessing probability ($\frac{1}{2^8} = 0.390625\%$). In quantification of unpredictability amongst QRNG and CS-PRNG in the unprocessed bit-stream, QRNGs show high unpredictability and the machine learning algorithm is able to perform no better than guessing probability. On the contrary, machine learning algorithm is able to catch dependencies in the next-bit prediction for ChaCha20 with a 6% probability. This is a major security concern as it is being used in many ongoing encryption services including transport security layer in gmail [14], encryption in virtual private network (VPN) services such as Nord VPN [112].

Tests included	PRNG (m=24)	PRNG (m=26)	PRNG (m=28)	PRNG (m=30)	PRNG (m=32)
$P_{ml}a_1c_1$	0.327%	89.076%	95.563%	65.938%	59.958%
$P_{ml}a_2c_2$	71.331%	87.184%	73.537%	51.408%	1.605%

Table 5.3: Machine Learning Model trained on a known PRNG (LCG) for model verification

Tests included	QRNG (Dataset A)	QRNG (Dataset B)	ChaCha20
P_{ml}	0.463%	0.414%	6.038%
P_g	0.3906%	0.3906%	0.3906%

Table 5.4: Testing Un-processed Quantum RNG and ChaCha20 (CS-PRNG) against machine learning model, P_{ml} : next bit prediction probability by machine learning (ml) model and P_g : next bit prediction probability on mere guessing($= \frac{1}{2^8}$)

5.5.2.2 Post-processed

For the post-processed bit-stream, machine learning algorithms as a metric perform poorly as they are unable to decipher any of the post-processed random bits for unpredictability. The unpredictability for LCG is poor for unprocessed and after post-processing it abruptly becomes completely unpredictable against hybrid machine learning algorithms. The same effect is seen with NIST-STS. For QRNGs and CS-PRNGs, the machine learning algorithm is also unable to predict the bit-stream. This reflects that machine learning algorithms as a measure of unpredictability are not able to distinguish between PRNGs, QRNGs, and CS-PRNGs once they are post-processed. This is the second claim that supports the no-go theorem. Researchers have tested random numbers against more complex models like transformers [113] and have shown that they also are unable to predict QRNGs. This supports our results and the no-go theorem as well. From a logical inference, one can say that post-processing of bit-streams heavily influences the quality of random numbers on statistical independence and unpredictability. This is true both for PRNGs and CS-PRNGs where both are intertwined. For QRNGs, the unpredictability measure checks the process rather than the output bit-stream. Hence, it is difficult to say for QRNGs whether the quantum unpredictability or randomness extraction has a major influence on the output bit-stream. To evaluate this, we went a little further by changing the post-processing length to different lengths and testing them against machine learning algorithms as shown in Tables 5.5 and 5.6. One can see that post-processing length doesn't answer this question but bolsters our faith in the no-go theorem.

Tests included		PRNG (m=24)	PRNG (m=26)	PRNG (m=28)	PRNG (m=30)	PRNG (m=32)
P_{ml}	l_{1M}	0.494%	0.458%	0.410%	0.361%	0.373%
P_{ml}	$l_{1.2M}$	0.351%	0.472%	0.361%	0.301%	0.371%
P_{ml}	$l_{1.5M}$	0.361%	0.385%	0.313%	0.409%	0.385%
P_{ml}	$l_{2.0M}$	0.403%	0.331%	0.439%	0.403%	0.409%

Table 5.5: Variation of post-processing length and testing with ML model

5.5.3 Kolmogorov Complexity

We discuss the Kolmogorov complexity as a third measure of randomness. Practical calculability methods to approximate Kolmogorov complexity put the constraint of effectively calculable measures in the otherwise generalized no-go theorem.

Tests included	QRNG (Dataset A)	QRNG (Dataset B)	ChaCha20
P_{ml} l_{1M}	0.470%	0.386%	0.518%
P_{ml} $l_{1.2M}$	0.311%	0.391%	0.351%
P_{ml} $l_{1.5M}$	0.377%	0.369%	0.353%
P_{ml} $l_{2.0M}$	0.337%	0.307%	0.563%

Table 5.6: Variation of post-processing length and testing with ML model

5.5.3.1 LCG results

For the unprocessed bit-stream, we see no clear pattern between the different levels of compression. However, in the post-processed bit-stream, every sequence converges to a similar complexity. This shows that post-processing influences our results heavily and brings every m value to the same footing.

Files	PRNG (m=24)	PRNG (m=26)	PRNG (m=28)	PRNG (m=30)	PRNG (m=32)
Pre-processed	1.11	1.20	0.717	1.239	1.258
Pre-Processed	1.254	1.105	1.257	1.294	0.682
Post-processed	1.382	1.382	1.382	1.382	1.382
Post-processed	1.382	1.382	1.382	1.382	1.382

Table 5.7: Kolmogorov Complexity calculated via LZ compression

5.5.3.2 QRNGs and CS-PRNG results

For the un-processed bit-stream, no clear evidence of difference between QRNG and CS-PRNG is seen on compression as a measure. However, analogous to post-processing results for LCG, results for post-processed QRNG and ChaCha20 bit-streams show similar compression ratios. This fact is further supported by the result that the method used in extracting randomness (Toeplitz hash function) itself employs a random number generator whose complexity, separately measured, is 1.382. Thus, it overrides any information that could have remained in the raw bit-stream. This provides our third evidence of the no-go theorem.

5.5.4 Borel Normality

We test different bit-streams, PRNG, QRNG, and CS-PRNG, for Borel Normality as a fourth measure of randomness [110]. Mathematically, we simply check for the condition Eq. 5.3 that whether it holds true at different bit-levels. We restrict ourselves

Files	QRNG (Dataset A)	QRNG (Dataset B)	ChaCha20
Pre-processed	1.347	1.349	1.349
Post-processed	1.382	1.382	1.382

Table 5.8: Kolmogorov Complexity calculated via LZ compression

to the 4-bit level where the 4-bit level bit-stream has $\frac{1}{2^4}$ probability for every possible 4-bit pattern. This restriction is a practicality and sample space constraint and thus can be put under the constraint, “efficiently calculable measures”.

5.5.4.1 Un-processed

We see that QRNGs don’t adhere to the strict condition of Borel Normality at any bit-level in the unprocessed bit-streams, indicating that quantum unpredictability isn’t similar to computational unpredictability. We also see that ChaCha20 clearly passes the Borel Normality criteria at every bit-level, indicating good compatibility with computational measures of randomness. No consistent pattern can be drawn from different PRNG bit-streams.

Number of bit-streams	PRNG (m=24)	PRNG (m=26)	PRNG (m=28)	PRNG (m=30)	QRNG (Dataset A)	QRNG (Dataset B)	ChaCha20
1-bit	1	1	1	0	0	0	1
2-bit	0	1	1	1	0	0	1
3-bit	0	1	1	1	0	0	1
4-bit	0	0	1	1	0	0	1

Table 5.9: Borel Normality Tests on Un-processed Data

5.5.4.2 Post-processed

For the post-processed bit-streams, one can clearly see that all the bit-streams irrespective of their origin (PRNG, QRNG or CS-PRNG) pass the Borel normality criteria. This is the fourth measure that supports the claims made by the no-go theorem. In addition, we notice the drastic change in Borel Normality criteria for QRNGs for un-processed and post-processed bit-streams. This is intuitive as well; the more unpredictable the bit-stream, the less it should adhere to a uniform distribution. We are also able to verify this as post-processed Borel Normality criteria show an anti-correlation to un-processed quantum bits (arising because of quantum unpredictability).

Number of bit-streams	PRNG (m=24)	PRNG (m=26)	PRNG (m=28)	PRNG (m=30)	QRNG (Dataset A)	QRNG (Dataset B)	ChaCha20
1-bit	1	1	1	1	1	1	1
2-bit	1	1	1	1	1	1	1
3-bit	1	1	1	1	1	1	1
4-bit	1	1	1	1	1	1	1

Table 5.10: Borel Normality Tests on Post-processed Data

5.6 Summary

We provide three major results with this study. Firstly, we provide four independent evidences of the no-go theorem against a chosen QRNG and PRNG. Typically, the initial work stating the no-go theorem [105] considered only two measures, namely, Kolmogorov complexity and Borel normality conditions. We reproduce similar conclusions with an additional statement referring where exactly the definition of effectively calculable measures constraint comes into the picture. We have appended their conclusions in a more rigorous manner from a practical standpoint using both NIST-STS and ML models. In an ideal case, one should use multiple hybridizations of space-time complexity models where the space model extracts the features out of the bit-stream while the time model finds out dependencies between the features so that our negative results are model-agnostic. Secondly, we see that ChaCha20 shows weakness against machine learning models despite the unprocessed bit-stream passing NIST-STS. This weakness could be further explored with advanced machine learning algorithms. Thirdly, we see how quantum unpredictability anti-correlates to computational standards as discussed in the introduction. To claim the advantage of QRNGs over PRNGs, QRNGs are unpredictable as they couldn't be caught by machine learning algorithms. However, they cannot be used without post-processing as they fail NIST-STS. After post-processing, one is unable to decipher between the QRNG and PRNG from the bit-stream.

"The greatest victories are the least noisy."

