Generation QI

Summer School

on Quantum Cryptography

Implementations of Quantum Cryptography

Contents

Chapter 1

Introduction to Quantum Optics

1.1 Theory

1.2 Assignments

Assignment 1.2.1

We have linear operators $\hat{a}$ and $\hat{a}^\dagger$ such that $[\hat{a}, \hat{a}^\dagger] = \hat{a}\hat{a}^\dagger - \hat{a}^\dagger\hat{a} = 1$. Check that

$$[\hat{a}^\dagger\hat{a}, \hat{a}] = -\hat{a}$$
$$[\hat{a}^\dagger\hat{a}, \hat{a}^\dagger] = \hat{a}^\dagger$$

Assignment 1.2.2

We define $\hat{N} = \hat{a}^\dagger\hat{a}$ (number operator). $\hat{N} = \hat{N}^\dagger$ and $\hat{N} = n\,|n\rangle$, where $n \in \mathbb{R}$. Show that eigenvalues of $\hat{N}$ are real, nonnegative numbers, assuming that eigenvectors are normalized ($||\,|n\rangle\,|| = 1$).

Assignment 1.2.3

Show that $|n+1\rangle$ and $|n-1\rangle$ are also eigenstates of $\hat{N}$.

Assignment 1.2.4

Proof that construction for obtaining all number operators' eigenstates is the following

$$|n\rangle = \frac{1}{\sqrt{n!}}(\hat{a}^\dagger)^n\,|0\rangle \ ,$$

where $|0\rangle$ is a vacuum state and $\hat{a}\,|0\rangle = 0$.

Assignment 1.2.5

Express coherent state in the photon number basis.

Hint

The coherent state is the eigenstate of the annihilation operator

$$\hat{a}\,|\alpha\rangle = \alpha\,|\alpha\rangle$$
$$\alpha \in \mathbb{C}$$

Assignment 1.2.6

Show that we have such a $|0\rangle$ that $\hat{a}\,|0\rangle = 0$.

Assignment 1.2.7

How does 2 mode state $|m, n\rangle$ look like after interfering with a beamsplitter?

Hint

The beamsplitter can be represented with the following unitary matrix

$$\hat{U}_{BS} = \begin{pmatrix} \cos\theta & e^{-i\phi}\sin\theta \\ -e^{i\phi}\sin\theta & \cos\theta \end{pmatrix}$$

The input-output relation is the following

$$\hat{a}_l^\dagger\,|0\rangle = \sum_m U_{ml}\hat{a}_m^\dagger\,|0\rangle$$

Assignment 1.2.8

Show how look the following two mode states: $|1, 0\rangle$, $|0, 1\rangle$, $|1, 1\rangle$, $|2, 0\rangle$, $|0, 2\rangle$.

Assignment 1.2.9

Show an optical implementation of $\sqrt{NOT}$ and Hadamard's gate.

Assignment 1.2.10

Check that the coherent state after passing through the balanced beam splitter remains the coherent state.

Assignment 1.2.11

Draw and Mach-Zehnder interferometer in the density matrix formalism.

Chapter 2

Practical Realizations of QKD Protocols

2.1 Theory

2.2 Assignments

Assignment 2.2.1

A good experimental practice is to be able to do a simple "back-of-the-envelope" calculation to do a quick estimate of what can be achieved in a certain system. The same is true for QKD. You are working with a commercial QKD system that is installed to distribute keys between two locations connected by approximately 40 kms of optical fiber. The system sends single-photons at a repetition rate of 1 MHz (assume a perfect source of Fock states with n = 1). The total optical misalignment including transmitter, receiver and channel disturbances amount to 1.5% on average. The total detection efficiency of the receiver hardware together with the single-photon detectors is 20% with a dark count rate of 20 counts/s. The optical fiber link has an attenuation of 13 dB.

a) Calculate the estimated quantum bit error rate (QBER) based on these numbers.

b) Recalculate now assuming that the system is used in to connect two other locations, which are now separated by 140 km (40 dB attenuation). All other parameters are unchanged. Why is the QBER much higher?

Assignment 2.2.2

BB84 and BBM92 are very similar protocols, especially if in BB84 the systems are prepared by the sender by means of measuring one photon from an entangled pair. Some security proofs for BB84 use this to simplify the problem to proving security of BBM92. This works well in the perfect case.

Assume that in BB84 the states are generated as stated above and two parties are at 140km distance. Compare BB84 and BBM92 protocols assuming the channel has no disturbance and attenuation of 0.3dB/km. In BBM92 the entanglement source is located exactly in the middle. Detectors are 20% efficient.

a) Assume detectors have no dark counts. What is the raw key rate for the two protocols if the states are generated at 10MHz?

b) Assume detectors have a dark count rate of 20 counts/s. What is the QBER of both protocols?

Assignment 2.2.3

Let's keep comparing BB84 and BBM92. Assume $g^{(2)} = 0.1$ meaning that the source produces 2 identical states instead of a single one with probability 0.1. Parameter $g^{(2)}$ is in practice nonzero even for so-called single photon sources. Assume that the probability to produce more than two states is 0.

To estimate the level of security the users have to compute the eavesdropper information about the key defined as $I = \min\{I(A : E), I(B : E)\}$. Assume that the eavesdropper can check how many photons are in the channel. If there are 2, she keeps one of them. Find I for both protocols if

a) Eavesdropper has quantum memory and she can wait for basis announcement before the measurement.

b) She has no quantum memory and has to measure immediately after photon capture.

