information

MDPI

Review

Quantum Randomness in Cryptography—A Survey of Cryptosystems, RNG-Based Ciphers, and QRNGs

Anish Saini *, Athanasios Tsokanos and Raimund Kirner

School of Physics, Engineering and Computer Science, University of Hertfordshire, Hatfield AL10 9AB, UK; a.tsokanos@herts.ac.uk (A.T.); r.kirner@herts.ac.uk (R.K.)
* Correspondence: a.saini@herts.ac.uk

Abstract: Cryptography is the study and practice of secure communication with digital data and focuses on confidentiality, integrity, and authentication. Random number generators (RNGs) generate random numbers to enhance security. Even though the cryptographic algorithms are public and their strength depends on the keys, cryptoanalysis of encrypted ciphers can significantly contribute to the unveiling of the cipher's key. Therefore, to ensure high data security over a network, researchers need to improve the randomness of keys as they develop cryptosystems. Quantum particles have a leading edge in advancing RNG technology as they can provide true randomness, unlike pseudo-random numbers generators (PRNGs). In order to increase the level of the security of cryptographic systems based on random numbers, this survey focuses on three objectives: Cryptosystems with related cryptographic attacks, RNG-based cryptosystems, and the design of quantum random number generators (QRNGs). This survey aims to provide researchers with information about the importance of RNG-based ciphers and various research techniques for QRNGs that can incorporate quantum-based true randomness in cryptosystems.

Keywords: cryptosystems; cryptoanalysis; RNG-based cipher; QRNG

check for updates

Citation: Saini, A.; Tsokanos, A.; Kirner, R. Quantum Randomness in Cryptography—A Survey of Cryptosystems, RNG-Based Ciphers, and QRNGs. *Information* **2022**, *13*, 358. https://doi.org/10.3390/info13080358

Academic Editor: Willy Susilo

Received: 10 May 2022
Accepted: 6 July 2022
Published: 27 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

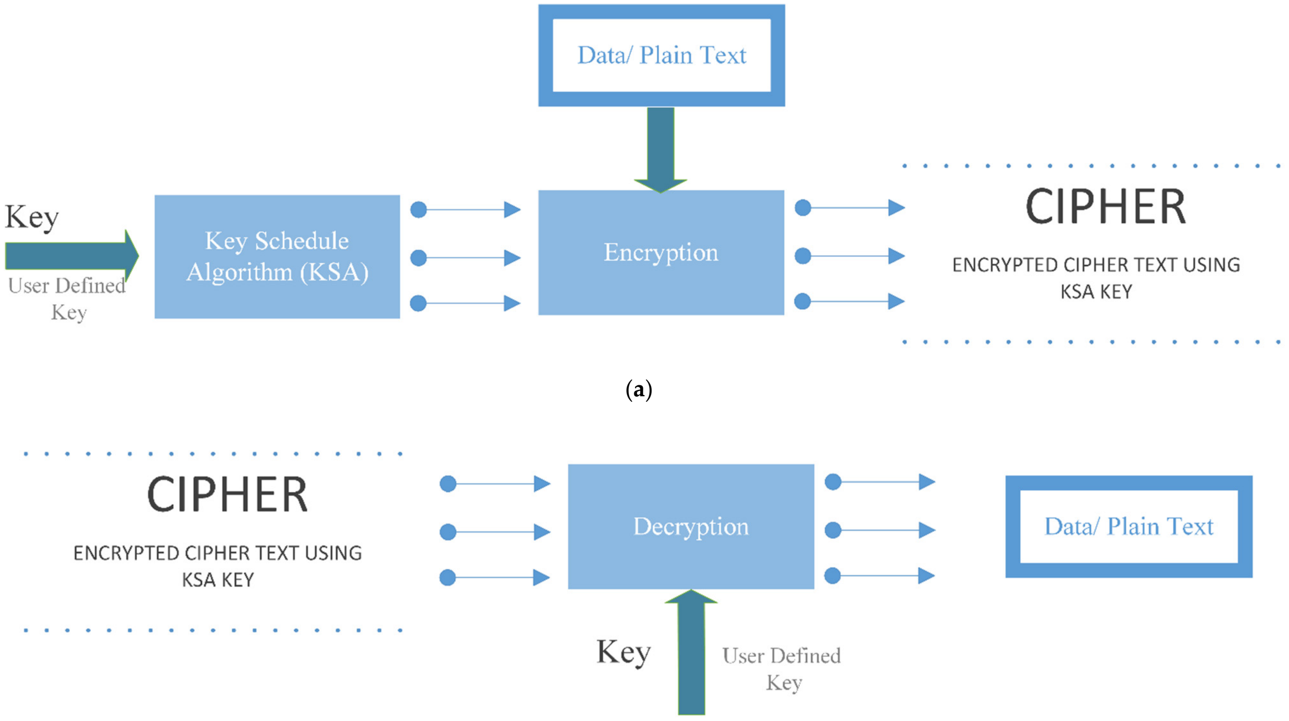
Information security is the principal concern whenever data is transmitted over a network, a confined physical space of connected digital equipment, or a public network, such as the Internet. The information is considered secure if it cannot be understood by someone other than the intended recipient. The unintended person trying to steal the information is referred to as a hacker. In cryptography [1], one part of cryptology, such information is termed encrypted data, and the cryptosystem is designed with the primary concern of confidentiality and a security measure to safeguard the information from unauthorized access. The cryptosystem includes three significant processes: the Key Schedule Algorithm (KSA), the Encryption Algorithm, and the Decryption Algorithm.

The KSA is a process of generating keys for encryption and decryption. It takes the user-defined or original key as a set of bits, such as 40, 128, 192, 256, or more significant bits, to expand them based on its processing steps or the number of rounds designed for the encryption algorithm. The purpose of the KSA is to make the key so strong that it is not vulnerable to attack, and hackers cannot find the original key.

The Encryption Algorithm (EA) encrypts data by using a key and converts it to an unreadable format, and this process is called encryption. On the other hand, a Decryption Algorithm (DA) involves using the same or different key for decoding the cipher and converting the cipher back into the original data, which is called decryption. Figure 1a,b shows the three processes.

Encryption transforms plaintext into the ciphertext and secure ciphers by associating various cryptographic properties such as nonlinearity, propagation criteria, correlation, and algebraic immunity. However, the security of a cipher depends on how vulnerable the used key is to a cryptanalysis attack [2]. In addition to KSA, which divides the key into subkeys, random numbers can be used to make the key more robust and complex.

Figure 2 shows the KSA and encryption of a cryptosystem with a random number generator (RNG). RNGs provide the random numbers or bits used in achieving randomness in a cryptosystem. Encryption encrypts the plain text into ciphertext with advancement in an RNG-based cryptosystem; the KSA key also incorporates the random number bits generated by an RNG.



(**a**)

(**b**)

Figure 1. (**a**) Key Schedule Algorithm (KSA) and encryption of a cryptosystem; (**b**) decryption of a cryptosystem.

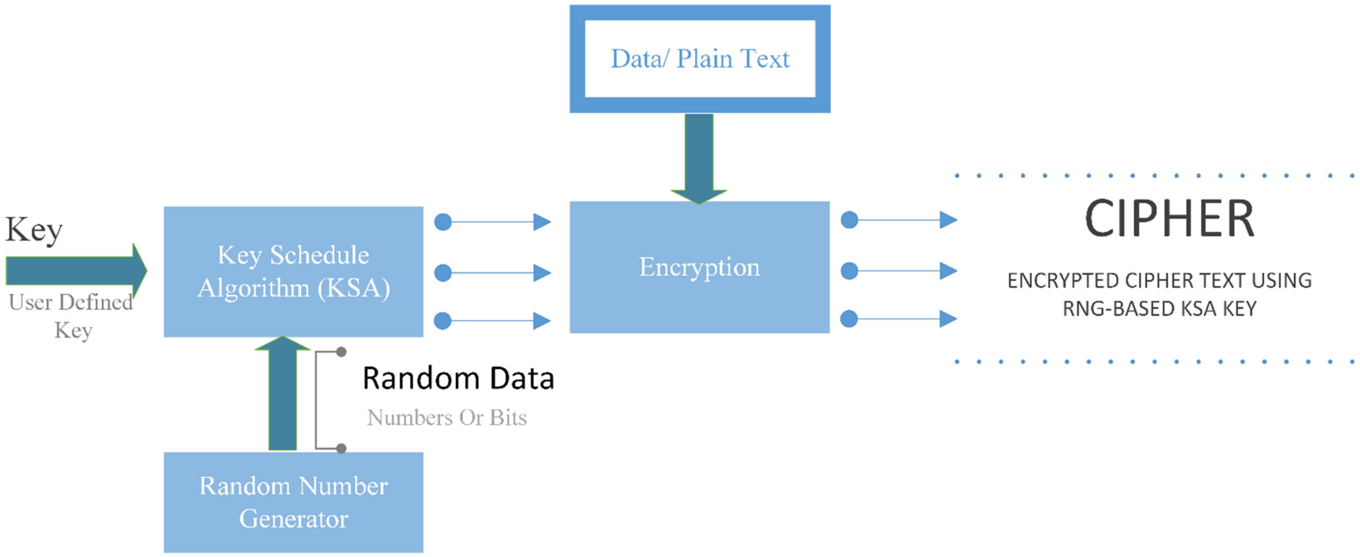


Figure 2. Key Schedule Algorithm (KSA) and encryption of an RNG-based cryptosystem.