Mother

6 Questioning the quantum advantage in QRNGs

6.1 Introduction

In the previous chapter, we saw no clear advantage of QRNGs over PRNGs using computational methods. In this chapter, we explore the advantage of QRNGs over the other popular alternative, TRNGs (true random number generators). Both QRNGs and TRNGs offer their advantage on the unpredictability front and being compact and less resource-intensive, with the striking difference that QRNGs have a quantum calibration metric in addition to the bit streams. The same cannot be said for TRNGs, which translates to a lack of trust in QKD technologies. A simple attack like the detector blinding attack [114] could be exploited by Eve, pertaining to the lack of a calibration metric. Other than unpredictability, both QRNGs and TRNGs need to be tested against typical NIST-STS for statistical correlations. For the choice of QRNGs and TRNGs, we choose noise-based random number generation processes, namely, shot noise-based using a LASER (Poissonian statistics) and optical noise-based using a LED (thermal sources of light). The difference between the two types of optical sources from a noise perspective is reflected in their quadrature space distributions, as shown in Figure 6.1. With this background in mind, we discuss the theoretical aspects required for understanding both kinds of random number generators.

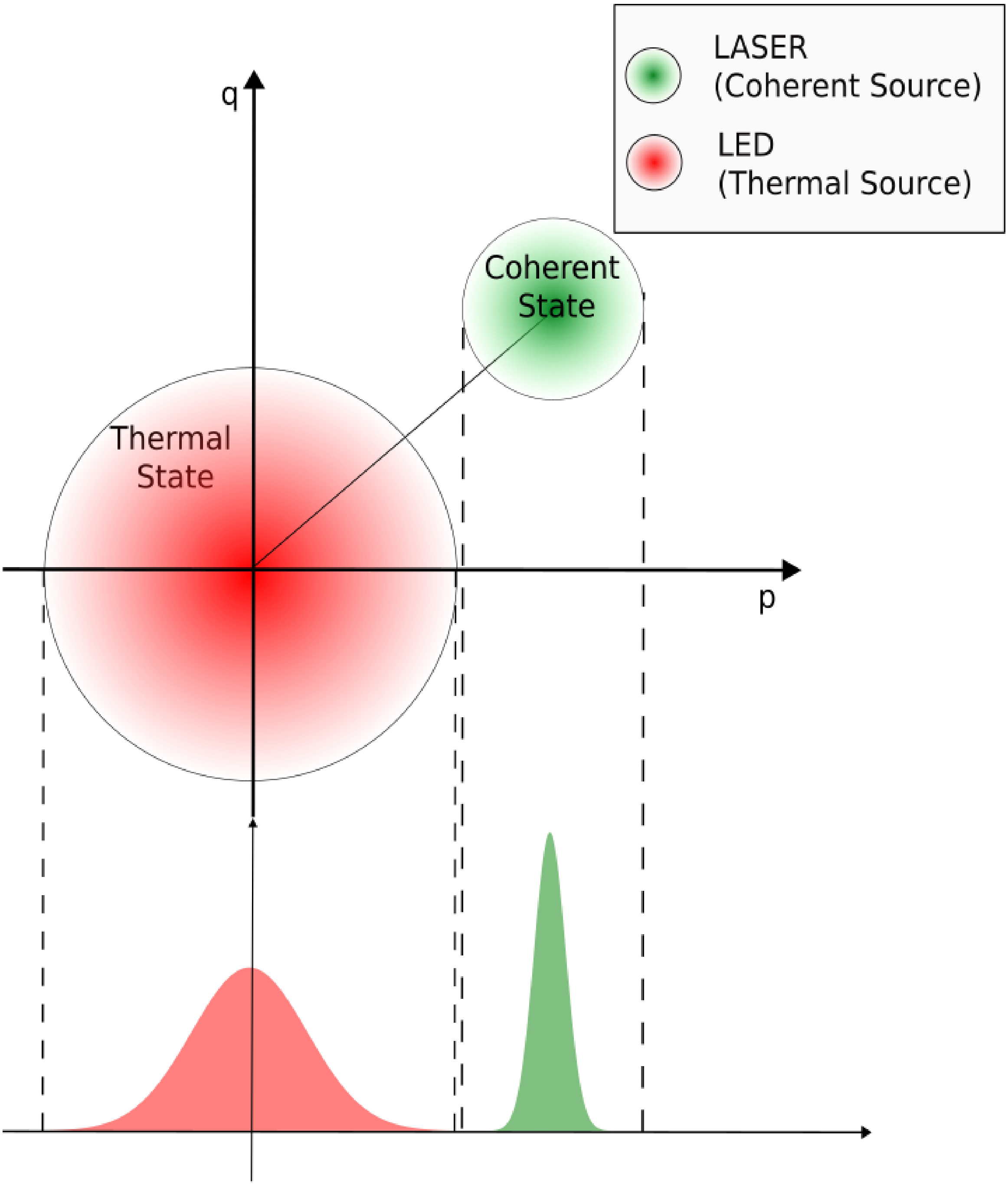


Figure 6.1: Comparing two different sources for optical noise based random number generation: TRNG (LED, Thermal-noise) and QRNG (LASER, Shot-noise), top shows the conceptual difference in the type of uncertainty relations followed by both the sources in the quadrature space while the bottom shows the corresponding one-dimensional projections of states as Gaussian distributions.

6.2 Theoretical Background to QRNG generation

This section is divided into four subsections. First, we discuss the common detection scheme for both sources, namely the balanced homodyne detection. We derive the difference signal being proportional to quadrature components in a θ rotated basis of measurement. Secondly, we discuss how entropies (related to measures of security) can be calculated from standard deviations in these quadratures. Thirdly, we discuss how random bits are generated from these quadrature uncertainties. Lastly, we relate how bits generated per sample and entropy relate to ϵ_{hash} security.

6.2.1 Balanced Homodyne Detection

Balanced homodyne detection is a technique of measuring the field quadrature (amplitudes) of a field instead of the intensity as done with single photon detectors. The scheme of balanced homodyne detection is depicted in Fig 6.2. The signal whose quadratures are to be measured interferes with a strong classical field called the local oscillator at a 50:50 beam-splitter. It is also assumed that the local oscillator should be intense compared to the signal for providing a precise phase reference. After the optical mixing of both the signal and local oscillator, each emerging beam is received by the photo-diodes. The detected photo-currents are subtracted and the subtracted signal is amplified and recorded. The subtracted signal, experimentally observed, can be written as

$$I_{21} = I_2 - I_1 \tag{6.1}$$

which in photon numbers, for perfect detection efficiency, can be written as

$$\hat{n}_{21} = \hat{n}_1 - \hat{n}_2 \tag{6.2}$$

where

$$\hat{n}_1 = \hat{a}'^{\dagger}_1 \hat{a}'_1 \qquad \text{and} \qquad \hat{n}_2 = \hat{a}'^{\dagger}_2 \hat{a}'_2 \tag{6.3}$$

are the number operators (proportional to intensity) of the output ports of the beam-splitter. From a theoretical perspective, we show that $\hat{n}_{21}$ is proportional to $\hat{q}_\theta$ and the magnitude of the complex amplitude, α_{LO} of the local oscillator. The output mode operators $\hat{a}_1^{'}$ and $\hat{a}_2^{'}$ of the fields emerging out of the beam-splitter can be written in terms of the input mode operators of the field, $\hat{a}$ and α_{LO} as

$$\hat{a}_1^{'} = \frac{1}{\sqrt{2}}(\hat{a} - \alpha_{LO}) \qquad \hat{a}_2^{'} = \frac{1}{\sqrt{2}}(\hat{a} + \alpha_{LO}) \tag{6.4}$$

The difference signal $\hat{n}_{21}$ in terms of $\hat{a}$ and α_{LO} can, then be written as:

$$\hat{n}_{21} = \alpha^*_{LO}\hat{a} + \alpha_{LO}\hat{a}^\dagger \tag{6.5}$$

Further, the signal mode operator, $\hat{a}$, can be written in terms of quadratures q and p as

$$\hat{a} = \frac{1}{\sqrt{2}}(\hat{q} + \iota\hat{p}), \tag{6.6}$$

and the local oscillator for a coherent state, α_{LO}, can be defined as

$$\alpha_{LO} = |\alpha| \exp \iota\theta. \tag{6.7}$$

Incorporating the above definitions of the local oscillator and signal in the difference signal equation, we get

$$\begin{aligned}
\hat{n}_{21} &= \frac{1}{\sqrt{2}}|\alpha^*| \exp(-\iota\theta)(\hat{q} + \iota\hat{p}) + |\alpha| \exp(\iota\theta)(\hat{q} - \iota\hat{p}) \\
&= \frac{1}{\sqrt{2}}|\alpha|\Big((\cos\theta - \iota\sin\theta)(\hat{q} + \iota\hat{p}) + (\cos\theta + \iota\sin\theta)(\hat{q} - \iota\hat{p})\Big) \\
&= \sqrt{2}|\alpha|\,(\hat{q}\cos\theta + \hat{p}\sin\theta) \\
&= \sqrt{2}|\alpha|\hat{q}_\theta
\end{aligned} \tag{6.8}$$

where

$$\hat{q}_\theta = \hat{q}\cos\theta + \hat{p}\sin\theta, \tag{6.9}$$

and θ is the phase between the signal and the local oscillator. Thus, we have shown that