Assignment 2.2.4

Most of the results for QKD were derived in the asymptotic limit. In practice the number of rounds is always finite and effects come into play. Assume that the amount of information which leaks to the eavesdropper during error correction phase is equal to $h(QBER)$ per bit of the key. This is a reasonable and often taken assumption. Assume that when you estimate QBER you will add 3 standard deviations to your estimate to make sure you did not underestimated the errors. If the sender in BB84 protocol sent 100.000 states and mean value of QBER is 5

What is the size of the final key after privacy amplification?

Hint

Remember that QBER needs to be computed separately for both bases.

Assignment 2.2.5

You are in possession of a new commercial QKD system which employs the BB84 protocol. You perform a characterization of the states produced by Alice with your own lab equipment, and you come to the conclusion that the four states produced are as follows:

$$\rho_H = \begin{bmatrix} 0.996 & 0.052 \\ 0.052 & 0.003 \end{bmatrix} \rho_V = \begin{bmatrix} 0.038 & -0.193 \\ 0.193 & 0.962 \end{bmatrix} \rho_{45} = \begin{bmatrix} 0.445 & 0.496 \\ 0.496 & 0.553 \end{bmatrix} \rho_{-45} = \begin{bmatrix} 0.691 & -0.462 \\ -0.462 & 0.309 \end{bmatrix}$$

Assuming that the receiver performs ideal projective measurements, and that there are no other imperfections, calculate the upper limit for the expected average QBER for this transmitter.

Chapter 3

Attacks on QKD and Quantum Networks

3.1 Theory

3.1.1 Attacks on QKD

3.1.2 Attacks on Quantum Networks

Introduction [30]

- **Confidentiality**

  No information is leaked to unauthorized parties.

- **Integrity**

  Information is accurate and trustworthy.

- **Availability**

  Access to the information by authorized parties.

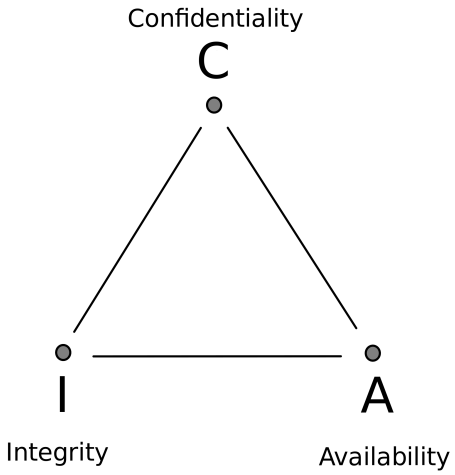

Figure 3.1: CIA Triad

Stage	**Application**
1. Trusted repeater	QKD without end-to-end security
2. Prepare and measure	QKD with end-to-end security
3. Entanglement generation	DI protocols
4. Quantum memory	Blind quantum computation
5. Few qubit fault tolerant	Distributed quantum computation
6. Quantum computing	Fast byzantine agreement

Table 3.1: Stages in the development of Quantum Internet. [33]

Type of nodes and links in Quantum Internet [30]

Types of quantum nodes

- End node (Enode 3.2)

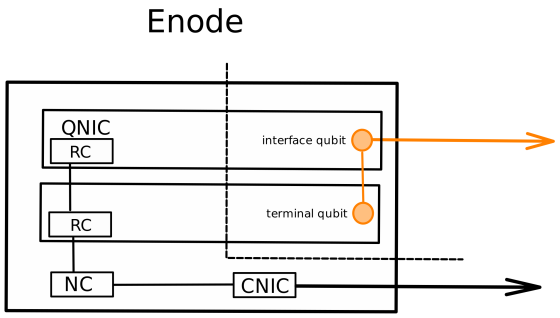

Figure 3.2: End node

- Measurement node (Mnode 3.3)

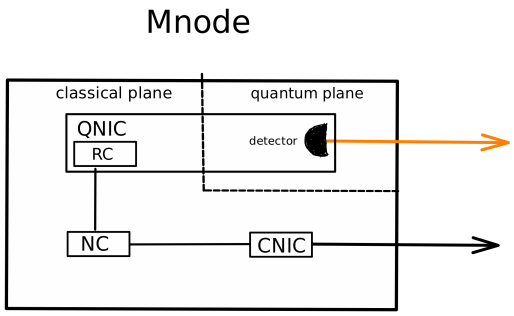

Figure 3.3: Measurement node

- Router node (Xnode 3.4)
- Intermediate node (Inode)
- Repeater node (Rnode 3.5)

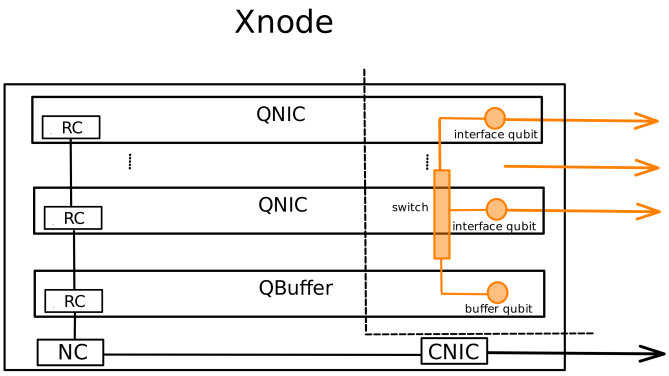

Figure 3.4: Router node

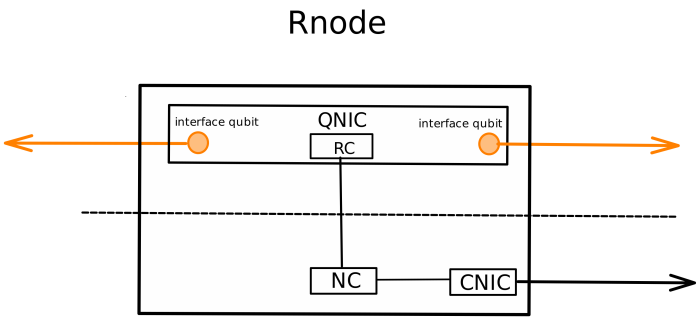

Figure 3.5: Repeater node

Attacks without control over quantum nodes [30, 20, 28]