RNGs' randomness critically depends on the type of RNG.

Pseudorandom number generators (PRNGs) [3,4] are based on algorithms for generating seemingly random numbers, which are determined by their seed, or initial value to enhance security. Shobhit Sinha et al. [5] compared various PRNGs based on their statistical randomness and cryptographic security, concluding that some PRNGs performed significantly worse than others.

Hardware and portable USB-based true random number generators (TRNGs) generate a truly random number based on digital noise intervention. The entropy or randomness source for TRNGs is usually electrical noise and an unpredictable component of electronic systems. Lishuang et al. [6] compared the advantages and challenges of entropy sources that use electrical noise.

True (non-pseudo) random data [7,8] can also be achieved by using a quantum random number generator (QRNG) based on quantum particles. Random number generators with quantum [9] randomness are superior to pseudo or true RNGs, as their source of randomness is invulnerable to environmental disturbance, such as temperature, voltage, or current and provides the highest level of entropy [10].

In this survey, we discuss and categorise various studies on cryptography and RNGs. In the first category, we study cryptosystems and cryptographic cryptoanalysis of the ciphers. Our second category focuses on different RNG-based cryptosystems in light of the RNGs' robustness. This category results help analyze the enhancement in the security of a cryptosystem. We will discuss in the last category the multiple features of QRNGs, analyzing how the QRNGs differ from each other and which of the QRNG features are best suited for any given application.

Lastly, this paper discusses and proposes open research problems for a cipher function, which relates to the randomness of quantum random numbers (QRN).

This research aims to contribute to the future of cryptography and to become a part of the open quantum-safe project [11]. The survey is organised as follows: Section II provides a survey methodology, categorised into three subsections. Section III focuses on open research problems of ciphers based on QRNGs. The paper is concluded in Section IV.

2. Survey Methodology

The survey methodology is subdivided into three categories:

A. Category I: Cryptosystems and cryptographic attacks
B. Category II: RNG-based cryptosystems
C. Category III: Research objectives of quantum-RNGs for cryptosystems

2.1. Category I: Cryptosystems and Cryptographic Attacks

In cryptology, a cryptography algorithm is an art of creating codes or algorithms that turn plain text into ciphertext and ciphertext into the original text. It establishes secure communication between two entities in the public domain, such as the Internet, where unauthorized users and information hackers are present.

The main difference between the cryptography algorithm is the relationship between the encryption and the decryption key. Logically, in any cryptographic system known as a cryptosystem, keys generated and expanded by KSA play an essential role in the cryptographic process.

A private key is used for both encryption and decryption, and algorithms are classified as symmetric key cryptography, whereas asymmetric key cryptography uses a pair of public and private keys for encryption and decryption or signing and verification.

Figure 3a,b visualizes symmetric key cryptography and asymmetric key cryptography, respectively.

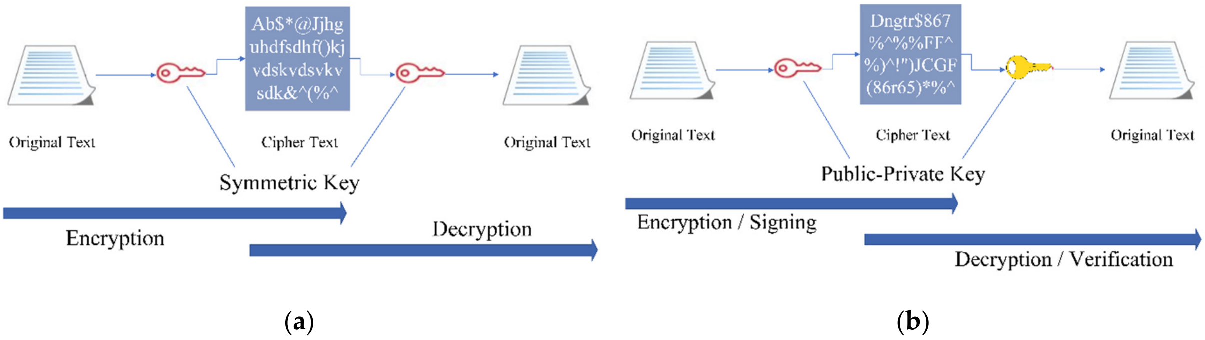


Figure 3. (**a**) Symmetric key cryptography; (**b**) Asymmetric key cryptography.

There are two types of symmetric key cryptography, block cipher and stream cipher.

Block cipher: A block cipher encrypts or decrypts a block of data. Encryption transforms plaintext into an equal length of ciphertext block, whereas the decryption performs the reverse process on the same block of ciphertext block. It primarily uses either of the following two network structures.

Feistel (F)-Network: In F-Network, the input data splits into two parts—the left and the right. The right part remains unchanged, and the left part goes through the operation with the key. After processing several rounds of operations designed by the algorithm, the combination of the left and right parts constitutes the ciphertext.

Substitution-permutation (SP) network: In the SP network, the substitution layer consists of a substitution (S)-box and operates on short data segments. Due to the layer's highly nonlinear property, it creates confusion in the internal ciphertext. The permutation layer diffuses the effect of substitution across the entire block.

Stream cipher: A stream cipher performs encryption or decryption on a digital data stream of 1 byte or 1 bit. When data is to be transmitted over a communication channel or through a web browser, this can be helpful. It depends on different structures broadly categorized as follows: arithmetic and XOR(AX), linear feedback shift register (LFSR), a combination of LFSR and nonlinear-feedback shift register (LFSR+NFSR), and pseudo-random function (PF)

Arithmetic and XOR(AX): Arithmetic operations include all bitwise operations (XOR, NAND, OR, Modulo operator) for cryptographic functions.

LFSR: LFSR is a shift register where the input bit is a previous state's linear function (exclusive-or (XOR)). The input is affected by previous stage values, so it operates as a feedback mechanism.

NFSR: NFSR is a shift register where the input bit is a previous state's non-linear function. The stream cipher also uses the LFSR+NFSR structure, which combines the LFSR and the NFSR.

PF: These functions are based on pseudo-random numbers generated by PRNGs.

Table 1 summarises the symmetric key cryptography categorized into block cipher and stream ciphers.

Equally important, asymmetric key cryptography, based on a pair of keys, is categorized into two approaches: public key cryptography and digital signature.

Public key cryptography: A public key cryptography system uses a pair of public and private keys. The KSA generates both the keys—a public key for the encryption and a private key for the decryption. Encryption is performed by the receiver's public key, generating the ciphertext, and the ciphertext can only be decrypted with the sender's private key. Factorization and Diffie–Hellman's (DH) key exchange are the two main techniques of public key cryptography.

Factorization: In factorization, a composite number, part of mathematical computation, is transformed into a product of smaller integers, whereas when the integers are primes, the process is known as prime factorization.

Table 1. List of symmetric key cryptography systems.

Symmetric-Key Cryptography **(A Secret Key Generated by KSA for Encryption & Decryption)**			
Block Ciphers		Stream Ciphers	
F Network	SP Network	AX	LFSR
1900's	1900's	1900's	1900's
— DES [12] — 3 DES [12] — IDEA [20] — Blowfish [21] — Tea [27] — CAST-128 [28] — XTEA [27] — RC2 [30] — RC6 [32] — CAST-256 [28] — MARS [33] — Twofish [34]	— Safer-K [13] — 3-Way [14] — SHARK [15] — AES [16]	— RC4 [17]	— A5/1 [18] — A5/2 [19]
		2000's	2000's
		— Rabbit [22] — SOBER-128 [23] — QUAD [24] — Trivium [25] — Salsa20 [26]	— SNOW 2 [29] — Crypto-1 [31]
		LFSR+NFSR	PF
		2000's	1900's
2000's			— ISAAC [17] — SEAL [39]
— Camellia, 2000 [38] — Threefish [40] — GOST [41]		— HC-256 [35] — Grain [36] — MICKEY [37]	2000's
			— Phelix [42]

Diffie–Hellman (DH) key exchange: DH is a cryptographic set of rules for exchanging the private key over an unsecured channel. The KSA of both the sender and receiver generates the public and private key pair. Both the parties share their public key. Once they get the public key, they calculate their secret or private key and use it for sending data securely.

Post-quantum public key cryptography: The cryptoanalysis of classical ciphers led researchers to develop quantum and classical computer-resistant cryptosystems. Moreover, these systems can communicate with an existing communication protocol, allowing their practical implementation for applications. The National Institute of Standards and Technology (NIST) [43] announced the four post-quantum public key encryption and key-establishment algorithms in round 3 compared to seventeen algorithms in round 2. NIST has started the process of standardizing these algorithms.