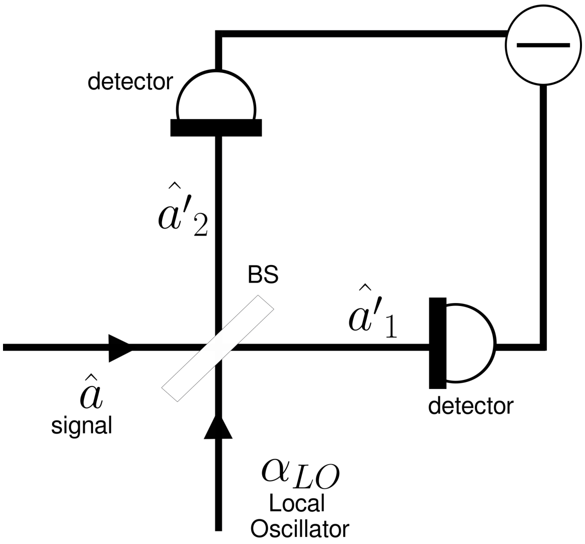


Figure 6.2: Theory of balanced homodyne detection

6.2.2 Calculating entropy of different sources

Typically for quadrature q and p, we have the Heisenberg uncertainty relation given by

$$\sigma(q)\sigma(p) \geq \frac{1}{2}\hbar. \tag{6.10}$$

The translation of this equation in terms of entropy [3] is

$$h(q) + h(p) \geq \log(e\pi\hbar), \tag{6.11}$$

where $h(q)$ is defined by the typical definition of entropy in terms of probability distribution $\Gamma(x)$ as

$$h(q) = -\int_{-\infty}^{+\infty} \Gamma(x)\log(\Gamma(x))\,dx. \tag{6.12}$$

For a coherent state the quadratures are Gaussian distributed [115] and are defined in terms of mean $\overline{x}$ and standard deviation $\sigma(x)$ as

$$\Gamma(x) = \frac{1}{\sqrt{2\pi\sigma(x)^2}}\ \exp\left(\frac{-(x-\overline{x})^2}{2\sigma(x)^2}\right) \tag{6.13}$$

Maximum entropy is achieved by variables which follow a Gaussian probability distribution [116], which can be easily shown by the method of Lagrange multipliers, and the entropy translates to

$$h(q) = \log(\sqrt{2e\pi\sigma(x)^2}). \tag{6.14}$$

In the case of other distributions, the entropy follows the inequality given by

$$h(q) \leq \log(\sqrt{2e\pi\sigma(x)^2}). \tag{6.15}$$

6.2.3 Generation of random numbers

To generate random numbers from the subtracted signal which is a Gaussian for both thermal and coherent states, we need to sample the Gaussian curve. A simple method involves dissecting the experimentally generated Gaussian curve into two parts with the mean as the centre. The data points on the right of the Gaussian are assigned bit “0” while the ones on the left are assigned bit “1”. Since the point has a quantum mechanical origin to be either on the left or right of the Gaussian curve, the quantum phenomenon in this underlying process dictates the unpredictable nature of generated random bits. This slicing around the mean can be extended to N slicing (pertaining to the resolution of ADC used) such that one can get higher bits per sample, denoted as B. This is an advantage over discrete variables where B is typically set to 1 per qubit. The experimental data generated is cross-verified for statistical correlations as discussed in Section 6.4.3.

6.2.4 Relation between entropy and post-processing

The calculation of entropy is an important measure to keep an eye on how much post-processing is required and how much ϵ_{hash} security a given system can provide. The definition of ϵ_{hash} security is identical for PRNGs and QRNGs. Compared to

PRNGs, QRNGs provide security to a much larger depth by calibrating the process of generation rather than the output bit stream with min-entropy. Typically, the entropy is related to post-processing length l, and the security parameter ϵ_{hash} by the formula,

$$l = \frac{H_{min} * n}{B} - 2\log_2(\epsilon_{hash}), \tag{6.16}$$

where B is bits/sample generated and n is the length of the raw bit-stream.

6.3 Experiment

We perform two experiments with a similar theoretical idea of balanced homodyne detection. The difference between them is about the source of incoming photons. The first source is a LASER, which is a coherent state and shows Poissonian spread of uncertainties in quadrature space. It is a monochromatic source at 633 nm with a very narrow line width. The other source is a typical LED, which is a broadband source whose wavelength response is discussed. Since wavelength plays a role in the security analysis of both the random number generators, we need to develop a common ground to compare the two sources.

6.3.1 Laser based QRNG

The LASER (He-Ne) emits photons at 633 nm. The setup is shown in Figure 6.3. The input beam falls on the beam-splitter and gets divided by amplitude. The beam is coupled to the photo-diodes of the balanced homodyne detector (PDB-435 A) through free space. The power control in each arm and before the beam-splitter is taken care of by the half waveplate and polarising beam-splitter combination. For the sake of brevity, this is not shown in the Figure. Equal distances between the beam-splitter and photo-diodes are ensured by putting one of the mirrors guiding the beam to the photo-diode on the translation stage. After subtracting the signal, an amplifier with a common mode rejection ratio (CMRR) of 30 dB is used to amplify the weak subtracted signal and reject the common electrical noises.

6.3.2 LED based TRNG

The LED has a broadband wavelength response function as shown in Figure 6.5. The beam falls on the beam-splitter and gets divided by amplitude. The beam after the beam-splitter is collected through a coupler to a multimode fibre which is further connected to the photo-diode. The power control in each arm and before the beam-splitter is taken care of by the half waveplate and polarising beam-splitter combination. The reason this works is that polarisation is an independent degree of freedom from the quadratures. Equal distances between beam-splitter and photo-diodes are ensured.

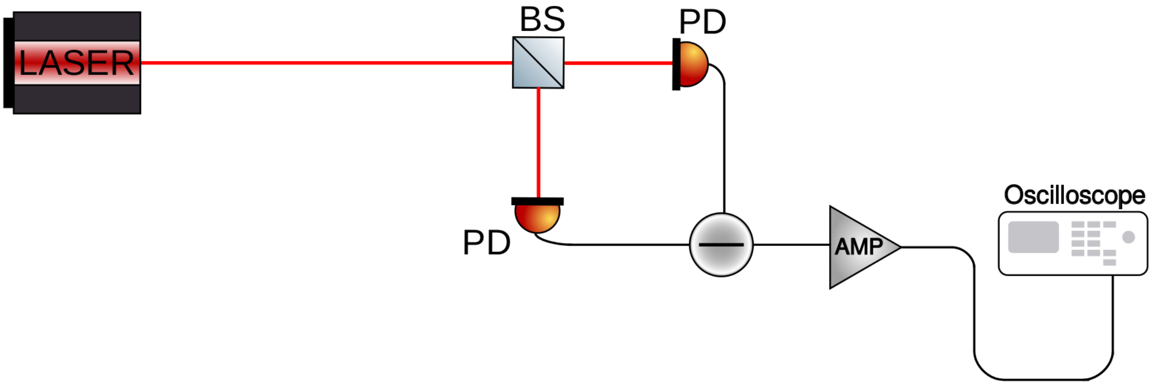


Figure 6.3: Experimental Setup for LASER Shot-noise based QRNG.

After subtracting the signal, an amplifier with a common mode rejection ratio of 30 is used to amplify the weak subtracted signal.

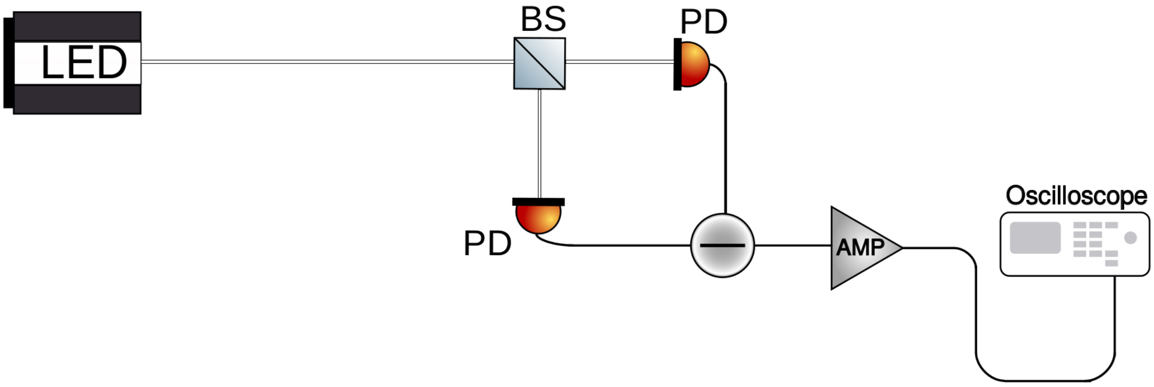


Figure 6.4: Experimental Setup for LED optical-noise based TRNG.

6.3.3 Wavelength response of LED

LED is a broadband source with a spectrum characterised by Figure 6.5. Comparing a broadband source to a mono-chromatic source is a non-trivial exercise. However, since we are only interested in random number generation, we circumvent this with the following technique. The methodology of optical noise is developed for monochromatic coherent sources (pertaining to Poissonian distribution), we modify our LED source to fit the description of a monochromatic source. It works only in this case as we need a reference wavelength for LED instead of the broad spectrum for random number generation. If we were to look for other properties, we were required to follow the typical technique of writing the thermal state in coherent state basis. We calculate the weighted average of the wavelength response of LED by numerically integrating under the curve using trapezoid's method. The weighted average is roughly 517 nm. It adds a minority contribution in comparative security analysis. This is reflected in calculations of the entropy of the broadband LED source and the monochromatic LASER source.