Attacks with control over quantum nodes

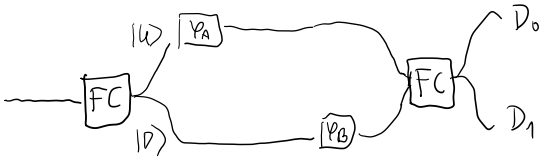

Figure 3.6: Weak coherent pulse is transformed by a fiber coupler (FC) with shape Y into up and down ways. Alice performs phase-shift by ϕ_A and Bob by ϕ_B. Then the signal is transformed by a fiber coupler with shape X and two its outcomes go to two detectors D0 and D1

3.2 Assignments

Assignment 3.2.1

Show that unitarity and linearity forbids cloning of unknown qubit state.

Assignment 3.2.2

In decoy Quantum key distribution by X-B. Wang, there are used 3 intensities: vacuum, μ and μ' (intensities are averages of the Poissonian distribution for each of the three weak coherent pulses). Assuming that (i) $\mu' > \mu$ and (ii) $\mu' e^{-\mu'} > \mu e^{-\mu}$, show that the state for intensity μ' reads

$$\rho_{\mu'} = e^{-\mu'} |0\rangle \langle 0| + \mu' e^{-\mu'} |1\rangle \langle 1| + c\frac{\mu'^2 e^{-\mu'}}{\mu^2 e^{-\mu}} \rho_c + d\rho_d \tag{3.1}$$

where $\rho_c := \frac{1}{c} \sum_{n=2}^{\infty} P_n(\mu) |n\rangle \langle n|$, with $P_n(\mu) = \frac{\mu^n e^{-\mu}}{n!}$ $c = 1 - e^{-\mu} - \mu e^{-\mu}$, $d = 1 - e^{-\mu'} + \mu' e^{-\mu'} + c\frac{\mu'^2 e^{-\mu'}}{\mu^2 e^{-\mu}}$, and $\rho_d \geq 0$ is some quantum state, which we don't have to specify. In the above $|0\rangle \langle 0|$ is the vacuum state, $|1\rangle \langle 1|$ is the single photon state and ρ_c is the multiphoton state i.e. signal with more than 1 photon, with intensity μ.

Hint

Observe first that owing to (i) and (ii) there is $P_n(\mu')/P_2(\mu') > P_n(\mu)/P_2(\mu)$.

Assignment 3.2.3

In the phase-encoding QKD protocol proposed by C. H. Bennett in 1992, the enconding by phases ϕ_A and ϕ_B are as on Fig. 3.6. Verify that the probability that detector D_0 clicks equals $cos^2(\frac{\phi_B - \phi_A}{2})$ and that D_1 clicks equals $sin^2(\frac{\phi_A - \phi_B}{2})$.

1. Hint : before entering the second fiber coupler (X), the state of the system is $e^{i\phi_A} |U\rangle + e^{i\phi_B} |D\rangle$. You can take out $e^{i\phi_A}$ in form of a global phase and neglect it.

2. Hint: apply the rule of turning into a superposition for each of the two states of the input superposition $(|U\rangle, |D\rangle)$ of the X coupler into $\frac{1}{\sqrt{2}}(|D_0\rangle + |D_1\rangle)$ and $\frac{1}{\sqrt{2}}(|D_0\rangle - |D_1\rangle)$

3. Hint: compute the absolute value of $1 \pm e^{i(\phi_B - \phi_A)}$ using Euler formula $e^{ix} = \cos x + i \sin x$ and $1 + \cos x = 2cos^2(x/2)$ and $\sin x = 2\sin(x/2)\cos(x/2)$. Furthermore $|a + ib| = \sqrt{a^2 + b^2}$.

Assignment 3.2.4 (Makarov's attack)

Alice uses bases $\{|+\rangle, |-\rangle\}$ and $\{|R\rangle, |L\rangle\}$ for encoding in BB84 protocol. Assuming Eve performs Makarov's attack on Bob's detectors fill the last column of the following table with the outcomes: **deterministic**, **no-click**, **discarded**.

Alice state	Eve's measurement	Bob's measurement	Outcome
$\|+\rangle_0$	$\|R\rangle, \|L\rangle$	$\|R\rangle, \|L\rangle$	
		$\|+\rangle, \|-\rangle$	
	$\|+\rangle, \|-\rangle$	$\|R\rangle, \|L\rangle$	
		$\|+\rangle, \|-\rangle$	
$\|-\rangle_1$	$\|R\rangle, \|L\rangle$	$\|R\rangle, \|L\rangle$	
		$\|+\rangle, \|-\rangle$	
	$\|+\rangle, \|-\rangle$	$\|R\rangle, \|L\rangle$	
		$\|+\rangle, \|-\rangle$	
$\|R\rangle_0$	$\|R\rangle, \|L\rangle$	$\|R\rangle, \|L\rangle$	
		$\|+\rangle, \|-\rangle$	
	$\|+\rangle, \|-\rangle$	$\|R\rangle, \|L\rangle$	
		$\|+\rangle, \|-\rangle$	
$\|L\rangle_1$	$\|R\rangle, \|L\rangle$	$\|R\rangle, \|L\rangle$	
		$\|+\rangle, \|-\rangle$	
	$\|+\rangle, \|-\rangle$	$\|R\rangle, \|L\rangle$	
		$\|+\rangle, \|-\rangle$	

Assignment 3.2.5 (Dead time detector's attack)

Eve attacks based on the dead time of detectors, by shining laser into Bob's device. List which triples of detectors will have dead time if after her pulse, given she encoded the pulse has polarization H, V, $(+)$ and $(-)$ corresponding to states $|0\rangle, |1\rangle$ and $|+\rangle, |-\rangle$. Find out the probability that Eve will force Bob's device to register the bit she knows given the four states are sent with probability 1/4 each from Alice's device.