Digital signature: The digital signature is an electronic signature in which the sender signs or encrypts the document with a private key. The receiver will decrypt or verify that document with the sender's public key. It can be incorporated into cryptography by using the digital signature standard or elliptic curve cryptography.

Digital signature standard (DSS): DSS is a standard established by NIST to generate digital signatures.

Elliptic curve cryptography (ECC): A cryptographic algorithm based on elliptic curves generates smaller key sizes while providing the same level of security as those without them.

Post-quantum digital signature: NIST announced the three post-quantum digital signature algorithms in round 3 compared to nine algorithms in round 2. These are another part of post-quantum cryptography. Furthermore, NIST has started the process of standardizing these algorithms.

Table 2 shows the asymmetric key approaches used by public key cryptography and digital signatures.

Table 2. List of asymmetric key cryptography systems.

Asymmetric-Key Cryptography **(A Public and a Private Key Generated by KSA for Encryption/Signing & Decryption/Verification)**			
Public-Key cryptography		Digital Signature	
Factorization	DH-Key Exchange	DSS	ECC
			2000's
— RSA, [44] 1978	— ElGamal, [45] 1985	— DSA, [46] 1994	— ECIES, [47] — ECDSA, [47] — EdDSA, [48]
Post-Quantum Public-Key cryptography [43]		Post-Quantum Digital Signature [43]	
— BIKE		— CRYSTALS-DILITHIUM	
— CRYSTALS-KYBER	— Classic McEliece	— GeMSS	
— HQC	— FrodoKEM	— MQDSS	
— LEDAcrypt	— LAC	— qTESLA	
— NTRU	— NewHope	— SPHINCS+	
— NTS-KEM	— NTRU Prime	— FALCON	
— SIKE	— ROLLO	— LUOV	
— RQC	— Three Bears	— Picnic	
— Round5	— SABER	— Rainbow	

The security of symmetric key or asymmetric key depends on the vulnerability of their key to the cryptographic attacks. Cryptoanalysis analyzes cryptosystems to look for weak points or opportunities for information leaks to access or find the key. Cryptoanalysis is another part of cryptology that differentiates different cryptographic attacks. We have categorised the cryptographic attacks into six categories: differential cryptoanalysis, linear cryptoanalysis, meet-in-the-middle, side-channel attacks, related key attacks and other attacks.

Differential cryptoanalysis (DC): DC refers to a chosen-plaintext attack, where the attacker can choose a plaintext and find the corresponding ciphertext in order for them to obtain the key. By computing the differences between the ciphertexts, the attacker can detect statistical patterns in their distribution. These differences constitute a differential attack.

Linear cryptoanalysis (LC): LC finds affine approximations to the cipher's action based on linear equations. Depending on linear equations, it comes close to a plaintext–ciphertext pair and attempts to find the key.

Meet-in-the-middle (Mt-in-M): An Mt-in-M attack is a known-plaintext attack that breaks the long chain of encryption blocks into half to analyze the blocks independently and find the key more accurately. This accuracy cost depends on breaking the encryption chain into smaller parts requiring more storage. However, it is still more efficient than a brute force attack regarding time and computational complexity.

Side-channel attacks (SCA): In side-channel attacks, information leaks from a physical cryptosystem. In addition to timing, power consumption and electromagnetic emissions can also be exploited.

Related key Attacks (RKA): An attack model called related key is a subset of cryptoanalytic attacks in which the attacker is able to select or identify the relationship between multiple keys. These keys are then used for both encryption and decryption operations.

Other attacks: Other attacks, apart from DC, LC, M-in-M, SCA, and RKA, are included in this category.

Table 3 shows the survey of cryptographic attacks on different ciphers with the key size of KSA.

Table 3. Cryptanalysis of cryptography ciphers.

		Cryptoanalysis					
Ciphers	***KSA Key Size (Bits)***	***DC***	***LC***	***Mt-in-M***	***SCA***	***RKA***	***Others***
DES	56	√	√	√			Brute-Force
3 DES	112 or 168						Sweet32
IDEA	128			Bicliques Attack			
Blowfish	32–448						Birthday Attack
Tea	128	√				√	
GOST	256	√		Reflection		√	
CAST 128	40 to 128	√					
XTEA	128	√		√		√	
RC2	1–128 Bytes					√	
RC6	128, 192, 256						Statistics Attack
CAST 256	128, 160, 192, 224, 256	√					
Mars	128, 192, 256			√			
Twofish	128, 192, 256	Truncated, Impossible			Power Analysis	√	
Camellia	128, 192, 256				Cache Timing		Square Attack
Threefish	256, 512, 1024	Boomerang Attack					
Safer-K	64,128	Boomerang Attack, Impossible				√	
3-Way	96					√	
Serpent	128, 192, 256	Differential-Linear			Power Analysis		
AES	128, 192, 256			Bicliques Attack		√	Brute-Force
RSA	2048 to 4096						Shor's Algorithm

The survey shows that different attacks on a cryptosystem are possible, even on an advanced encryption system (AES), a highly secure block cipher. These attacks illustrate a requirement to modify various cryptosystems to make them more secure against cryptographic attacks. Figure 4 represents the graphic view of the cipher's cryptoanalysis by the specific cryptographic attacks.

In the next category, we have discussed different RNGs and have focused on cryptosystems based on RNGs.

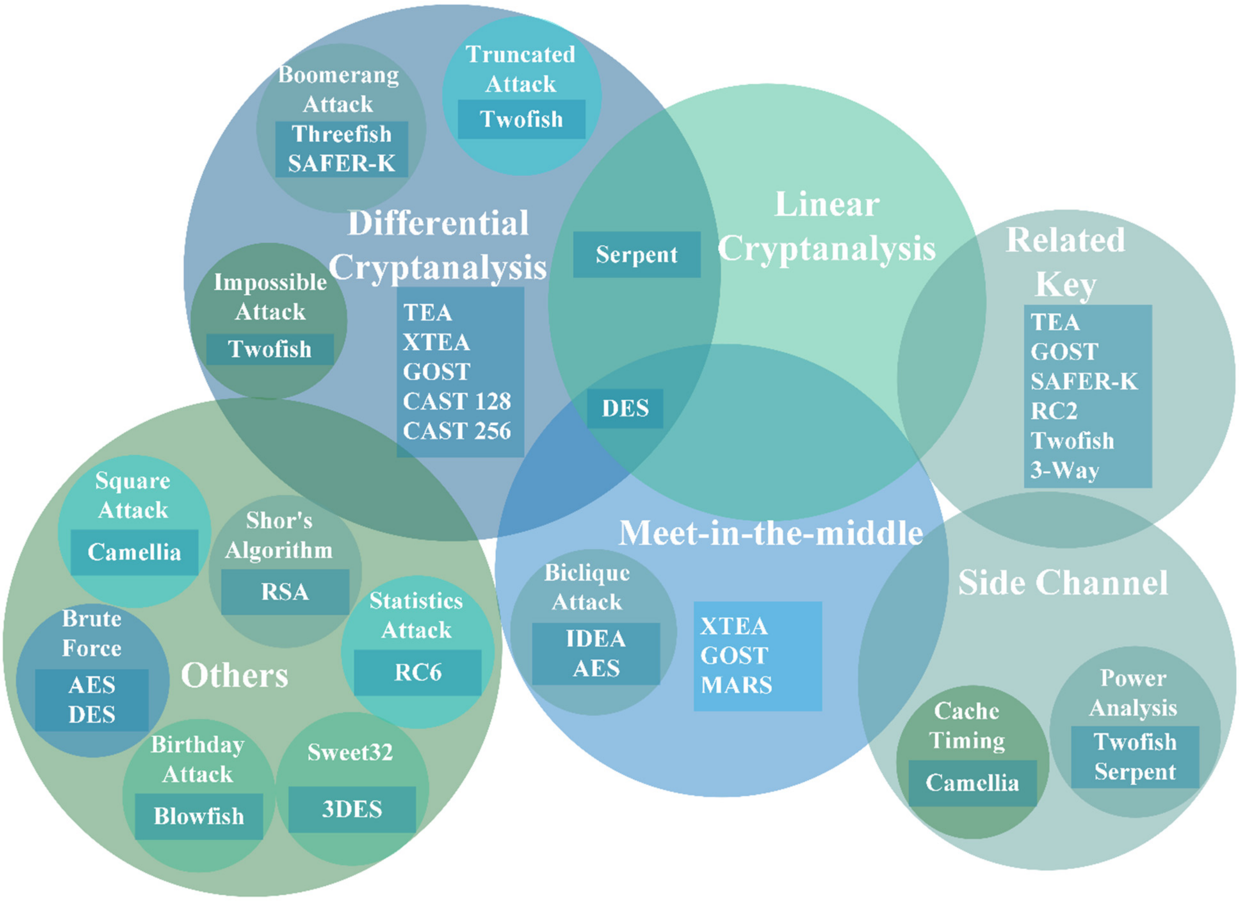


Figure 4. Graphic review of cryptographic cryptoanalysis.