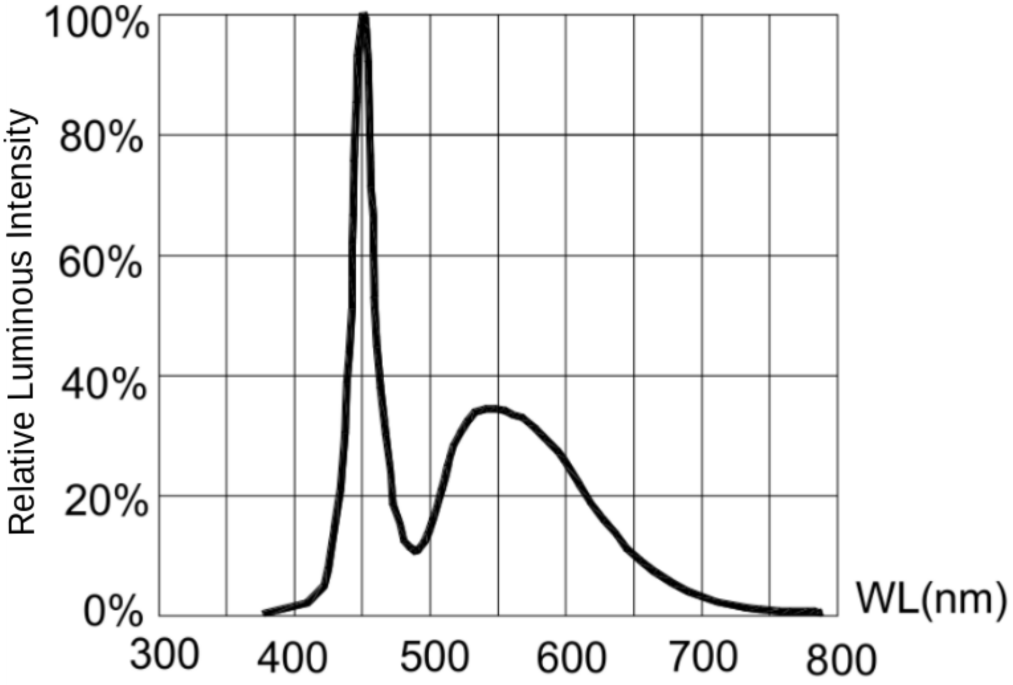


Figure 6.5: Specification sheet of LED (Model: C503B-WAN-CCADB151) describing wavelength spread for a typical white light LED.

6.4 Results and discussion

This section is divided into three parts. Firstly, we calibrate the quantum nature of our LASER source and claim that the same cannot be said for LED-based sources. Secondly, we discuss the advantage of LASER shot-noise based QRNGs over LED optical noise based TRNGs on the entropy front. Thirdly, we discuss the statistical correlations amongst both the raw bit streams generated (from different sources) and how we break these correlations using the Toeplitz hash function.

6.4.1 Verification of Quantum Origin

To verify that the bit-stream generated has a quantum origin, one needs to differentiate the two sources. LASER, being a Poissonian source, generates a coherent state which is the minimum uncertainty area in quadrature-phase space. For thermal sources, typically, the noise is equal in both the quadratures. However, it is larger than that of a Poissonian source. One can show that for a Poissonian source like LASER, mean (proportional to the input power of the beam) equals the variance of a said quadrature, say q. This graph should follow a straight line if the source is Poissonian.

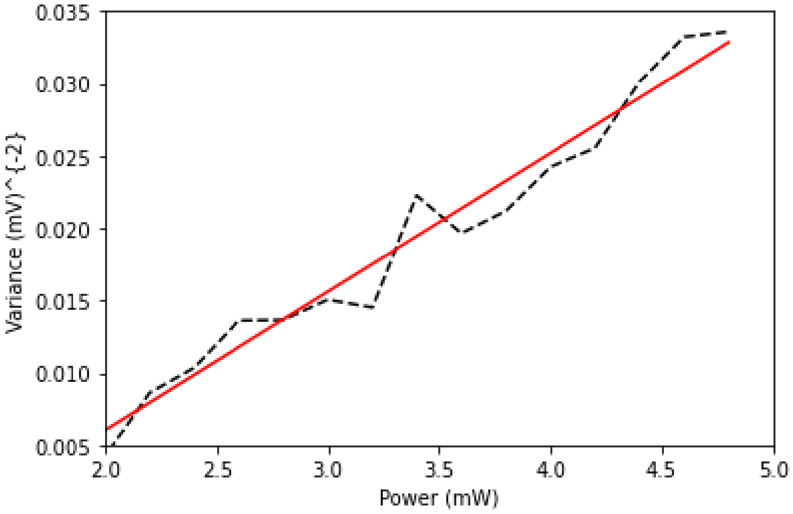


Figure 6.6: Verification of QRNG having Quantum Origin: Poissonian statistics verified by with Power (≈ mean) linearly proportional to variance for LASER shot-noise.

6.4.2 Unpredictability

To calculate the unpredictability of the bit-streams, we cannot use simple conventional methods like NIST 800-90B [117], which calculates the min-entropy of the source as a measure of unpredictability based on Monte Carlo methods. Since Monte Carlo methods use PRNGs to calculate min-entropy, the methodology translates to PRNGs calibrating the unpredictability of QRNGs based on computational hardness. Hence, noise-based QRNGs are often calibrated on the process being used. Here, we calibrate their unpredictability from the width of the Gaussian curve (standard deviation) as previously discussed. The experimental Gaussian curves (both for LASER and LED) are highlighted in the figure below. From these experimental curves, one can calculate entropies [3]. . However, since both experiments were performed at different powers and had different frequencies, the variance of the experimental Gaussians, $V(U)$, doesn't give a direct measure of entropy. To get the variance of quadratures, one needs to scale these variances with a scaling factor ϕ []. They are related as

$$V(U) = \phi V(\hat{q}). \tag{6.17}$$

The scaling factor ϕ depends on many experimental factors by the relation,

$$\phi \approx P_{LO}\kappa^2 g^2 Bhf \tag{6.18}$$

where P_{LO} is the local-oscillator power, κ is the PIN diodes' responsivity (in A W^1), g is the total amplification of the receiver (in V A^1), B is the electronic bandwidth (in Hz), h is Planck's constant, and f is the optical frequency. Other than P_{LO} and f, all other components are considered equivalent (pertaining to negligible contributions to the security formula). The ratio of entropies calculated from the standard deviations of the experimental Gaussians, and accommodated for scaling factor, gives the relative security advantage of QRNGs over TRNGs. This is calculated by the formula,

$$\frac{h(q)_C}{h(q)_T} = \frac{\log(2e\pi\sigma(q)_C^2)}{log(2e\pi\sigma(q)_T^2)} = 3.5. \tag{6.19}$$

The above result clearly shows that the entropy of a coherent LASER source is 3.5 times more than that of a thermal LED source for the application of a random number generator. LASER-based QRNG provides 3.5 times more security than LED-based TRNG. This security is reflected in the length of the bit-stream required to encrypt a message. To encrypt a message, one needs a 3.5 times shorter bit-stream length of the QRNG over TRNG providing the same level of security.

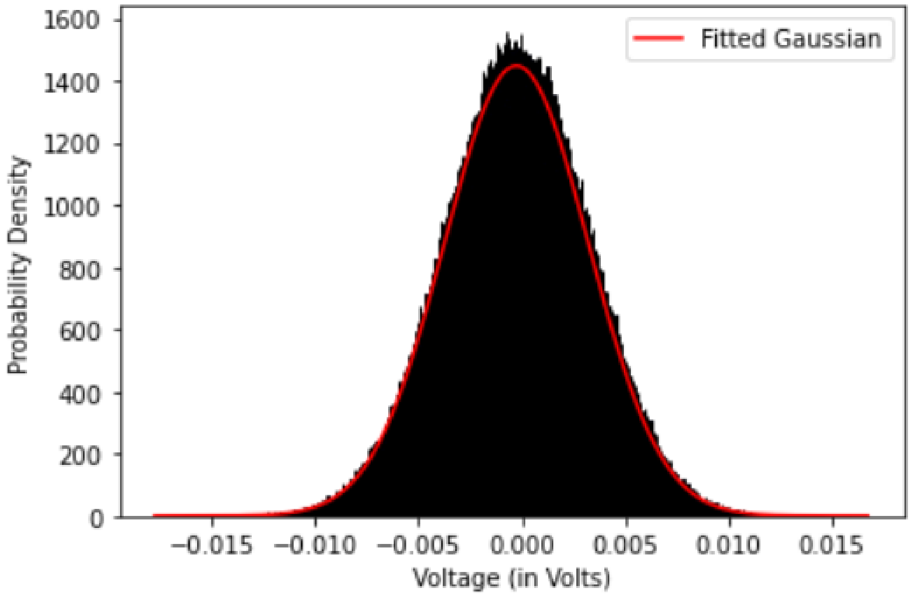


Figure 6.7: Distribution of the subtracted signal for LASER source.