Chapter 4

Quantum Secrecy Beyond QKD, Post-Quantum Cryptography

4.1 Theory

4.2 Assignments

Assignment 4.2.1

Shor's paper on factorization has the following passage in Section 5:

> To find a factor of an odd number n, given a method of computing the order r of x, choose a random $x \pmod n$, find its order r and compute $\gcd(x^{r/2} - 1, n)$.

In this exercise we will explore the reduction of order finding to factorization.

Recall that the order of an element x modulo n is the smallest positive integer r such that $x^r \equiv 1 \pmod n$.

- Write a script that finds (or compute by hand) orders of a few elements in $\mathbb{Z}_{21}^*$. (Note that elements of $\mathbb{Z}_{21}^*$ are those integers that are coprime with 21). Order of which element can be used to find a factor of 21?
- Factor RSA modulus $N = 4133053$ using the following information $ord_N(3) = 1032247$, $ord_N(5) = 2064494$.
- (extra) Looking at the expression $gcd(x^{r/2} - 1, n)$ one may notice that the term $x^{r/2}$ may be an extremely large number, i.e. $2^{r/2}$ has r binary digits and r may be close to n. It will be infeasible to compute $gcd(x^{r/2} - 1, n)$ directly when n is larger than a few milions. What is the way out of this difficulty?

Hint: If you use `python`, function `pow(x,r,n)` that computes $x^r \pmod n$ using a square-and-multiply method that does not produce very large intermediate values may turn out to be handy.

Assignment 4.2.2

A digital signature *forgery* is a pair consisting of a message m and an associated signature s that was not produced using the original private signing key and yet successfully passes the signature verification process.

Consider the following "simplification" of the Winternitz one-time signature scheme meant to hash fixed size messages m of 32-bits. Let h be a cryptographic hash function.

Key Generation: The private key is generated as random bit strings $x_i \leftarrow \mathsf{RND}()$ for $i = 0, \ldots, 3$. The public key: $y_i = h^{256}(x_i)$, $Y = h(y_0||y_1||y_2||y_3)$.

Signing: 32-bit message m is split into bytes $m = m_0||m_1||m_2||m_3$ and the signature $s = s_0||s_1||s_2||s_3$ is computed as $s_i = f^{m_i}(x_i)$.

Verification: Given the message m, its signature s and the public key Y compute $v_i = f^{256-m_i}(s_i)$, accept the signature if $Y = h(v_0||v_1|v_2||v_3)$.

Show how to find forgeries for this signature scheme. How would you fix that algorithm?

Assignment 4.2.3

Consider hash-based signature schemes that are based on one-time signatures combined with Merkle trees. When generating a key pair, one needs to generate secret values for one-time signatures in the leaf nodes of the Merkle tree. We could do this by generating random, uniformly distributed strings, but then storing all those secrets (e.g. 2^{16}) would make the total private key impractically large ($2^{16} \cdot 256$ bits). Can you think of a better way of generating those keys so that the private key state can be e.g. 256 bits?

Assignment 4.2.4

Consider the original McElice cryptosystem. It can be briefly described as follows.

Key generation: The private key consists of $n \times n$ permutation matrix P, a nonsingular $k \times k$ matrix S and a generator matrix G of a Goppa code over $\mathbb{F}_2^d$ with an irreducible polynomial $g \in \mathbb{F}_{2^d}[x]$ of degree t. G is of dimension $k = n - td$ where code length $n = 2^d$ and t is the number of correctable errors.

Encryption: To compute the encryption of a message m of length k one computes $m \cdot S \cdot G \cdot P$ and adds a random error vector e of weight t and length n: $y = m \cdot S \cdot G \cdot P + e$.

Decryption: Using the private key S, G, P we can compute $y \cdot P^{-1}$ which is equal to $m \cdot S \cdot G + e \cdot P^{-1}$. Since $m \cdot S \cdot G$ is a codeword and the permuted error vector $e \cdot P^{-1}$ still has weight t, we can use decoding algorithm to recover m.

Consider a concrete parameter set with $d = 10$ and $t = 40$ and therefore code length $n = 1024$ and dimension $k = 624$. Assume you received three encryptions y, y' and y'' of the same message m.

- Can you think of any regularity or relation between those ciphertexts?
- Show how to recover m based on y, y' and y'' and the public key.
- How would you prevent such attacks?

Assignment 4.2.5 (Anonymous entanglement)

1. Consider that Alice, Bob and Charlie share a GHZ state

$$|\psi\rangle_{ABC} = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle).$$

   If Charlie makes a measurement in the X-basis (i.e., the $\{|+\rangle, |-\rangle\}$ basis) what is the resulting state of Alice and Bob if Charlie obtains outcome $+$? And what if he obtains outcome $-$?

2. Now consider an N-partite GHZ state. If $N-k$-parties make a measurement of the X-basis. What is the resulting k-partite state as a function of the outcomes of the $N-k$ parties?

Assignment 4.2.6 (Noisy network)

Consider a network consisting of 4 nodes and a server, i.e., a central source that distributes a GHZ state to the nodes. Let us consider that during the distribution, the state undergoes noise that can be modeled by a global depolarizing channel, i.e. the state distributed to the parties is of the form

$$\rho = (1-p)\,|GHZ_4\rangle\langle GHZ_4| + p\frac{\mathbb{I}}{16}.$$

where $|GHZ_4\rangle = \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle)$ and p is the noise parameter. What is the fidelity of the bipartite anonymous entanglement that can be created in this network? What is the threshold value of p such that the state is still useful for teleportation?