2.2. Category II: RNG-Based Cryptosystems

The cryptosystems are affected by the objectives and operations to perform their functions. The RNGs can play a significant role in providing cryptography's fundamental function, randomness, and enhancement of the security of the cryptosystem. The randomness can be achieved by different random method generators.

- Pseudo-random number generator (PRNG)
- True random number generator (TRNG)
- Circuit design-based random number generator (CDRNG)
- Quantum random number generator (QRNG)

Pseudo-random number generator (PRNG): PRNGs are computer-based algorithms that use mathematical computation to generate random numbers. A PRNG sequence of the random number is periodic, which means the sequence repeats itself periodically, and deterministic, in that the sequence reproduction is possible if the initial value is known. Mathematical computations like linear congruential generators, LFSRs, and chaos provide pseudo-randomness.

A PRNG is a cryptographic secure PRNG, or CSPRNG if it passes the next-bit test (if the attacker knows the first k bits, the attacker cannot predict the k + 1 bits) and withstands the state compromise extensions (if any stream of bits is guessed correctly, the last bits cannot be predicted).

Programming languages have techniques for generating CSPRNG. The secrets [49] module from Python and SecureRandom [4] class from Java support CSPRNG.

True random number generator (TRNG): Random numbers generated by TRNGs are unbiased, independent, and unpredictable as produced by different physical phenomena such as infinite noise, voltage fluctuation, clock jitter, atmospheric radio noise, continuous and discrete time chaotic system.

There are various USB interface-based TRNGs that help to quickly connect with a laptop to work with any application. Alea II [50] by Araneus Information Systems, TrueRNG v3 [51] by ubldit, and infinite noise TRNG [52] by Crowd Supply are some of the USB-based TRNGs. On the other hand, RANDOM.ORG is a service-oriented online platform that provides TRNGs for various applications such as games, lotteries, and web pages.

Circuit design-based random number generator (CDRNG): Circuit design, including various electronic components, such as gates, integrated circuits, etc., is also helpful in generating RNGs. Pietro Nannipieri et al. [53] proposed TRNGs using the Fibonacci–Galois ring oscillator for cryptographically secure applications on a field programmable gate array (FPGA), resulting in the best entropy. Luca Crocetti et al. [54] proposed an all-digital RNG with high portability and entropy. As a result, it is ready to integrate with the European Processor Initiative (EPI) [55] chip.

Quantum random number generator (QRNG): The third classification of RNG is quantum-based and derived from quantum mechanics. Different QRNGs generate established and robust outcomes by applying different entropy sources, the source of randomness, based on quantum physics.

USB-based QRNGs are well tested and certified for generating high-quality and unique random numbers such as Quantis [56] by IDQ. They also invented a chip-based QRNG that can fit into small devices like smartphones. Another project named QRANGE [57] under the quantum flagship focused on CMOS technology generating quantum random numbers. ANU QRNG [58] is an online platform where anybody can generate quantum random numbers with just one click and use them.

Figure 5 illustrates the categories of RNGs and their implementation methodology or practical approach for integrating them into a cryptosystem.

These RNGs can be used in the RNG-based cryptosystem to enhance the security in the KSA or encryption. Table 4 shows the survey of RNG-based cryptosystems, improving different parameters due to the random numbers generated by incorporating RNGs. Moreover, the survey indicates that RNGs can be used with any cryptosystem and make it more secure than the system without RNGs.

Table 4. Survey of RNG-based cryptosystems.

Cipher Design	RNG	Type of Cryptosystem	Result	Parameters Improved	Ref.
Blostream	PRNG	Symmetric	Immune to Brute-Force, Statistical, Deferential, Distinguishing and Correlations Attacks	➢ Speed, ➢ Memory Requirements	[59]
Hybrid Cryptosystem	PRNG + TRNG	Symmetric	Strong Key	➢ Non-deterministic ➢ Unpredictable ➢ Reproducible ➢ Periodic	[60]
Text Encryption Stream	PRNG	Symmetric	Immune To All Known-Plaintext Cryptanalysis Attacks	➢ Large Plaintext ➢ Key Sensitivity	[61]
Present	PRNG + TRNG	Symmetric	Better Performance with High Security	➢ Slices ➢ LookUp Table ➢ Frequency ➢ Power	A [62]

Table 4. *Cont.*

Cipher Design	RNG	Type of Cryptosystem	Result	Parameters Improved	Ref.
Text Encryption Algorithms	PRNG	Symmetric	Strength against Linear, Differential and Statistical Attacks	➢ Plaintext sensitivity ➢ Key sensitivity ➢ Robustness against known plaintext attack	[63]
Diffie-Hellman Key Exchange—Using QRNG	QRNG	Asymmetric	Non-Vulnerable Cryptographic System	➢ Key Entropy ➢ Plaintext Entropy ➢ Ciphertext Entropy	B [64]
CCAES—Chaos-based	PRNG	Symmetric	More Secure and Effective Resistant to Differential Attacks	➢ Histogram ➢ Correlation ➢ NPCR and UACI ➢ Information Entropy	[65]
Image Encryption Algorithm—Chaos-based	PRNG	Symmetric	Security Enhancement	➢ Entropy ➢ Cross Correlation ➢ Mean Square Error ➢ PSNR	[66]
Authenticated Encryption with Associated Data (AEAD)—Chaos-based	PRNG	Symmetric	Highly Secure for Ciphertext and Authentication Tag Resistant to Differential, Linear, Algebraic, and Timing Attacks.	➢ Privacy and Integrity ➢ Measured by Statistical Test suite NIST, DIEHARD and ENT.	C [67]
Hybrid RSA	PRNG	Asymmetric	Strong Encryption	➢ Histogram ➢ Correlation ➢ NPCR and UACI ➢ Key Sensitivity ➢ Key Space ➢ Information Entropy	[68]

NPCR—Number of Changing Pixel Rate, UACI—Unified Averaged Changed Intensity, PSNR—Peak Signal to Noise Ratio. Note: Publication project supported by: A—The Xiamen University Malaysia Research Fund (XMUMRF); B—Prometeo Project of the Ministry of Education Superior, Science, Technology and Innovation of the Republic of Ecuador; C—Fundamental Research Grant Scheme funded by the Ministry of Higher Education of Malaysia (MOHE).

In the next category, we have focused on QRNG and its research design, which is the basis of incorporating quantum randomness into a cryptosystem.

2.3. Category III: Research Objectives of Quantum RNGs for Cryptosystems

QRNGs as an external device are best suited when the sender and receiver, e.g., Alice and Bob, are the same. Alice wants to store data like personal files (driving license, passport, other identity proofs) and highly secure work files like military data on the cloud. Consequently, the data should be encrypted before sending it to the network and stored in the cloud. Here, KSA comprises the QRNG bits to make the encrypted key stronger and more complex.

We have discussed classifications of RNGs in category II, and the survey shows the improvement in various cryptosystem ciphers even using pseudo-randomness, which is not truly random.

It is impossible to consider true randomness in PRNG-generated [5] sequences as they are implemented by software, based on mathematical algorithms and determined by an initial (seed) value. In contrast, the TRNG and QRNG, based on unpredictable physical means, generate the true randomness in the sequence of random numbers. Furthermore, there are two significant differences between TRNGs and PRNGs. TRNGs [69] are non-deterministic and use a physical mechanism, whereas PRNGs are deterministic and utilize mathematical algorithms. Although TRNGs are unpredictable, they are still vulnerable to attack because if a failure occurs in the TRNG hardware system, it is hard to detect it.

Dr. Mads Haahr [70] introduced the service of providing different TRNGs online. Some are freely available like number-based TRNGs, lists, strings, and map-based TRNGs; some are paid and used for random drawings. Web tools and widget- based TRNGs are also freely available for the website to be integrated into the webpage and provide the TRNG-based random number. As atmospheric noise creates these random numbers, they are better than pseudo-random numbers.