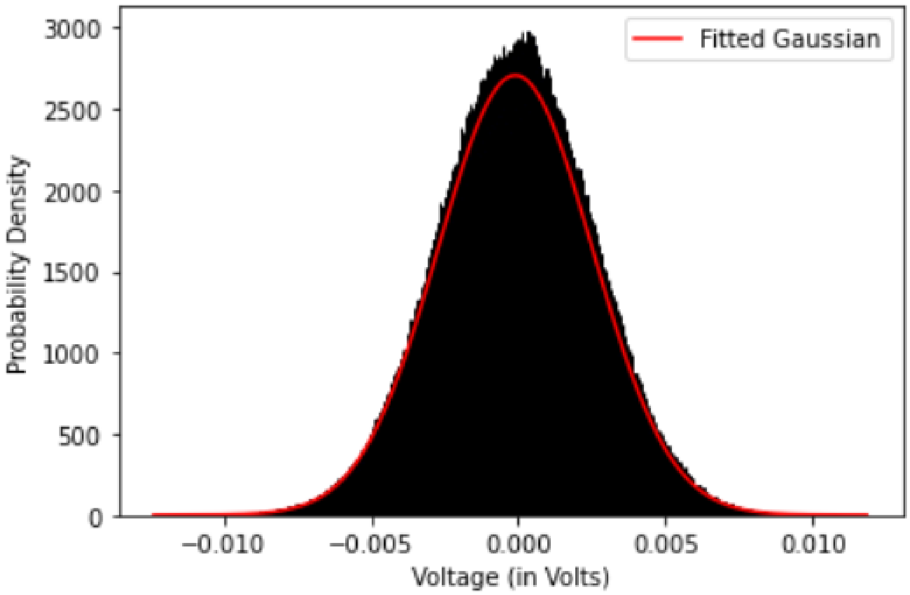


Figure 6.8: Distribution of the subtracted signal for LED source.

6.4.3 Statistical Correlations

Since both QRNG and TRNG have advantages on the unpredictability front, the generated bit-streams pass statistical independence tests. The unprocessed bit-streams generated from both cases fail to qualify for independence of statistical correlations. This is indeed the case. One reason for these statistical dependencies could be the bias we introduce in the assignment of bits (12 bits per sample for 12-bit ADC). These 12 bits generated per sample are the premise for higher bit-rates in continuous-variable QRNGs. To get rid of the statistical dependencies, we post-process the random bits using the Toeplitz hash function as described in Eq. 3.18 in Chapter 3 to be the typical standard used to break the statistical correlations. This is followed by testing for independence against statistical correlations for both types of random numbers generated. Both bit-streams are able to pass the minimum threshold of being good random number generators in the conventional sense as shown in Table 6.1. The table shows the performance of the bit-streams generated by LASER and LED against hypothesis testing, claiming that the said sequence is random. As highlighted by the p-values in different tests, all values are above the threshold test-static $\alpha = 0.01$. This testing supports the fact that the bit-streams generated by both sources don't have measurable statistical correlations.

Tests included	QRNG (LED)	QRNG (Laser)
Approximate Entropy	0.519	0.909
Block Frequency	0.131	0.072
Cumulative Sums	0.286	0.838
DFT	0.603	0.788
Frequency	0.349	0.679
Linear Complexity	0.685	0.635
Longest Runs	0.953	0.741
Non Overlapping Template Matching	0.649	0.977
Overlapping Template Matching	0.231	0.611
Random Excursions	0.721	0.502
Random Excursions Variant	0.535	0.536
Rank	0.528	0.190
Run's	0.993	0.148
Serial	0.589	0.949
Universal	0.850	0.729

Table 6.1: NIST Test results for bit stream of length 1.2 M.

6.5 Summary

In our experimental setup, to achieve higher bit-length per sample, we kept the bits per sample to 12 pertaining to the 12-bit resolution of the ADC used in the oscilloscope for sampling. The bit-stream generated was 5 million (M) bits. The post-processed length is fixed at 1.2 M. Once these three parameters are fixed for both kinds of sources, LASER and LED, only two parameters remain in the Eq. 6.16, namely, min-entropy and ϵ_{hash}. We calculate the min-entropy of both sources from the standard deviation of the fitted Gaussians (which is above the ADC resolution). Comparing the ratio of the standard deviation of both the Gaussians gives an estimate of the relative entropy ratio. As previously stated, the more the entropy of the system, the more secure the random number generator is (security defined by ϵ_{hash}). This gives us insight that the advantage of QRNG over TRNG is in the fact that given the same bit-length, QRNGs provide more security compared to TRNGs in terms of the security parameter, ϵ_{hash}. However, given a bit-stream one is unable to decipher whether the source has a quantum, optical, electronic, algorithmic or stochastic origin. We are only able to catch the weaklings amongst random number generators. The advantage of QRNGs over TRNGs also matches with our intuition that QRNGs are better calibratable compared to TRNGs. This work defines a trade-off between cheap but less secure LED-based TRNG against more secure but expensive LASER-based QRNG. We believe LED-based TRNG is enough for IoT applications as only the amount of bit-length changes for a given security parameter.

"A warrior is not about perfection or victory or invulnerability. He's about absolute vulnerability."

Nolte, Nick

7 Summary

This thesis points to the need for developing a test suite for quantum random number generators. Particularly, we have focused on testing the strength of a QRNG on three different fronts, namely the quantum correlation front, the computational front, and the quantum origin front, to prove its advantage over the conventional alternatives, namely PRNG and TRNG.

7.1 Brief

Chapter 1 sets the foundation of the need for QRNGs in the domain of random number generation. We classify random number generators on different fronts, say on grounds of assumptions of devices involved which provide a trade-off between bit-rate and security of QRNGs or another classification based on whether the random numbers generated are algorithmically random and Turing-computable. Towards the end of the chapter, we define a quantum random number generator and how its quality (in terms of unpredictability and statistical correlations) gets affected over its lifetime.

Chapter 2 develops a quantum toolkit to measure the quantum advantage of QRNGs by discussing different kinds of quantum phenomena. For example, HOM, Quantum Entanglement, and second-order field correlations $g^2(0)$ are quantum measures which can be translated to randomness measures of entropy. Furthermore, we discuss QRNGs developed on weak coherent pulses (as a source of single photons) and compare them against SPDC based single photon sources (SPS). We conclude this chapter by claiming a direct correlation between the quality of the single photon source and the randomness generated via the single photon source. This is reflected in terms of entropy calculated. SPDC, being a better single photon source, provides better quality random bits; however, the process is device dependent.

Chapter 3 discusses the computational toolkit for QRNGs which is borrowed from pseudo random number generators. Once the random bits are generated from a quantum process, bit-streams are written in a computationally similar language in which the algorithmic methods are written. We discuss the typical methods of extracting randomness from weak sources of randomness i.e. Toeplitz hash functions. We further discuss entropy as a measure of randomness and discuss it in full generalization where smooth conditional min-entropy, often used as a conservative measure in QRNGs, bounds the quantum relative Rényi entropy with IID sources. Towards the end, we discuss the internal working of CS-PRNGs (Chacha20) and LFSR based random numbers used in quantum cryptography via low-cost FPGA as a cheaper alternative to QRNGs.

Chapter 4 discusses the first testing front for QRNGs, namely the quantum correlation front. Out of the two quantum entangled qubits, the correlated bits are used for the generation of random numbers while the relative phase information between the qubits is used to evade the need of putting trust in the generation devices. This is reflected in the form of the Hong-Ou-Mandel curve on the source front while it is reflected in Bell inequality violation (combined with density matrix and estimation theory) on the detector front. This advantage of evading the need to trust the devices is an unparalleled strength of QRNGs over any other generation method in the realm of privacy.

Chapter 5 discusses the need for QRNGs from a user perspective. If the final bit-stream before putting to any use, is in the form of a .txt file or as a beacon service, how does it matter whether the bits generated come from quantum hardware or computer software? Is there still a quantum signature present in the post-processed QRNGs? We test this strength, if any, of QRNGs on the computational front with four different measures quantifying randomness based on the output bit-stream. These measures include NIST-STS, machine learning models, measures of incomputability like Kolmogorov complexity, and Borel normality. The chapter concludes by strengthening the no-go theorem for QRNGs as proposed by [105] and claims that there is no differentiator between QRNGs and PRNGs once they are post-processed. This holds for good and bad PRNGs both. In addition, as a side result, we also come to an understanding that machine learning algorithms might be able to differentiate unprocessed bit-streams coming out of PRNGs and QRNGs as the former are computationally not too complex (which gives rise to computational unpredictability) compared to the testing machine learning model. Thus, we can say that based on the post-processed output bit-stream, computational methods of generation are good enough compared to quantum generation methods. There is a distinction between unprocessed bit-streams. However, quantum generation methods cannot eliminate biases from the experiment. Thus, they always need post-processing. If one is unable to differentiate such methods from computational methods of generation, the randomness beacon service of QRNGs appears not worth investigating.

Chapter 6 discusses another class of quantum random number generators, namely noise based quantum random number generators, to test the strength of quantum random numbers on grounds of quantum origin front. These quantum origin sources generate random numbers equivalent to computational methods and thus don't show any discernible advantage on the computational front. However, on the unpredictability front, they provide 3.5 times more security compared to optical TRNGs (LED based). This relative advantage comes at the cost of expensive LASER compared to LED. Thus QRNGs based on quantum origin sources also provide an advantage. However, this advantage can be compensated by using a 3.5 times longer string of TRNG for encryption.

After discussing the different strengths of random number generators, it becomes obvious that quantum correlations have an advantage that can be used to prove device independence. Computational methods cannot differentiate a post-processed bit-stream from whether it arose from quantum hardware or computer software. This makes the randomness beacon a redundant service. Quantum origin based QRNGs cannot prove their advantage against computational methods. Although extremely fast, they could not show a significant advantage over other electronic noise based random numbers other than being more precisely calibratable.