Hint

A two-qubit state is useful for teleportation if it has singlet fraction $> \frac{1}{2}$.

Chapter 5

Private randomness

5.1 Theory

5.1.1 Introduction and basic concepts

Importance of randomness

Randomness is an essential concept in many sciences[5]

1. Philosophy
2. Mathematics
3. Computer Science
4. Physics
5. Technology

Types of randomness

We can distinguish two fundamental types of randomness[5]

1. Epistemic (apparent) randomness - lack of full knowledge of the system or its dynamic.
2. Ontic (intrinsic, inherent) randomness - true randomness that occurs even if we have full knowledge and unbounded computational abilities.

Ontic randomness is absent in the classical world and can only arise as a result of quantum effects. Some authors even connect ontic randomness with the notion of "free will". **Applications of randomness**

There are countless examples of randomness applications such as

1. Cryptography - there is no cryptography without randomness (for counterargument see [2])
2. Randomized algorithms - the quality of randomness can influence both the results and the complexity.
3. (Partially-)Random games (casinos, lotteries, stock markets) - can affect your wallet
4. and many, many others...

Random number generators

When working with randomness in practice in most cases we have some abstract device or process that generates a sequence of numbers. We call that device a Random Number Generator (RNG) or source of randomness. It is important to remember that generally in applications random refers to devices, not numbers. In most cases, we can't say that the number is random.

There is no such thing as a random number – there are only methods to produce random numbers

— John von Neumann[32]

Pseudorandom number generators

Pseudorandom number generators (PRNG) are algorithms that in a fully deterministic way produce a sequence of numbers from some input seed. The sequence should "look" like one produced by true RNG. The seed used can be fully random. It should be "hard" to determine the next bits or the seed from previously generated bits. PRNG has multiple applications in classical cryptography. For example when combining private key protocols with public key protocols.

Basic pseudorandom number generator example

One of the most basic types of PRNG is a linear congruential generator. We start with the seed X_0. Then in each step, we calculate a new value X_{i+1} according to the recursive formula

$$X_{i+1} = (aX_i + c) \mod m$$

where a, b, and m are some fixed generator parameters. Then to construct a pseudo-random string part of the bits from each X_i are used. For example in the GCC compiler rand() function (glibc-2.26 library) $m = 2^{31}$, $a = 1103515245$, $c = 12345$, and bits 30 to 0 are used.

It is very important to remember that such types of PRNG are not suitable for cryptographic applications (and most basic ones implemented in programming libraries) so specialized more

secure and complicated PRNG should be used.

Linear congruential generator with parameters used in different software and compiler: https://en.wikipedia.org/wiki/Linear_congruential_generator.

List of pseudo-random number generators: https://en.wikipedia.org/wiki/List_of_random_number_generators.

5.1.2 Definitions

Weak source of randomness

(a) SV-Source

Definition 1 (ϵ-SV-source) *A Santha-Vazirani Source (ϵ-SV Source) is the source that produces a sequence of binary numbers $(x_0, x_1, \dots)$ that comes from the distribution described by binary random variables $(X_0, X_1, \dots)$ that fulfills*

$$\forall_{n\in\mathbb{N}} \forall_{x_0,\dots,x_n\in\{0,1\}} \frac{1}{2} - \epsilon \leq P(X_n = x_n | X_{n-1} = x_{n-1}, \dots, X_0 = x_0, E) \leq \frac{1}{2} + \epsilon$$

where E represents all past random variables that can influence the source (random variables in past light cone).

For $\epsilon = 0$, we have a fully random source and for $\epsilon = 1/2$ it can even be completely deterministic.

SV-Source examples We can see some simple examples of the SV Source realization:

- Repeated flipping a biased coin.
- Flipping sequence of the biased coin with a different bias.
- Flipping biased coin in such a way that the bias of the coin depends on the outcomes of previous throws.

(b) Min-entropy source

We can also define a weak source of randomness in different ways[7].

Definition 2 (Min-entropy source) *A min-entropy Source is a single shot source that produces a single sequence of n binary numbers $(x_0, \dots, x_n)$ that comes from the distribution described by n-bit random variable X. A source is called (n, k)-min-entropy source if*

$H_\infty(X|E) \geq k$ *where*

$$H_{\infty(X|E)} = -\log_2 \max_{X \in \{0,1\}^n} P(X|E)$$

We can also say that it has a min-entropy rate $R = k/n$.

(b) Block min-entropy source

We can also consider a block min-entropy source that produces consecutive n-bit blocks. The formulation then is similar to the SV-source.

Definition 3 (Block min-entropy source) *A Block min-entropy Source is a source that produces a sequence of consecutive n-bit numbers* $(x_0, x_1, \dots)$ *that comes from the distribution described by n-bit random variables* $X_1, X_2, \dots$. *A source is called block* (n, k)*-min-entropy source if*

$$\forall_{m \in \mathbb{N}} \forall_{x_0, \dots, x_m \in \{0,1\}^n} H_\infty(X_m = x_m | X_{m-1} = x_{m-1}, \dots, X_0 = x_0, E) \geq k.$$

It can be shown that the assumptions on the block min-entropy source are weaker than the ones on the SV source. In other words, SV sources have more structure.

Examples of potential sources of weak randomness

There are many candidates for weak sources of randomness including:

- imperfect, noisy quantum RNG
- Geiger counters,
- Zener diodes,
- human body (heart rate, electricity, brain waves)
- stock markets,
- natural events (weather),
- cosmic events (pulsars, background radiation).