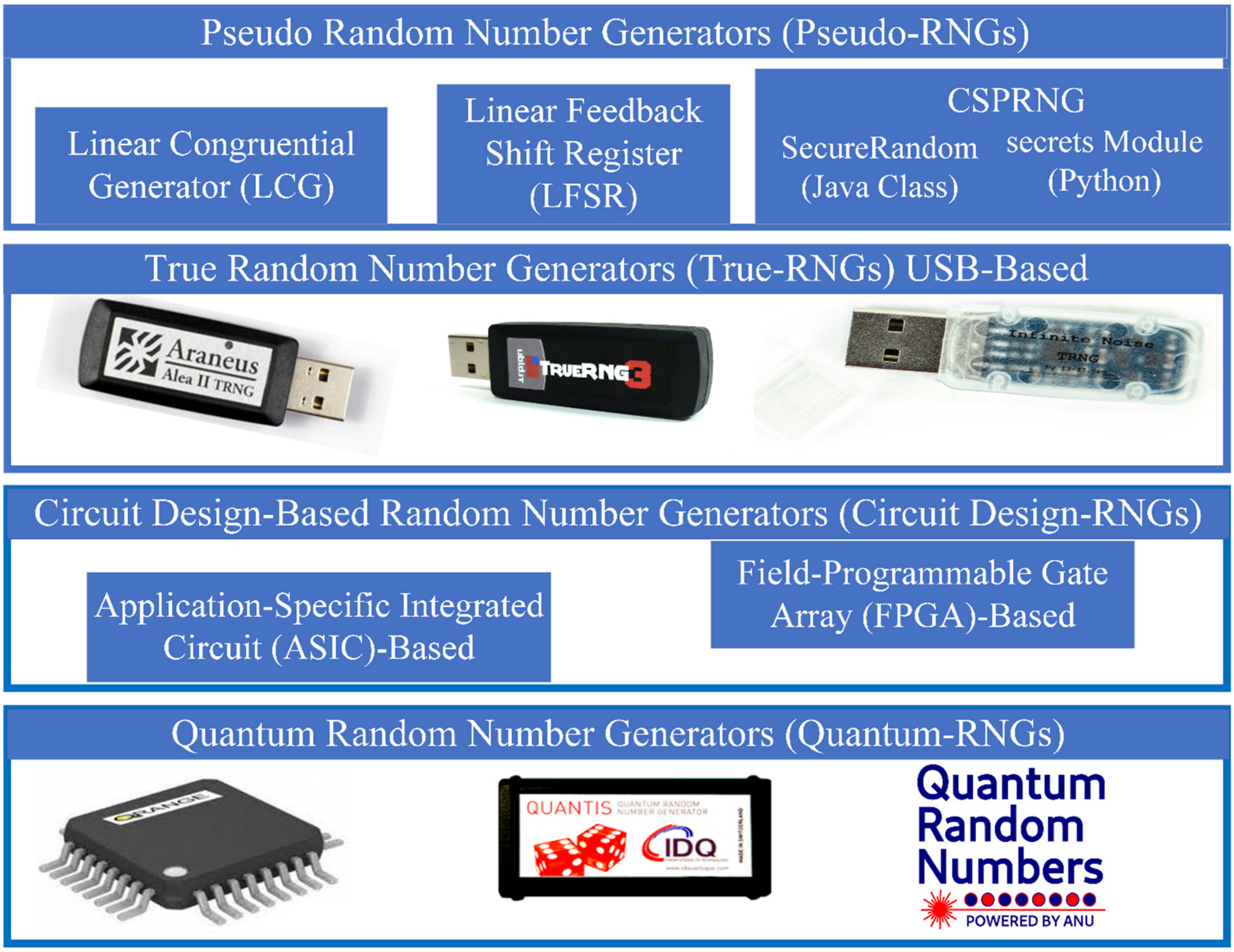

Pseudo Random Number Generators (Pseudo-RNGs)
Linear Congruential Generator (LCG)
Linear Feedback Shift Register (LFSR)
CSPRNG
SecureRandom (Java Class)
secrets Module (Python)
True Random Number Generators (True-RNGs) USB-Based
Araneus Alea II TRNG
TRUERNG3
Infinite Noise TRNG
Circuit Design-Based Random Number Generators (Circuit Design-RNGs)
Application-Specific Integrated Circuit (ASIC)-Based
Field-Programmable Gate Array (FPGA)-Based
Quantum Random Number Generators (Quantum-RNGs)
ORANGE
QUANTIS QUANTUM RANDOM NUMBER GENERATOR
IDQ
Quantum Random Numbers
POWERED BY ANU

Figure 5. Categorization of random number generators: Implementation with applications.

In some cases of TRNGs, the entropy generated by an entropy source may be confused with thermal noise or shot noise [71]. However, a highly secure and unique random number is based on the quantum principle and generated by a QRNG [4]. A QRNG-based cryptosystem incorporates quantum randomness to generate a more robust key of the KSA or enhance encryption security.

Furthermore, Shor's algorithm [72] concluded that quantum computers could easily break public key cryptography. It proposed to solve the prime factorization of RSA and results with high probability. The quantum random numbers generated by QRNGs are also quantum-based and might secure the new cryptosystem without a prime number. Also, incorporating quantum randomness challenges this type of quantum algorithm for cryptoanalysis of the cipher.

In this category, we discuss the QRNG, which generates unbiased, high-speed, and unpredictable random numbers by various methods such as laser pulses, phase diffusion, photon-based methods, and other quantum mechanics techniques. Figure 6 shows the difference between RNGs in terms of their generation methods, properties, disadvantages, and advantages.

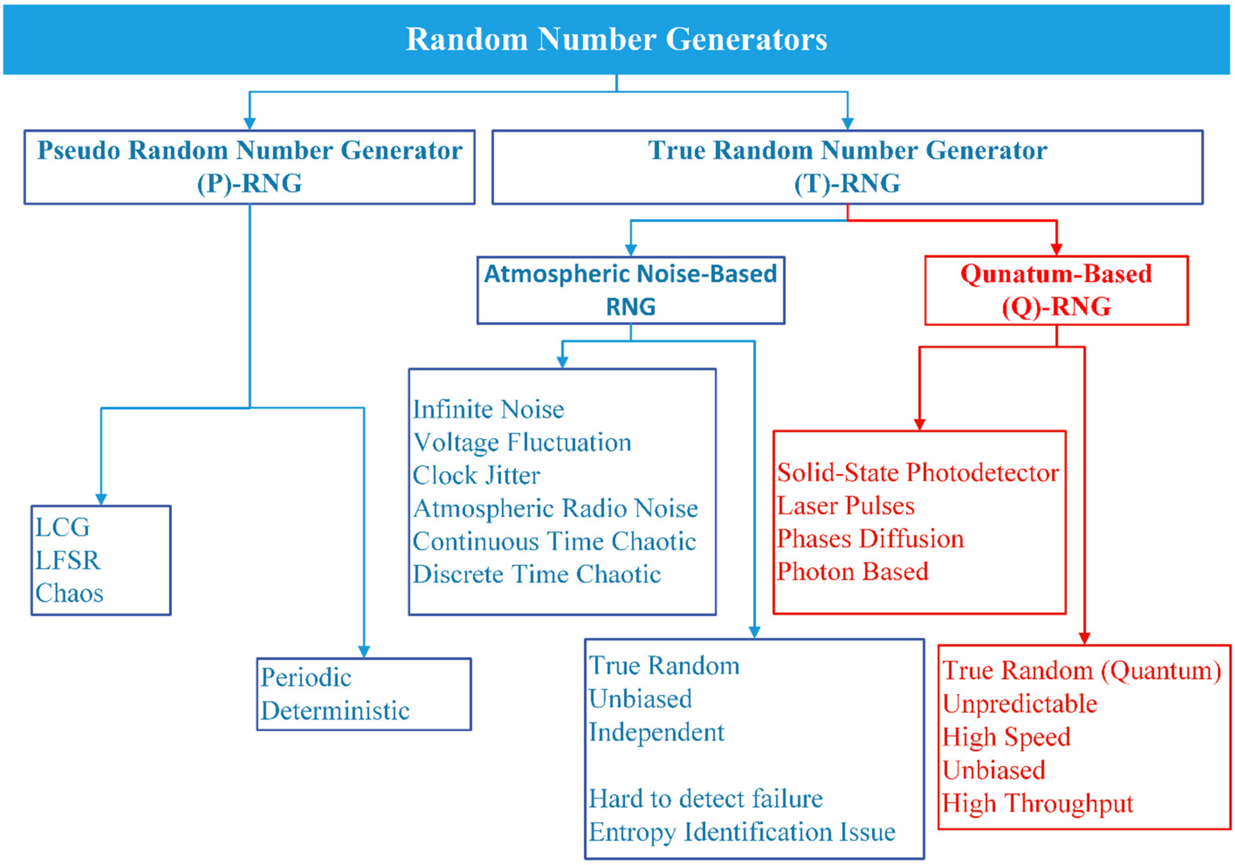


Figure 6. Categorization of Random Number Generators: generation methods, properties, disadvantages advantages.

We have reviewed 20 QRNGs and divided them into three important research objectives: High speed (HS), bias improvement (BI) and high efficiency (HE).

High speed (HS): The speed of the RNG device is indicated by the rate at which random bits are generated per second by the device. Speed is critical in determining the optimal trade-off for the RNG device. Different research methodologies attempt to identify the shortest time it takes for the device to generate the most random bits.

Bias improvement (BI): Random bits should be bias-free. Bias occurs when a significantly more zeros than ones, or vice versa, are generated. If the difference between zeros and ones is small enough, it should not introduce bias into the quantum random bits. Numerous research proposals have been designed to improve the bias in order to achieve balanced quantum random bits.