One very interesting convergence of thoughts is observed. From an algorithmic information theory point of view, the strongest definition as proposed by Hertling and Weihrauch [118] is "A randomness space is a triple (X, B, μ), where X is a topological space, B, a map from **N** to the powerset of X, is a total numbering of subspaces of the topology of X, and μ is a measure defined on the σ-algebra generated by the topology of X." For quantum random number generators, this is consistent, say, starting with polarising beam-splitter based random number generators. The randomness space directly translates in the quantum domain where the σ-algebra is Lie algebra, defining the topological space X, μ is the commutation relation between the orthonormal basis of polarisation which is used as a measure for randomness. Also, we have restricted ourselves to one of the mappings B which translates to commutation relations for polarisation. We can have more mappings in B defining continuous variable or entanglement based QRNGs. This convergence of two independent approaches from algorithmic information theory with quantum random number generators is a good sign amongst the common consensus of the advantage of quantum random number generators. .

7.2 Scope for Future Work

The scope of this direction of research is plentiful. As an example, the comparison between PRNGs and quantum origin based random numbers is still unexplored. Both are comparably fast, but is there an advantage of shot noise based QRNGs to be used in IoT applications providing better privacy and security to users? This can be tested by using advanced machine learning algorithms against these sources. Another point to investigate could be the role of classical processing done by ADC and other components in generating patterns in the bit-stream.

On the computational front, a better understanding of randomness is required. Different post-processing methods need to be looked into as Toeplitz and Trevisan extractors remain the major players in the post-processing domain even after 10 years of research. As the number of parameters in the machine learning community has already risen to billions, it is just a matter of time that someone deploys such models to reverse engineer post-processing algorithms of QRNGs and make them pointless. Today, non-malleable extractors [119] which could be tweaked for post-processing applications are a good bet.

Until now, only known resources like quantum entanglement, density matrices and uncertainty relations have been connected to entropy formalism for randomness characterisation. This list needs to be more inclusive of other quantum optical measures such as second-order intensity correlation. Other than entropy methods, statistical methods to decipher the quality of random numbers generated from different sources appear a dead end. The quantum mechanics part of QRNGs is completely independent of the computational front, as hinted by measures like smooth conditional min-entropy (based on density matrix). The next step is the chip implementation of QRNGs [120] so that they can directly be used in quantum cryptographic applications or IoT applications.

Another possible direction of research is in Optical Neural Networks [121]. Since neural networks often require different combinations of samples while training (which are often picked randomly), quantum random numbers could provide a method to choose these samples randomly and provide a direct optical source of randomness in optical experiments.

Quantum random number generators form the fundamental understanding of photonic quantum computers, showing advantages either through the Ising model or through Gaussian Boson Sampling algorithms. These algorithms, built on quantum randomness, provide quantum advantage in solving graph problems through graph isomorphism. Since Ising models can easily be implemented on SLMs, quantum algorithms providing advantage in shortest vector problem algorithms can be implemented on SLMs in bulk optics. The list of problems that can be solved either via the Ising model or Gaussian Boson Sampling is ever increasing as an avalanche. Quantum random number generators provide a fundamental understanding of solving these computational problems using quantum algorithms.

In another problem, Toeplitz hash function is implemented on FPGA using Fourier transform methods for faster and longer block length processing. It would be interesting to see an analogous optical implementation using methods from optical computing. This could be a good technique to have optical one-way hash functions for encryption in the future.

Publications related to this Thesis

Published

1. **Vardaan Mongia**, Abhishek Kumar, Shashi Prabhakar, Anindya Banerji, Ravindra Pratap Singh, “Device-Independent Quantum Random Number Generation”, Physics Letters A, Volume 526, 28 November 2024, 129954.
   DOI: https://doi.org/10.1016/j.physleta.2024.129954

Under Review

2. Pooja Chandravanshi, Jaya Krishna Meka, **Vardaan Mongia**, Ravindra Pratap Singh and Shashi Prabhakar, “ LFSR based RNG on low cost FPGA for QKD applications.”
   DOI: https://doi.org/10.48550/arXiv.2307.16431

Under preparation

3. **Vardaan Mongia**, Abhishek Kumar, Shashi Prabhakar, Ravindra Pratap Singh, “Strengthening the no-go theorem for QRNGs”.

4. **Vardaan Mongia**, Pooja Chandravanshi, Jayanth Ramakrishnan, Shashi Prabhakar, Ravindra Pratap Singh, “Questioning the “quantum” advantage in shot-noise based QRNGs”.

Other Publications

5. Sarika Mishra, Ayan Biswas, Satyajeet Patil, Pooja Chandravanshi, **Vardaan Mongia**, Tanya Sharma, Anju Rani, Shashi Prabhakar, S Ramachandran, and Ravindra Pratap Singh, “BBM92 quantum key distribution over a free space dusty channel of 200 meters”, Journal of Optics **24**, 074002 (2022).
   DOI: https://doi.org/10.1088/2040-8986/ac6f0b

6. Anju Rani, **Vardaan Mongia**, Parvatesh Parvatikar, Rutuj Gharate, Jayanth R, Pooja Chandravanshi, Tanya Sharma, and Ravindra Pratap Singh “Passive QRNG-based BB84 protocol using heralded single photon source”. (Under Review)

References

[1] Joakim Argillander, Alvaro Alarcón, Chunxiong Bao, Chaoyang Kuang, Gustavo Lima, Feng Gao, and Guilherme B Xavier. Quantum random number generation based on a perovskite light emitting diode. *Communications Physics*, 6(1):157, 2023.

[2] Ali Anwar, Chithrabhanu Perumangatt, Fabian Steinlechner, Thomas Jennewein, and Alexander Ling. Entangled photon-pair sources based on three-wave mixing in bulk crystals. *Review of Scientific Instruments*, 92(4), 2021.

[3] Patrick J Coles, Mario Berta, Marco Tomamichel, and Stephanie Wehner. Entropic uncertainty relations and their applications. *Reviews of Modern Physics*, 89(1):015002, 2017.

[4] Francis Galton. Dice for statistical experiments. *Nature*, 42(1070):13–14, 1890.

[5] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*. SIAM, 1992.

[6] SarahButcher. What to expect from Morgan Stanley's psychometric tests. `https://capd.mit.edu/blog/2021/09/30/what-to-expect-from-morgan-stanleys-psychometric-tests/`. [Online; accessed 24-January-2022].

[7] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 21–30, 2007.

[8] Tatsuaki Okamoto and Kazuo Ohta. How to utilize the randomness of zero-knowledge proofs. In *Conference on the Theory and Application of Cryptography*, pages 456–475. Springer, 1990.

[9] Claude E Shannon. Communication theory of secrecy systems. *The Bell system technical journal*, 28:656–715, 1949.

[10] Bernice Brown. Some tests of the randomness of a million digits. *RAND Corporation, Research PaperP-44*, 1948.

[11] Jonghyun Kim and Hyungil Chae. A 10-gbps, 0.121-pj/bit, all-digital true random-number generator using middle square method. In *2022 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, pages 1–3. IEEE, 2022.

[12] George Marsaglia. The structure of linear congruential sequences. In *Applications of number theory to numerical analysis*, pages 249–285. Elsevier, 1972.

[13] Phu Nguyen Phan Hai, Hoa Nguyen Hong, Bao Bui Quoc, and Trang Hoang. A comparative research on vpn technologies on operating system for routers. In *2021 International Conference on Advanced Technologies for Communications (ATC)*, pages 89–93. IEEE, 2021.

[14] Antoine Delignat-Lavaud, Cédric Fournet, Markulf Kohlweiss, Jonathan Protzenko, Aseem Rastogi, Nikhil Swamy, Santiago Zanella-Béguelin, Karthikeyan Bhargavan, Jianyang Pan, and Jean Karim Zinzindohoue. Implementing and proving the tls 1.3 record layer. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 463–482. IEEE, 2017.

[15] TN Palmer and G Seregin. The real butterfly effect. *Nonlinearity*, 27(9):R123, 2014.

[16] M Alvaro, M Carretero, and LL Bonilla. Noise-enhanced spontaneous chaos in semiconductor superlattices at room temperature. *Europhysics Letters*, 107(3):37002, 2014.

[17] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999.

[18] Thomas Jennewein, Ulrich Achleitner, Gregor Weihs, Harald Weinfurter, and Anton Zeilinger. A fast and compact quantum random number generator. *Review of Scientific Instruments*, 71(4):1675–1680, 2000.

[19] Matthias Leifgen, Tim Schröder, Friedemann Gädeke, Robert Riemann, Valentin Métillon, Elke Neu, Christian Hepp, Carsten Arend, Christoph Becher, Kristian Lauritsen, et al. Evaluation of nitrogen-and silicon-vacancy defect centres as single photon sources in quantum key distribution. *New journal of physics*, 16(2):023021, 2014.

[20] Xuyang Wang, Tao Zheng, Yanxiang Jia, Qianru Zhao, Yu Zhang, Yuqi Shi, Ning Wang, Zhenguo Lu, Jun Zou, and Yongmin Li. Compact quantum random number generator based on a laser diode and silicon photonics integrated hybrid chip. *arXiv preprint arXiv:2401.11099*, 2024.

[21] Stefano Pironio, Antonio Acín, Serge Massar, A Boyer de La Giroday, Dzmitry N Matsukevich, Peter Maunz, Steven Olmschenk, David Hayes, Le Luo,

T Andrew Manning, et al. Random numbers certified by bell's theorem. *Nature*, 464(7291):1021–1024, 2010.

[22] Ben Quinn and Charles Arthur. Playstation network hackers access data of 77 million users. *The Guardian*, 27, 2011.