5.1.3 Randomness Tests

How to check RNG?

The important question is can we check if (and how much) random an RNG is, based only on its outputs?

Standard randomness test suite

One of the most popular frameworks for testing RNGs and PRNGs is Statistical Test Suite by NIST[4]. We should remember that this test suite is not complete (as no other test set is). And as the authors state it is not sufficient for cryptographic purposes.

Standard randomness test suite

It is composed of 15 test groups.

1. The Frequency (Monobit) Test,
2. Frequency Test within a Block,
3. The Runs Test,
4. Tests for the Longest-Run-of-Ones in a Block,
5. The Binary Matrix Rank Test,
6. The Discrete Fourier Transform (Spectral) Test,
7. The Non-overlapping Template Matching Test,
8. The Overlapping Template Matching Test
9. Maurer's "Universal Statistical" Test,
10. The Linear Complexity Test,
11. The Serial Test,
12. The Approximate Entropy Test,
13. The Cumulative Sums (Cusums) Test,
14. The Random Excursions Test, and
15. The Random Excursions Variant Test.

SV-Test Idea

For the purpose of checking if human heart rate can be treated as a weak source of randomness for quantum randomness amplification protocols, we developed the so-called SV-Test[31]. The goal of this test is to estimate the ϵ parameter assuming that the binary string comes from some SV Source.

SV-Test Epsilon for single history length

For binary string $s = s_1, \ldots, s_n$ of length n produced by the ϵ-SV Source we can calculate series of ϵ parameters using formula

$$\epsilon_h = \max_{w_h} \left| \frac{|s|_{w_h}}{|s|_{w'_h}} - \frac{1}{2} \right|$$

where $w_h = (x_1, x_2, \ldots, x_h, x_{h+1})$ is binary sequence, $w'_h = (x_1, x_2, \ldots, x_h)$, and $|a|_b$ denotes number of occurrence of substring b in string a. Additionally, we define $\epsilon_0 = ||s|_0/n - 1/2|$.

SV-Test Combining epsilons

With all ϵ_h calculated, we have to find a way to combine them to obtain a single ϵ value that estimates the parameter of ϵ-SV Source. One possible way to achieve this is to use the following formula.

$$\epsilon = \frac{1}{w(\lfloor \log_2(n) \rfloor - 1)} \sum_{i=0}^{\lfloor \log_2(n) \rfloor - 1} \frac{\epsilon_i}{(i+1)}$$

with $w(h) = \sum_{i=0}^{h} \frac{1}{i+1}$. Why does the sum end? The other interesting formulas to obtain single epsilon will be part of future research.

SV-Test Derivation

$$\underset{n \in \mathbb{N}}{\forall} \underset{s_0, \ldots, s_{n+1} \in \{0,1\}}{\forall} \frac{1}{2} - \epsilon \leq P(S_{n+1} = s_{n+1} | S_n = s_n, \ldots, S_0 = s_0, E) \leq \frac{1}{2} + \epsilon$$

$$\underset{n \in \mathbb{N}}{\forall} \max_{s_0, \ldots, s_{n+1} \in \{0,1\}} \left| P(S_{n+1} = s_{n+1} | S_n = s_n, \ldots, S_0 = s_0) - \frac{1}{2} \right| \leq \epsilon$$

$$\varepsilon_h \coloneqq \max_{s_{n-h+1}, \ldots, s_{n+1} \in \{0,1\}} \left| \frac{P(s_{n+1}, s_n, s_{n-1}, \ldots, s_{n-h+1})}{P(s_n, s_{n-1}, \ldots, s_{n-h+1})} - \frac{1}{2} \right|.$$

$$\tilde{\varepsilon}_h \coloneqq \max_{w_h} \left| \frac{\frac{|s|_{w_h}}{n-h}}{\frac{|s|_{w'_h}}{n-h+1}} - \frac{1}{2} \right| \underset{n \to \infty}{\approx} \max_{w_h} \left| \frac{|s|_{w_h}}{|s|_{w'_h}} - \frac{1}{2} \right|$$

Classical randomness amplification

The important question is can we amplify the weak source of randomness? For example, can we take n bits from ϵ-SV source and use some deterministic function that produces $k \leq n$ bits in such a way that the resulting sequence is ϵ'-SV source with $\epsilon' < \epsilon$. Let's consider the case with a single biased coin where $P(H) = p, P(T) = 1 - p$, and $p \in (0, 1/2)$. Tossing this coin is a weak random source with $\epsilon = |p - 1/2|$. We can amplify randomness by collecting coin tosses into pairs and making new sequence in the following way:

$$\begin{cases} HH \mapsto - \\ HT \mapsto 0 \\ TH \mapsto 1 \\ TT \mapsto - \end{cases}$$

The important question is can we amplify the weak source of randomness? For example, can we take n bits from ϵ-SV source and use some deterministic function that produces $k \leq n$ bits in such a way that the resulting sequence is ϵ'-SV source with $\epsilon' < \epsilon$. Santha and Vazirani show that in general, that is not possible[29].

Theorem 1 (No-go theorem for randomness amplification) *Do not exist universal extractor that can amplify the randomness of a single SV-Source.*

It is however possible to perform amplification from two **independent** SV sources.

Amplification of two independent weak sources.

Let $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$ be to string from two **independent** ϵ-SV sources. Then we can obtain a single more random bit using extractor $Ex(X, Y)$ defined in the following way

$$Ex(X, Y) \coloneqq (x_1 \cdot y_1) \oplus_2 (x_2 \cdot y_2) \oplus_2 \cdots \oplus_2 (x_n \cdot y_n).$$

Repeating this process we will obtain new ϵ'-SV source with $\epsilon' < \epsilon$.