High efficiency (HE): The performance of a QRNG can be improved by researching different technologies on which the generator can be built. The efficiency of a device is defined in terms of extracting random bits or maximising the percentage of bits with specific physical photon-based conditions.

Table 5 shows the survey of different quantum random number generation techniques and their research objectives.

The next section addresses some open research challenges specific to QRNG-based cryptosystems.

Table 5. Literature review of QRNGs.

QRNG Proposal Techniques	Ref.	HS	BI	HE	Other Properties
Weak laser pulses	[73]	√	√		
Random quantum states in mathematica	[74]	√			
Tapered amplifiers	[75]		√		Self-Detecting
One single-photon detector	[76]		√		
Photon arival time selectively	[77]		√		
Three-types QRNG	[78]	√			Self-testing
Ultra-fast QRNG—pulsed laser diode	[79]	√			
Heterodyne-based	[80]	√			Security
16 × 16 pixel QRNG based on SPADs	[81]	√		√	
Laser phase fluctuations	[82]	√			
One and two entropy sources	[83]			√	
Phase diffusion process-based	[84]	√			
SPAD-based QRNG using FPGA	[85]			√	
SPAD-based QRNG pixel-based	[86]			√	
Multi-bit	[87]			√	
Uncharacterized laser andsunlight	[88]	√			
Coupled quantum dots	[89]		√		
Phase diffusion in lasers	[90]		√		Interference Quality, Input/Output Monitoring
Quantis—USB	[8]	√	√		Autocalibration Status Monitoring
Qrange	[57]	√			Security

Note: HS, high speed; BI, bias improvement; HE, high efficiency.

3. Open Research Problems

In Section II, we reviewed the literature, and our analysis identified some open research problems to address and enhance the security of cryptosystems. RNGs play a significant role in securing systems that use random numbers. QRNs [9,32] are based on true randomness and can be used to strengthen the security of existing cryptography systems.

Future research will be needed to address the following challenges:

(a) The analysis of incorporating the quantum randomness in stream cipher operations compared to pseudo-based ciphers.

Boolean arithmetic-based cryptography mainly uses cryptographic-secure pseudo-random number generators (CSPRNG). A pseudo-Hadamard transform based on the modular operation of boolean arithmetic provides diffusion in cryptographic cipher. The diffusion depends on the change of an input bit. RNGs can generate the input bit to affect the cryptanalysis of the cipher.

Guillermo Sosa-Gómez et al. [91] showed the cryptanalysis of PRNGs for cryptographic ciphers. Also, a correlation analysis of the entropy of PRNGs and Hadamard values indicate a strong correlation. On the other hand, QRNG based on the photon can mitigate this cryptoanalysis.

For most CSPRNGs, the entropy is derived from the operating system, along with a PRNG generator [92]. The underlying PRNG often “reseeds”, which means that when an entropy is supplied by the operating system (e.g., from user input, system interruptions, disk I/O, or hardware random generators), it changes its internal state. However, quantum randomness is derived from quantum particles without the intervention of reseeding.

(b) Design a KSA by using different entropy sources for quantum random bits in order to randomize the keyspace for differential attacks.

The differential attacks try to find the key with the chosen plaintext and corresponding ciphertext. The subkeys of KSA are vulnerable if the keyspace between them is not random or large. Incorporating QRNGs can generate random subkeys, increasing or enhancing the randomness of their keyspace. The stronger the keyspace, the more challenging it is to detect even a single subkey. As a result, a QRNG-based KSA may produce a strong cipher that is not vulnerable to differential attack.

(c) An in-depth study is needed to analyse the effects of key-related attacks on QRN-based ciphers.

In key-related attacks, the attacker knows (or chooses) a relation between several keys (up to 256 in some recent attacks) and is given access to encryption functions with such related keys. Keys and plaintexts can be selected with specific differences when using differential related-key attacks.

The QRNG will support the true randomness in the cryptography key against key-related attacks. The effect of the true randomness on the cryptanalysis of the cipher is to be analysed to secure the key and to make it truly random.

(d) Designing a new high-speed encryption cipher based on different research designs (high-speed and bias-free) by using QRNs to enhance the security of the cryptosystem.

The encryption speed of the block cipher is one of the major primary concerns when designing a new cipher. However, the security of the cipher must not be compromised on the basis of time, but the time complexity of an algorithm can make it more competitive against alternatives. The QRNG will add true randomness into the key by generating the quantum random bits from an external photon device. These random bits are highly important to secure the cipher, but might come with a time cost. New cryptographic primitives are sought after that reduce this time and introduce quantum randomness, leading to a highly secure cryptosystem.

(e) Research analysis on storing and exchanging the QRNG-based keys generated for asymmetric cryptosystem over the cloud.

The QRNGs are based on devices that can generate photons. These devices can be with sender or receiver, and are hence important when both have the same key, i.e., symmetric cryptosystem. They help generate two keys, a pair of public and private keys, but transfer them with the knowledge of quantum random bits and store them on the cloud as staging post needs further research.

4. Conclusions

We have surveyed cryptosystems, different cryptanalysis techniques, and RNGs. The analysis shows that cryptoanalysis is possible for various ciphers and, therefore, requires modification in the cryptosystem to secure them. RNGs provide random numbers in the cryptosystem and enhance the cipher’s security by making the encryption robust or resistant to various cryptographic attacks. In addition, QRNGs provide true randomness, compared to PRNGs and atmospheric noise-based TRNGs, because they are based on quantum principles by different theories of quantum physics. Finally, the identification of open research problems guides researchers in the cryptography security field to improve the security based on the quantum randomness in a cryptosystem.

Funding: This research review work received funding by the University of Hertfordshire, United Kingdom.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Stallings, W. *Cryptography and Network Security: Principles and Practices*, 4th ed.; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 2005.
2. Smart, N.P.; Rijmen, V.; Warinschi, B.; Watson, G. Algorithms, Key Sizes and Parameters Report. ENISA, Nov. 2014. Available online: https://www.enisa.europa.eu/publications/algorithms-key-size-and-parameters-report-2014 (accessed on 9 September 2021).