[23] Suné Von Solms and Renier Van Heerden. The consequences of edward snowden nsa related information disclosures. In *ICCWS 2015—The Proceedings of the 10th International Conference on Cyber Warfare and Security: ICCWS2015*, page 358, 2015.

[24] Sandeep Rao and Corbet Shaen. Mt. gox–the fall of a giant. *Understanding Cryptocurrency Fraud, edited by Shaen Corbet*, pages 71–82, 2022.

[25] Tommaso Lunghi, Jonatan Bohr Brask, Charles Ci Wen Lim, Quentin Lavigne, Joseph Bowles, Anthony Martin, Hugo Zbinden, and Nicolas Brunner. Self-testing quantum random number generator. *Phys. Rev. Lett.*, 114:150501, Apr 2015.

[26] Harshank Shrotriya, Kishor Bharti, and Leong-Chuan Kwek. Robust semi-device-independent certification of all pure bipartite maximally entangled states via quantum steering. *Physical Review Research*, 3(3):033093, 2021.

[27] Carlos Abellan. *Quantum Random Number Generators for Industrial Applications.* Phd thesis, ICFO, Barcelona, Spain, June 19 2018.

[28] Christopher C Gerry and Peter L Knight. *Introductory quantum optics.* Cambridge university press, 2023.

[29] M. D. Reid and D. F. Walls. Violations of classical inequalities in quantum optics. *Phys. Rev. A*, 34:1260–1276, Aug 1986.

[30] Robert W Boyd, Alexander L Gaeta, and Enno Giese. Nonlinear optics. In *Springer Handbook of Atomic, Molecular, and Optical Physics*, pages 1097–1110. Springer, 2008.

[31] Lev Davidovich Landau and Evgenii Mikhailovich Lifshitz. *Quantum mechanics: non-relativistic theory*, volume 3. Elsevier, 2013.

[32] Giuliana Di Martino, Yannick Sonnefraud, Mark S Tame, Stéphane Kéna-Cohen, F Dieleman, ŞK Özdemir, MS Kim, and Stefan A Maier. Observation of quantum interference in the plasmonic hong-ou-mandel effect. *Physical Review Applied*, 1(3):034004, 2014.

[33] Jonathan P Dowling, James D Franson, Hwang Lee, and Gerard J Milburn. Towards scalable linear-optical quantum computers. *Experimental Aspects of Quantum Computing*, pages 205–213, 2005.

[34] Henry Semenenko, Philip Sibson, Mark G Thompson, and Chris Erven. Interference between independent photonic integrated devices for quantum key distribution. *Optics letters*, 44(2):275–278, 2019.

[35] Philip Georgi, Marcello Massaro, Kai-Hong Luo, Basudeb Sain, Nicola Montaut, Harald Herrmann, Thomas Weiss, Guixin Li, Christine Silberhorn, and Thomas Zentgraf. Metasurface interferometry toward quantum sensors. *Light: Science & Applications*, 8(1):70, 2019.

[36] SP Walborn, AN De Oliveira, S Pádua, and CH Monken. Multimode hong-ou-mandel interference. *Physical review letters*, 90(14):143601, 2003.

[37] E Schr et al. Die gegenw rtige situation in der quantenmechanik. *Die Naturwissenschaften*, 23(48):807–812, 1935.

[38] John S Bell. On the einstein podolsky rosen paradox. *Physics Physique Fizika*, 1(3):195, 1964.

[39] Dietrich Dehlinger and MW Mitchell. Entangled photons, nonlocality, and bell inequalities in the undergraduate laboratory. *American Journal of Physics*, 70(9):903–910, 2002.

[40] John F Clauser and Abner Shimony. Bell's theorem. experimental tests and implications. *Reports on Progress in Physics*, 41(12):1881, 1978.

[41] Alain Aspect, Jean Dalibard, and Gérard Roger. Experimental test of bell's inequalities using time-varying analyzers. *Physical review letters*, 49(25):1804, 1982.

[42] Lynden K. Shalm, Evan Meyer-Scott, Bradley G. Christensen, Peter Bierhorst, Michael A. Wayne, Martin J. Stevens, Thomas Gerrits, Scott Glancy, Deny R. Hamel, Michael S. Allman, Kevin J. Coakley, Shellee D. Dyer, Carson Hodge, Adriana E. Lita, Varun B. Verma, Camilla Lambrocco, Edward Tortorici, Alan L. Migdall, Yanbao Zhang, Daniel R. Kumor, William H. Farr, Francesco Marsili, Matthew D. Shaw, Jeffrey A. Stern, Carlos Abellán, Waldimar Amaya, Valerio Pruneri, Thomas Jennewein, Morgan W. Mitchell, Paul G. Kwiat, Joshua C. Bienfang, Richard P. Mirin, Emanuel Knill, and Sae Woo Nam. Strong loophole-free test of local realism. *Phys. Rev. Lett.*, 115:250402, Dec 2015.

[43] Mark Um, Xiang Zhang, Junhua Zhang, Ye Wang, Shen Yangchao, D-L Deng, Lu-Ming Duan, and Kihwan Kim. Experimental certification of random numbers via quantum contextuality. *Scientific reports*, 3(1):1627, 2013.

[44] A Ahmad and A Al Maashri. On sequence lengths of some special external exclusive or type lfsr structures–study and analysis. *The Journal of Engineering Research [TJER]*, 11(2):1–14, 2014.

[45] KN Devika and Ramesh Bhakthavatchalu. Design of reconfigurable LFSR for VLSI IC testing in ASIC and FPGA. In *2017 international conference on communication and signal processing (ICCSP)*, pages 0928–0932. IEEE, 2017.

[46] Amirhossein Ebrahimzadeh, Abolfazl Falahati, et al. Frequency hopping spread spectrum security improvement with encrypted spreading codes in a partial band noise jamming environment. *Journal of Information Security*, 4:1–6, 2013.

[47] Gattineni Madhupavani and Kadiyam Suneetha. Design of random number generation using 256 Bit LFSR in FPGA. *International Journal of Advanced Technology and Innovative Research*, 9, 2017.

[48] Jyotsna Kumar Mandal, Suresh Chandra Satapathy, Manas Kumar Sanyal, and Vikrant Bhateja. *Proceedings of the First International Conference on Intelligent Computing and Communication*, volume 458. Springer, 2016.

[49] Lenore Blum, Manuel Blum, and Mike Shub. A simple unpredictable pseudo-random number generator. *SIAM Journal on computing*, 15(2):364–383, 1986.

[50] Jaime Gutierrez. Attacking the linear congruential generator on elliptic curves via lattice techniques. *Cryptography and Communications*, 14(3):505–525, 2022.

[51] Daniel J Bernstein et al. Chacha, a variant of salsa20. In *Workshop record of SASC*, volume 8, pages 3–5. Citeseer, 2008.

[52] Fabrizio De Santis, Andreas Schauer, and Georg Sigl. Chacha20-poly1305 authenticated encryption for high-speed embedded iot applications. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, pages 692–697. IEEE, 2017.

[53] Yoav Nir and Adam Langley. Chacha20 and poly1305 for ietf protocols. Technical report, 2018.

[54] Mark Wilde. On the quantum renyi relative entropies and their use. QuICS Seminar, 2020.

[55] Milán Mosonyi and Tomohiro Ogawa. Quantum hypothesis testing and the operational interpretation of the quantum rényi relative entropies. *Communications in Mathematical Physics*, 334:1617–1648, 2015.

[56] Sumeet Khatri and Mark M Wilde. Principles of quantum communication theory: A modern approach. *arXiv preprint arXiv:2011.04672*, 2020.

[57] Kun Fang and Hamza Fawzi. Geometric rényi divergence and its applications in quantum channel capacities. *Communications in Mathematical Physics*, 384(3):1615–1677, 2021.

[58] Miklos Santha and Umesh V Vazirani. Generating quasi-random sequences from semi-random sources. *Journal of computer and system sciences*, 33(1):75–87, 1986.

[59] John Von Neumann et al. Various techniques used in connection with random digits. *Applied Math Series*, 12(36-38):1, 1951.

[60] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Cambridge university press, 1995.

[61] Umesh V Vazirani. Generating quasi-random sequences from slightly-random sources.

[62] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, et al. *A statistical test suite for random and pseudorandom number generators for cryptographic applications*, volume 22. US Department of Commerce, Technology Administration, National Institute of . . . , 2001.

[63] Sheldon M Ross. *Introduction to probability models*. Academic press, 2014.

[64] Robert G Brown, Dirk Eddelbuettel, and David Bauer. Dieharder. *Duke University Physics Department Durham, NC*, pages 27708–0305, 2018.

[65] G Marsaglia. Random number cdrom including the diehard battery of tests of randomness. *http://stat. fsu. edu/pub/diehard/cdrom*, 1995.

[66] M Émile Borel. Les probabilités dénombrables et leurs applications arithmétiques. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 27(1):247–271, 1909.

[67] AMD. Artix 7 35T Arty FPGA Evaluation Kit, 2015.

[68] Jayakrishna Meka and Pooja Chandravanshi. LFSR based RNG for QKD application, 2023.

[69] Ayan Biswas, Anindya Banerji, Pooja Chandravanshi, Rupesh Kumar, and Ravindra P. Singh. Experimental side analysis of bb84 qkd source. *IEEE Journal of Quantum Electronics*, 57:1–7, 2021.

[70] Peter Kietzmann, Thomas C Schmidt, and Matthias Wählisch. A guideline on pseudorandom number generation (prng) in the iot. *ACM Computing Surveys (CSUR)*, 54(6):1–38, 2021.