5.1.4 Quantum protocols

Quantum device-dependent RNG

In the quantum world, some measurements are random from the QM axioms. We can use it to create a "very simple" quantum RNG. For example, we can take quantum state $|+\rangle := 1/\sqrt{2}(|0\rangle + |1\rangle)$ and measure it in standard base $\{|0\rangle, |1\rangle\}$. The measurement result of this process is either 0 or 1 each with probability 1/2. We, therefore, created the perfect RNG. Is it really perfect and is that mean that we are done?

Quantum device-dependent RNG drawbacks

Problems with device-dependent quantum RNG

- We cannot create the perfect quantum state in the laboratory.
- We cannot perform perfect measurements.
- If we buy such quantum RNG it is almost impossible to very what the device really does.

Fortunately, we can partially solve those problems with a device-independent approach.

Quantum randomness protocols

There are a few classes of randomness quantum protocols including:

- randomness amplification,
- randomness expansion,
- randomness certification.

5.1.5 Quantum Randomness Amplification

Quantum randomness amplification general scheme 5.1

The first quantum randomness amplification protocol was invented by Colbeck and Renner[14]. The protocol is based on chain Bell inequality and works for $\epsilon < 0.058$.

Quantum randomness amplification protocol

Colbeck and Renner protocol[14]:

1. Repeat the following for all $q = 1, \ldots, M$: Invoke D_A and D_B with inputs $\alpha = \pi/(2N)A_q$ and $\beta = \pi/(2N)B_q$ respectively, where $A_q \in \{0, 2, \ldots, 2N-2\}$ and $B_q \in \{1, 3, \ldots, 2N-1\}$ are chosen at random, using bits from the sources S_A and S_B, respectively, and record the outcomes $X_q \in \{-1, 1\}$ and $Y_q \in \{-1, 1\}$, respectively.

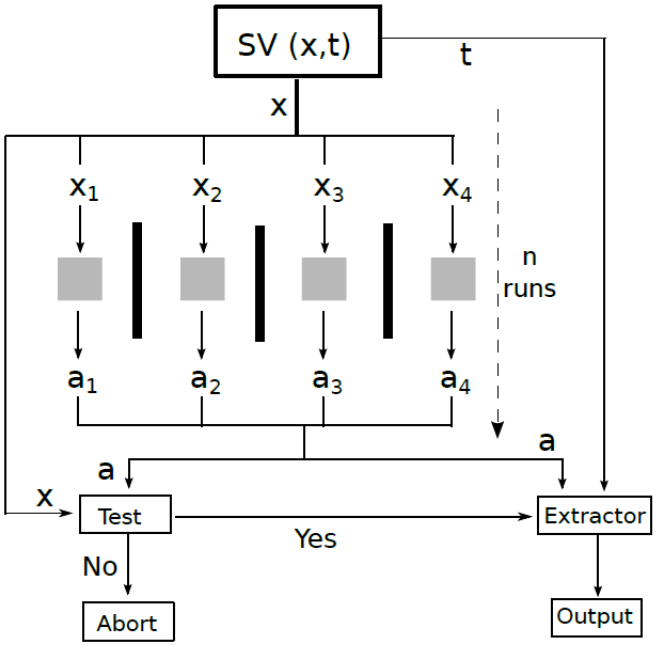

Figure 5.1: General scheme of most quantum randomness amplification protocol[5].

2. Define the set $T := \{q : |A_q - B_q| = 1 \text{ or } (A_q, B_q) = (0, 2N-1)\}$. Check that the cardinality $|T|$ of T satisfies $|T| \in [M/N, 3M/N]$. If this test fails then set R = FAIL and abort.

3. For each $q \in T$ check that $X_q = Y_q$ (if $(A_q, B_q) \neq (0, 2N-1)$) or $X_q \neq Y_q$ (otherwise). If for at least one $q \in T$ this test fails then set R = FAIL and abort.

4. Choose $f \in T$ at random, using bits from the source S_A, and output $R = X_f$.

Quantum randomness amplification protocols properties

In quantum randomness amplification protocols, there are a few features that we would like to have such as:

- type of the source (SV or min-entropy),
- security against quantum or non-signaling adversary,
- public or private source,
- composability,

There are also a few parameters that we want to improve such as:

- range of allowed weak source parameter (ϵ of ϵ-SV Source),

- number of parties (devices),
- source rate,
- quantum rate,
- amplification rate,
- noise tolerance (robustness),
- type of non-locality,
- amount of correlations between the weak, source, and quantum devices, and
- ¡10-¿ security against all or some family of attacks.

Quantum randomness amplification protocols

Since the first quantum randomness amplification protocol [14] many new ones were invented to obtain or improve features and parameters described in the last points [15, 16, 11] and more [23, 8, 26, 34], and a few more recent ones [19, 25].

5.1.6 Other protocols

Quantum randomness expansion

Quantum randomness expansion is the process in which we take some small strings of random bits (seed) and produce a longer string of random bits. These types of protocols are also based on non-local inequalities (in most cases Bell inequality). The first such protocols were developed by Colbeck et al.[13, 12]. Draft of randomness expansion protocol based on[12] and[13]:

1. The seed string X is divided into two strings x_1 and R of the same length.
2. Alice uses two bits from x_1 to choose the non-local test on GHZ state and collect results for the devices.
3. Alice checks appropriate inequality and aboard or build string x' form outputs.
4. She repeat steps 2 and 3 until whole string x is used and the string x' of the same length is created.
5. She performs privacy amplification using a $universal_2$ hash function, where the random string R is used to choose the hash function obtaining string s.