3. Sahmoud, S.; Elmasry, W.; Shadi, A. Enhancement the Security of AES Against Modern Attacks by Using Variable Key Block Cipher. *Int. Arab. J. e-Technol.* **2013**, *3*, 17–26.
4. Oracle, SecureRandom. 2020. Available online: https://docs.oracle.com/javase/8/docs/technotes/guides/security/StandardNames.html#SecureRandom (accessed on 23 June 2020).
5. Sinha, S.; Islam, S.H.; Obaidat, M.S. A comparative study and analysis of some pseudorandom number generator algorithms. *Secur. Priv.* **2018**, *1*, e46. [CrossRef]
6. Gong, L.; Zhang, J.; Liu, H.; Sang, L.; Wang, Y. True Random Number Generators Using Electrical Noise. *IEEE Access* **2019**, *7*, 125796–125805. [CrossRef]
7. Melia, J.; Huttner, B.; Moulds, R.; Walenta, N.; Fuller, A.; Quantum-Safe Security Working Group. *Quantum Random Number Generators*; Cloud Security Alliance: Bellingham, WA, USA, 2016.
8. ID Quantique, What Is the Q in QRNG ? 2020. Available online: https://www.idquantique.com/random-number-generation/overview/ (accessed on 7 July 2020).
9. ID Quantique, Understanding Quantum Cryptography. ID Quantique SA. 2020. Available online: https://www.idquantique.com/quantum-safe-security/quantum-key-distribution/ (accessed on 7 July 2020).
10. ID Quantique, Gaming-and-Lotteries. Available online: https://www.idquantique.com/random-number-generation/applications/gaming-and-lotteries/ (accessed on 7 July 2020).
11. Open Quantum Safe. Available online: https://openquantumsafe.org/ (accessed on 9 May 2022).
12. Biryukov, A.; de Cannière, C. Data encryption standard (DES). In *Encyclopedia of Cryptography and Security*; Springer: Boston, MA, USA, 1999. [CrossRef]
13. Massey, J.L. SAFER K-64: A byte-oriented block-ciphering algorithm. In Proceedings of the International Workshop on Fast Software Encryption, Cambridge, UK, 9–11 December 1993; Lecture Notes in Computer Science. Springer: Berlin/Heidelberg, Germany, 1994; Volume 809, pp. 1–17. [CrossRef]
14. Daemen, J.; Govaerts, R.; Vandewalle, J. A new approach to block cipher design. In Proceedings of the International Workshop on Fast Software Encryption, Cambridge, UK, 9–11 December 1993; Springer: Berlin/Heidelberg, Germany, 1994; pp. 18–32.
15. Anderson, R.; Biham, E.; Knudsen, L. Serpent: A Proposal for the Advanced Encryption Standard. NIST AES Proposal. 1998, pp. 1–23. Available online: https://bitbucket.org/nicholascapo/network-security-project/src/fcbc6e93e555/Literature/serpent.pdf (accessed on 9 October 2021).
16. Daemen, J.; Rijmen, V. *The Design of Rijndael*; Springer: Berlin/Heidelberg, Germany, 2002.
17. Jenkins, R.J., Jr. ISAAC and RC4. 1993. Available online: http://burtleburtle.net/bob/rand/isaac.html (accessed on 12 June 2022).
18. Quirke, J. Security in the GSM System. AusMobile. 1 May 2004. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.108.1509&rep=rep1&type=pdf (accessed on 9 October 2021).
19. Security Algorithms Group of Experts, ETR 278—ETSI Technical Report. 1996. Available online: https://cryptome.org/espy/ETR278e01p.pdf (accessed on 4 May 2021).
20. Lai, X.; Massey, J.L. A Proposal for a New Block Encryption Standard. In Proceedings of the Advances in Cryptology—EUROCRYPT ’90, Workshop on the Theory and Application of Cryptographic Techniques, Aarhus, Denmark, 21–24 May 1990; Lecture Notes in Computer Science. Damgård, I.B., Ed.; Springer: Berlin/Heidelberg, Germany, 1991; Volume 473, pp. 389–404. [CrossRef]
21. Schneier, B. Description of a new variable-length key, 64-bit block cipher (Blowfish). In Proceedings of the International Workshop on Fast Software Encryption, Cambridge, UK, 9–11 December 1993; Springer: Berlin/Heidelberg, Germany, 1994; pp. 191–204.
22. Boesgaard, M.; Vesterager, M.; Pedersen, T.; Christiansen, J.; Scavenius, O. Rabbit: A new high-performance stream cipher. In Proceedings of the 10th International Workshop, Fast Software Encryption 2003, Lund, Sweden, 24–26 February 2003; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer: Berlin/Heidelberg, Germany, 2003; Volume 2887, pp. 307–329. [CrossRef]
23. Hawkes, P.; Rose, G.G. Primitive Specification for SOBER-128. IACR Cryptology ePrint Archive. 2003, p. 81. Available online: http://dblp.uni-trier.de/db/journals/iacr/iacr2003.html#HawkesR03a (accessed on 2 January 2020).
24. Berbain, C.; Gilbert, H.; Patarin, J. QUAD: A multivariate stream cipher with provable security. *J. Symb. Comput.* **2009**, *44*, 1703–1723. [CrossRef]
25. Christophe, D.C.; Preneel, B. Trivium Specifications. 2006, Volume 507932. Available online: https://www.ecrypt.eu.org/stream/p3ciphers/trivium/trivium_p3.pdf (accessed on 2 January 2020).
26. Bernstein, D.J. The salsa20 family of stream ciphers. In *New Stream Cipher Designs*; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin/Heidelberg, Germany, 2008; Volume 4986, pp. 84–97. [CrossRef]
27. Wheeler, D.J.; Needham, R.M. TEA, a tiny encryption algorithm. In Proceedings of the Fast Software Encryption, Second International Workshop, Leuven, Belgium, 14–16 December 1994; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer: Berlin/Heidelberg, Germany, 1995; pp. 363–366. [CrossRef]
28. Adams, C. The CAST-128 Encryption Algorithm. RFC Editor. May 1997. Available online: https://www.rfc-editor.org/info/rfc2144 (accessed on 12 June 2021).

29. Ekdahl, P.; Johansson, T. A new version of the stream cipher SNOW. In Proceedings of the Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002, St. John's, NL, Canada, 15–16 August 2002; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer: Berlin/Heidelberg, Germany, 2003; Volume 2595, pp. 47–61. [CrossRef]
30. Knudsen, L.R.; Rijmen, V.; Rivest, R.L.; Robshaw, M.J.B. On the Design and Security of RC_2. In Proceedings of the Fast Software Encryption, 5th International Workshop, FSE '98, Paris, France, 23–25 March 1998; Springer: Berlin/Heidelberg, Germany, 1998; pp. 206–221.
31. Crypto-1. Available online: https://en.wikipedia.org/wiki/Crypto-1 (accessed on 12 June 2021).
32. Rivest, R.L.; Robshaw, M.J.B.; Sidney, R.; Yin, Y.L. The RC6 Block Cipher. In Proceedings of the First Advanced Encryption Standard (AES) Conference, Ventura, CA, USA, 20–22 August 1998; p. 16.
33. Burwick, C.; Coppersmith, D.; D'Avignon, E.; Gennaro, R.; Halevi, S.; Jutla, C.; Matyas, S.M., Jr.; O'Connor, L.; Peyravian, M.; Safford, D.; et al. MARS—A Candidate Cipher for AES. NIST AES Proposal. 1998, pp. 8–23. Available online: http://cryptosoft.de/docs/Mars.pdf (accessed on 12 June 2021).
34. Schneier, B.; Kelsey, J.; Whiting, D.; Wagner, D.; Hall, C.; Ferguson, N. Twofish: A 128-Bit Block Cipher. NIST AES Proposal. 1998, Volume 15, pp. 1–27. Available online: https://www.schneier.com/wp-content/uploads/2016/02/paper-twofish-paper.pdf (accessed on 7 July 2021).
35. Wu, H. Stream Cipher HC-256. Available online: https://eprint.iacr.org/2004/092.pdf (accessed on 9 May 2022).
36. Hell, M.; Johansson, T.; Meier, W. Grain: A stream cipher for constrained environments. *Int. J. Wirel. Mob. Comput.* **2007**, *2*, 86–93. [CrossRef]
37. Babbage, S.; Dodd, M. The Stream Cipher MICKEY 2.0. ECRYPT Stream Cipher Project, Report. 2006, pp. 1–12. Available online: https://www.cosic.esat.kuleuven.be/ecrypt/stream/p2ciphers/mickey/mickey_p2.pdf (accessed on 12 June 2021).
38. Japan's First 128-Bit Block Cipher 'Camellia' Approved as a New Standard Encryption Algorithm in the Internet. NTT News Release. Available online: https://www.ntt.co.jp/news/news05e/0507/050720.html (accessed on 12 June 2021).
39. Fontaine, C. SEAL. In *Encyclopedia of Cryptography and Security*; Springer: Boston, MA, USA, 2011; p. 543.
40. Ferguson, N.; Lucks, S.; Schneier, B.; Whiting, D.; Bellare, M.; Kohno, T.; Callas, J.; Walker, J. Threefish. Available online: https://www.schneier.com/academic/skein/threefish/ (accessed on 2 August 2021).
41. Cannière, C. GOST. In *Encyclopedia of Cryptography and Security*; Springer: Boston, MA, USA, 2011. [CrossRef]
42. Whiting, D.; Schneier, B.; Lucks, S.; Muller, M. Phelix. 2004. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.8097&rep=rep1&type=pdf (accessed on 7 April 2021).
43. NIST, Post-Quantum Cryptography. Available online: https://csrc.nist.gov/Projects/post-quantum-cryptography/round-2-submissions (accessed on 12 June 2022).
44. Rivest, R.L.; Shamir, A.; Adleman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
45. ElGamal, T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. Available online: https://link.springer.com/content/pdf/10.1007/3-540-39568-7_2.pdf (accessed on 9 May 2022).
46. *FIPS 186*; Digital Signature Standard (DSS). National Institute of Standards and Technology: Gaithersburg, MD, USA, 1994.
47. *SEC 1 Ver. 2.0*; Standards for Efficient Cryptography, SEC 1: Elliptic Curve Cryptography. Certicom Research: Mississauga, ON, Canada, 2009.
48. Josefsson, S.; Liusvaara, I. Edwards-Curve Digital Signature Algorithm (EdDSA). *J. Chem. Inf. Modeling* **2017**, *53*, 1689–1699. Available online: https://www.rfc-editor.org/info/rfc8032 (accessed on 12 December 2021).
49. Python Software Foundation, Secrets. 2022. Available online: https://docs.python.org/3/library/secrets.html (accessed on 9 May 2022).
50. Araneus Information Systems Oy, Araneus Alea II. 2022. Available online: https://www.araneus.fi/products/alea2/en/ (accessed on 9 May 2022).
51. Ubld.it, TrueRNG v3. 2022. Available online: https://ubld.it/truerng_v3 (accessed on 9 August 2021).
52. Crowd Supply, Infinite Noise TRNG. 2022. Available online: https://www.crowdsupply.com/leetronics/infinite-noise-trng (accessed on 9 May 2022).
53. Nannipieri, P.; Di Matteo, S.; Baldanzi, L.; Crocetti, L.; Belli, J.; Fanucci, L.; Saponara, S. True random number generator based on fibonacci-galois ring oscillators for fpga. *Appl. Sci.* **2021**, *11*, 3330. [CrossRef]
54. Crocetti, L.; di Matteo, S.; Nannipieri, P.; Fanucci, L.; Saponara, S. Design and Test of an Integrated Random Number Generator with All-Digital Entropy Source. *Entropy* **2022**, *24*, 139. [CrossRef]
55. European Processor Initiative (EPI). Available online: https://www.european-processor-initiative.eu/ (accessed on 12 June 2022).
56. IDQ, Quantis-Random-Number-Generator. 2020. Available online: https://www.idquantique.com/random-number-generation/products/quantis-random-number-generator (accessed on 7 July 2020).
57. QRANGE, Qrng. 2020. Available online: https://qrange.eu/project/qrng (accessed on 27 July 2020).
58. ANU QRNG. 2022. Available online: https://qrng.anu.edu.au/ (accessed on 5 September 2021).
59. Kashmar, A.H.; Ismail, E.S. Blostream: A high speed stream cipher. *J. Eng. Sci. Technol.* **2017**, *12*, 1111–1128.
60. Patnala, T.R. A Modernistic way for KEY Generation for Highly. In Proceedings of the 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; pp. 892–897.