[71] Lih-Yuan Deng and Dale Bowman. Developments in pseudo-random number generators. *Wiley Interdisciplinary Reviews: Computational Statistics*, 9(5):e1404, 2017.

[72] Yi Jian, Min Ren, E Wu, Guang Wu, and Heping Zeng. Two-bit quantum random number generator based on photon-number-resolving detection. *Review of Scientific Instruments*, 82(7), 2011.

[73] Yong Shen, Liang Tian, and Hongxin Zou. Practical quantum random number generator based on measuring the shot noise of vacuum states. *Physical Review A*, 81(6):063814, 2010.

[74] Lawrence E Bassham III, Andrew L Rukhin, Juan Soto, James R Nechvatal, Miles E Smid, Elaine B Barker, Stefan D Leigh, Mark Levenson, Mark Vangel, David L Banks, et al. *SP 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications.* National Institute of Standards & Technology, 2010.

[75] Gene Novark and Emery D Berger. Dieharder: securing the heap. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 573–584, 2010.

[76] John Walker. Ent: a pseudorandom number sequence test program. *Software and documentation available at/www. fourmilab. ch/random/S*, 2008.

[77] Barbara M. Terhal. Bell inequalities and the separability criterion. *Physics Letters A*, 271(5):319–326, 2000.

[78] Sonia Mazzucchi, Nicolò Leone, Stefano Azzini, Lorenzo Pavesi, and Valter Moretti. Entropy certification of a realistic quantum random-number generator based on single-particle entanglement. *Phys. Rev. A*, 104:022416, Aug 2021.

[79] Rutvij Bhavsar, Sammy Ragy, and Roger Colbeck. Improved device-independent randomness expansion rates using two sided randomness. *New Journal of Physics*, 25(9):093035, 2023.

[80] Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. Efficient certifiable randomness from a single quantum device. 2022.

[81] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press, 2010.

[82] Xiongfeng Ma, Feihu Xu, He Xu, Xiaoqing Tan, Bing Qi, and Hoi-Kwong Lo. Postprocessing for quantum random-number generators: Entropy evaluation and randomness extraction. *Physical Review A*, 87(6):062327, 2013.

[83] William K. Wootters and W. S. Leng. Quantum entanglement as a quantifiable resource [and discussion]. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 356(1743):1717–1731, 1998.

[84] Michał Horodecki, Paweł Horodecki, and Ryszard Horodecki. Separability of mixed states: necessary and sufficient conditions. *Physics Letters A*, 223(1):1–8, 1996.

[85] Pawel Horodecki. Separability criterion and inseparable mixed states with positive partial transposition. *Physics Letters A*, 232(5):333–339, 1997.

[86] M. Lewenstein, B. Kraus, J. I. Cirac, and P. Horodecki. Optimization of entanglement witnesses. *Phys. Rev. A*, 62:052310, Oct 2000.

[87] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23:880–884, Oct 1969.

[88] Roman Schmied. Quantum state tomography of a single qubit: comparison of methods. *Journal of Modern Optics*, 63(18):1744–1758, 2016.

[89] Beth Schaefer, Edward Collett, Robert Smyth, Daniel Barrett, and Beth Fraher. Measuring the stokes polarization parameters. *American Journal of Physics*, 75(2):163–168, 2007.

[90] Daniel F. V. James, Paul G. Kwiat, William J. Munro, and Andrew G. White. Measurement of qubits. *Phys. Rev. A*, 64:052312, Oct 2001.

[91] Joseph M Lukens, Kody J H Law, Ajay Jasra, and Pavel Lougovski. A practical and efficient approach for bayesian quantum state estimation. *New Journal of Physics*, 22(6):063038, jun 2020.

[92] Ryszard Horodecki, Pawel Horodecki, and Michal Horodecki. Violating bell inequality by mixed spin-12 states: necessary and sufficient condition. *Physics Letters A*, 200(5):340–344, 1995.

[93] C. K. Hong, Z. Y. Ou, and L. Mandel. Measurement of subpicosecond time intervals between two photons by interference. *Phys. Rev. Lett.*, 59:2044–2046, Nov 1987.

[94] Robert Konig, Renato Renner, and Christian Schaffner. The operational meaning of min-and max-entropy. *IEEE Transactions on Information theory*, 55(9):4337–4347, 2009.

[95] Agata M Brańczyk. Hong-ou-mandel interference. *arXiv preprint arXiv:1711.00080*, 2017.

[96] Pierre L'Ecuyer and Richard Simard. Testu01: A c library for empirical testing of random number generators. *ACM Trans. Math. Softw.*, 33(4), August 2007.

[97] Robert G Brown. Dieharder, 2004.

[98] John Kelsey, Bruce Schneier, David Wagner, and Chris Hall. Cryptanalytic attacks on pseudorandom number generators. In *International workshop on fast software encryption*, pages 168–188. Springer, 1998.

[99] Anil K Jain, Robert P. W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1):4–37, 2000.

[100] Arvind Narayanan. Understanding social media recommendation algorithms. 2023.

[101] Jing Yang, Shuangyi Zhu, Tianyu Chen, Yuan Ma, Na Lv, and Jingqiang Lin. Neural network based min-entropy estimation for random number generators. In *International Conference on Security and Privacy in Communication Systems*, pages 231–250. Springer, 2018.

[102] Robert Konig, Renato Renner, and Christian Schaffner. The operational meaning of min- and max-entropy. *IEEE Transactions on Information Theory*, 55(9):4337–4347, 2009.

[103] Masahito Hayashi and Toyohiro Tsurumaru. More efficient privacy amplification with less random seeds via dual universal hash function. *IEEE Transactions on Information Theory*, 62(4):2213–2232, 2016.

[104] Cristian S. Calude, Michael J. Dinneen, Monica Dumitrescu, and Karl Svozil. Experimental evidence of quantum randomness incomputability. *Phys. Rev. A*, 82:022102, Aug 2010.

[105] Toyohiro Tsurumaru, Tsubasa Ichikawa, Yosuke Takubo, Toshihiko Sasaki, Jaeha Lee, and Izumi Tsutsui. Indistinguishability between quantum randomness and pseudorandomness under efficiently calculable randomness measures. *Phys. Rev. A*, 109:022243, Feb 2024.

[106] Ding-Xuan Zhou. Theory of deep convolutional neural networks: Downsampling. *Neural Networks*, 124:319–327, 2020.

[107] Barak Hadad, Sahar Froim, Erez Yosef, Raja Giryes, and Alon Bahabad. Deep learning in optics—a tutorial. *Journal of Optics*, 25(12):123501, nov 2023.

[108] Hu Yuanfu and Wu Xunsen. The methods of improving the compression ratio of lz77 family data compression algorithms. In *Proceedings of Third International*

Conference on Signal Processing (ICSP'96), volume 1, pages 698–701 vol.1, 1996.

[109] Ming Li and Paul Vitányi. Two decades of applied kolmogorov complexity. 1988.

[110] Crist ian Caludet. Borel normality and algorithmic randomness. In *Developments in Language Theory*, page 113. World Scientific, 1993.

[111] Cai Li, Jianguo Zhang, Luxiao Sang, Lishuang Gong, Longsheng Wang, Anbang Wang, and Yuncai Wang. Deep learning-based security verification for a random number generator using white chaos. *Entropy*, 22(10):1134, 2020.

[112] Lukas Osswald, Marco Haeberle, and Michael Menth. Performance comparison of vpn solutions. 2020.

[113] Zhenwei Li, Bao Feng, Liangjie Cui, Hui Wang, Yuxiang Bian, Genle Piao, and Xingyu Zhou. Quantify randomness of quantum random number with transformer network. In *2023 3rd International Conference on Intelligent Power and Systems (ICIPS)*, pages 17–22, 2023.

[114] Johannes Thewes, Carolin Lüders, and Marc Aßmann. Eavesdropping attack on a trusted continuous-variable quantum random-number generator. *Physical Review A*, 100(5):052318, 2019.

[115] Ulf Leonhardt. *Measuring the quantum state of light*, volume 22. Cambridge university press, 1997.

[116] Thomas M Cover and Joy A Thomas. Information theory and statistics. *Elements of information theory*, 1(1):279–335, 1991.

[117] Meltem Sonmez Turan, Elaine Barker, John Kelsey, K McKay, M Baish, and Mike Boyle. Nist, sp 800–90b: Recommendation for the entropy sources used for random bit generation. *Tech. Rep*, 2018.

[118] Peter Hertling and Klaus Weihrauch. Randomness spaces. In *International Colloquium on Automata, Languages, and Programming*, pages 796–807. Springer, 1998.

[119] Eshan Chattopadhyay and Xin Li. Explicit non-malleable extractors, multi-source extractors, and almost optimal privacy amplification protocols. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 158–167. IEEE, 2016.

[120] Nicolò Leone, Davide Rusca, Stefano Azzini, Giorgio Fontana, Fabio Acerbi, Alberto Gola, Alessandro Tontini, Nicola Massari, Hugo Zbinden, and Lorenzo

Pavesi. An optical chip for self-testing quantum random number generation. *APL Photonics*, 5(10), 2020.

[121] Tingzhao Fu, Jianfa Zhang, Run Sun, Yuyao Huang, Wei Xu, Sigang Yang, Zhihong Zhu, and Hongwei Chen. Optical neural networks: progress and challenges. *Light: Science & Applications*, 13(1):263, 2024.