61. Amro, A.; El-Alfy, E.S.M. Known-plaintext attack and improvement of PRNG-based text encryption. In Proceedings of the 7th International Conference on Information and Communication Systems, ICICS 2016, Irbid, Jordan, 5–7 April 2016; pp. 233–238. [CrossRef]
62. Kowsalya, T.; Babu, R.G.; Parameshachari, B.D.; Nayyar, A.; Mehmood, R.M. Low area PRESENT cryptography in FPGA using TRNG-PRNG key generation. *Comput. Mater. Contin.* **2021**, *68*, 1447–1465. [CrossRef]
63. Mishra, M.; Mankar, V.H. Text Encryption Algorithms based on Pseudo Random Number Generator. *Int. J. Comput. Appl.* **2015**, *111*, 1–6. [CrossRef]
64. Mogos, G. Use quantum random number generator in Diffie-Hellman key exchange protocol. In Proceedings of the 2016 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, 19–21 May 2016; pp. 1–6. [CrossRef]
65. Arab, A.; Rostami, M.J.; Ghavami, B. An image encryption method based on chaos system and AES algorithm. *J. Supercomput.* **2019**, *75*, 6663–6682. [CrossRef]
66. Banthia, A.K.; Tiwari, N. Image Encryption using Pseudo Random Number Generators. *Int. J. Comput. Appl.* **2013**, *67*, 1–8. [CrossRef]
67. Teh, J.S.; Samsudin, A. A chaos-based authenticated cipher with associated data. *Secur. Commun. Netw.* **2017**, *2017*, 9040518. [CrossRef]
68. Çavuşoğlu, Ü.; Akgül, A.; Zengin, A.; Pehlivan, I. The design and implementation of hybrid RSA algorithm using a novel chaos based RNG. *Chaos Solitons Fractals* **2017**, *104*, 655–667. [CrossRef]
69. Hughes, R.; Nordholt, J. Strengthening the Security Foundation of Cryptography with Whitewood's Quantum-Powered Entropy Engine. 2016. Available online: http://www.whitewoodencryption.com (accessed on 3 August 2021).
70. Mads, H. Random.org. 1998. Available online: https://www.random.org/ (accessed on 12 June 2022).
71. Lee, J.; Seo, Y.; Heo, J. Analysis of random number generated by quantum noise source and software entropy source. In Proceedings of the 9th International Conference on Information and Communication Technology Convergence: ICT Convergence Powered by Smart Intelligence, Maison Glad Jeju, Jeju Island, Korea, 17–19 October 2018; pp. 729–732. [CrossRef]
72. Shor, P.W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* **1997**, *26*, 1484–1509. [CrossRef]
73. Wei, W.; Guo, H. Quantum random number generator based on the photon number decision of weak laser pulses. In Proceedings of the Optics InfoBase Conference Papers, Shanghai, China, 30 August–3 September 2009. Available online: http://arxiv.org/abs/0811.0082 (accessed on 9 May 2022).
74. Miszczak, J.A. Employing online quantum random number generators for generating truly random quantum states in Mathematica. *Comput. Phys. Commun.* **2013**, *184*, 257–258. [CrossRef]
75. Pooser, R.C.; Evans, P.G.; Humble, T.S.; Grice, W.P.; Williams, B.P. Self correcting quantum random number generators using tapered amplifiers. In Proceedings of the Optics InfoBase Conference Papers, Kyoto, Japan, 30 June–4 July 2013. [CrossRef]
76. Soares, E.d.L.; Mendonca, F.A.; Ramos, R.V. Quantum random number generator using only one single-photon detector. *IEEE Photonics Technol. Lett.* **2014**, *26*, 851–853. [CrossRef]
77. Wang, J.M.; Xie, T.Y.; Zhang, H.F.; Yang, D.X.; Xie, C.; Wang, J. A bias-free quantum random number generation using photon arrival time selectively. *IEEE Photonics J.* **2015**, *7*. [CrossRef]
78. Ma, X.; Yuan, X.; Cao, Z.; Qi, B.; Zhang, Z. Quantum random number generation. *npj Quantum Inf.* **2016**, *2*, 16021. [CrossRef]
79. Siswanto, M.; Rudiyanto, B. Designing of quantum random number generator (QRNG) for security application. In Proceedings of the 3rd International Conference on Science in Information Technology: Theory and Application of IT for Education, Industry and Society in Big Data Era, ICSITech, Bandung, Indonesia, 25–26 October 2017; pp. 273–277. [CrossRef]
80. Avesani, M.; Marangon, D.G.; Vallone, G.; Villoresi, P. Quantum Random Number Generator at 17 Gbps. *Nat. Commun.* **2018**, *9*, 5365. [CrossRef]
81. Xu, H.; Perenzoni, D.; Tomasi, A.; Massari, N. A 16 × 16 Pixel Post-Processing Free Quantum Random Number Generator Based on SPADs. *IEEE Trans. Circuits Syst. II Express Briefs* **2018**, *65*, 627–631. [CrossRef]
82. Hasan, R.S.; Tawfeeq, S.K.; Mohammed, N.Q.; Khaleel, A.I. A true random number generator based on the photon arrival time registered in a coincidence window between two single-photon counting modules. *Chin. J. Phys.* **2018**, *56*, 385–391. [CrossRef]
83. Shaw, G.; Sivaram, S.R.; Prabhakar, A. Quantum Random Number Generator with One and Two Entropy Sources. In Proceedings of the 2019 National Conference on Communications (NCC), Bangalore, India, 20–23 February 2019; pp. 1–4. [CrossRef]
84. Septriani, B.; de Vries, O.; Gräfe, M. Quantum random number generation (QRNG) by phase diffusion process in a gain-switched semiconductor laser—New insights. In Proceedings of the Conference on Lasers and Electro-Optics, San Jose, CA, USA, 5–10 May 2019. [CrossRef]
85. Tontini, A.; Gasparini, L.; Massari, N.; Passerone, R. SPAD-Based Quantum Random Number Generator with an Nth-Order Rank Algorithm on FPGA. *IEEE Trans. Circuits Syst. II Express Briefs* **2019**, *66*, 2067–2071. [CrossRef]
86. Nicola, M.; Gasparini, L.; Meneghetti, A.; Tomasi, A. A SPAD-based random number generator pixel based on the arrival time of photons. In Proceedings of the 2017 1st New Generation of CAS, NGCAS, Genova, Italy, 6–9 September 2017; pp. 213–216. [CrossRef]
87. Sarkar, A.; Chandrashekar, C.M. Multi-bit quantum random number generation from a single qubit quantum walk. *Sci. Rep.* **2019**, *9*, 12323. [CrossRef] [PubMed]

88. Li, Y.-H.; Han, X.; Cao, Y.; Yuan, X.; Li, Z.-P.; Guan, J.-Y.; Yin, J.; Zhang, Q.; Ma, X.; Peng, C.-Z.; et al. Quantum random number generation with uncharacterized laser and sunlight. *npj Quantum Inf.* **2019**, *5*, 97. [CrossRef]
89. McCabe, H.; Koziol, S.M.; Snider, G.L.; Blair, E.P. Tunable, Hardware-Based Quantum Random Number Generation Using Coupled Quantum Dots. *IEEE Trans. Nanotechnol.* **2020**, *19*, 292–296. [CrossRef]
90. Imran, M.; Sorianello, V.; Fresi, F.; Potì, L.; Romagnoli, M. Quantum random number generator based on phase diffusion in lasers using an on-chip tunable soi unbalanced Mach-Zehnder interferometer (uMZI). In Proceedings of the Optics InfoBase Conference Papers, Optical Fiber Communication Conference, San Diego, CA, USA, 8–12 March 2020. [CrossRef]
91. Sosa-Gómez, G.; Rojas, O.; Páez-Osuna, O. Using hadamard transform for cryptanalysis of pseudo-random generators in stream ciphers. *EAI Endorsed Trans. Energy Web* **2020**, *7*, e1. [CrossRef]
92. Nakov, S. Secure-Random-Generators. Available online: https://cryptobook.nakov.com/secure-random-generators (accessed on 3 September 2021).