6. The concatenation of X and s is the result of the protocol.

Quantum randomness certification

In a non-locality scenario if we observe a violation of a Bell-like inequality then the outcomes are guaranteed to be random. This process of generating randomness is independent of the used device hence, this is a device-independent RNG. Obtained in such a manner random numbers could be certifiable, and private. We should, however, notice an important problem. Performing a Bell-like test requires random inputs (measurement settings). This leads to a contradiction! A possible, but vague solution is to assume that the experimentalists can make free choices. This, however, loops back to the question of randomness and free will.

Non-locality without inputs

A possible solution would be to find non-local scenarios that violate Bell-like inequality without inputs. In a recent paper[21] the authors try to certify non-locality without inputs in a "semi-device independent" scenario. Although, the concept itself is interesting, in my opinion, "semi-device independent" is an overstatement compared to previous semi-device independent protocols (see for example[24, 17]). Furthermore, this approach is totally unsuitable for cryptographic applications.

The Big Bell Test

A very interesting project called "The Big Bell Test"[1] that used human-generated randomness as inputs to the Bell experiments was performed by "The Big Bell Test collaboration" https://thebigbelltest.icfo.eu/. The experiment involved over 100 scientists from 15 research centers and over 100000 participants. The 90 million bits were processed.

Quantum randomness protocols as part of other protocols.

In many situations, it is possible to use quantum randomness protocols to prepare random input for other protocols, for example for QKD. Sometimes we can even use that kind of method to build protocols that already accept week random inputs instead of fully random ones.

5.1.7 Additional Topics and Conclusions

Real life commercial implementations

There are many technological approaches and platforms to realize different kinds of quantum RNG. For review see for example[22]. There are companies that sell QRNG. There are commercially available QRNG on-chip for internal applications in computer systems. This chip is also inside one model of a popular smartphone line. There is a website by The Australian National University that lives streams random numbers from QRNG (quantum randomness

beacon) https://qrng.anu.edu.au

Random numbers

Theorem 2 *Let $\alpha \in [0,1]$ be a randomly chosen real number. By random we mean hear sampled from a continuous uniform distribution. So this is a purely mathematical concept. Then with the probability one α is both a normal number and an uncomputable number.*

This means that the binary expansion of that number is a perfectly random sequence like one obtained from perfect RNG.

5.1.8 Conclusions

Other topics

List of some other connected topics that we do not have time to cover:

- Description of the test cases in the NIST test suite[4] (will be partially covered during Workshop), and other randomness tests[9, 6]
- More complicated classical randomness extractors, hash functions, and connected results like Leftover Hash Lemma (see [18, 27, 3, 10])
- Abstract quantum and other theory states that contains randomness.
- Randomness in networks.

Conclusions and take home messages

- Randomness is very important, both from fundamental and application points of view, in many fields of science and life.
- True randomness comes from quantum theory.
- There are many useful quantum randomness protocols.

5.2 Assignments

The goal of these exercises is to learn some basic concepts of statistical estimation of randomness for a given binary string produced by some random number generator. The first exercise goal is to estimate the ϵ parameter for the sequence produced by ϵ-Santha-Vazirani (SV) Source. It is based on the paper by Stankiewicz et al. [31]. All other exercises are based on Statistical Test Suite by NIST [4]. The idea is to learn how some of the test types from the package work and try them on some short examples. Some of these examples are the ones from the paper itself, and others are slightly modified ones. If you have any problems with the exercises or want to learn about more complicated test types, we advise you to consult the original report [4].

In some of the assignments, the final step requires using one of the following special, non-elementary functions.

Complementary error function:

$$\operatorname{erfc}(z) = \frac{2}{\sqrt{\pi}} \int_z^\infty e^{-u^2} \mathrm{d}u. \tag{5.1}$$

This function can be calculated numerically using for example www.wolframalpha.com using input erfc(x).

Incomplete gamma function:

$$\operatorname{igamc}(x, a) = \frac{1}{\Gamma(a)} \int_x^\infty e^{-t} t^{a-1} \mathrm{d}t, \tag{5.2}$$

where $\Gamma(a)$ is standard gamma function This function can be calculated numerically using for example www.wolframalpha.com using input Gamma(x, a)/Gamma(x).

Assignment 5.2.1 (Estimation of ϵ parameter for SV Source)

For binary string $s = s_1, \ldots, s_n$ of length n produced by the ϵ-SV Source we can calculate series ϵ parameters using formula

$$\epsilon_h = \max_{w_h} \left| \frac{|s|_{w_h}}{|s|_{w'_h}} - \frac{1}{2} \right| \tag{5.3}$$

where $w_h = (x_1, x_2, \ldots, x_h, x_{h+1})$ is binary sequence, $w'_h = (x_1, x_2, \ldots, x_h)$, and $|a|_b$ denotes number of occurrence of substring b in string a. Additionally, we define $\epsilon_0 = ||s|_0/n - 1/2|$.

Task 1: Try to argue why for the string s of length n it is enough to consider only ϵ_h for $h \leq log_2(n) - 1$.

Task 2: According to the above formula calculate appropriate ϵ_h for the sequences

- 011001

- 00011011
- 0000100110101111000

With all ϵ_h calculated, we have to find a way to combine them to obtain a single ϵ value that estimates the parameter of ϵ-SV Source. One possible way to achieve this is to use the following formula.

$$\epsilon = \frac{1}{w(\lfloor \log_2(n) \rfloor - 1)} \sum_{i=0}^{\lfloor \log_2(n) \rfloor - 1} \frac{\epsilon_i}{(i+1)} \tag{5.4}$$

with $w(h) = \sum_{i=0}^{h} \frac{1}{i+1}$.

Task 3: Calculate the final epsilon for the examples from Task 2.

Assignment 5.2.2 (The Frequency Test)

The Frequency (Monobit) Test is the most basic test in the NIST suite. Its goal is to check the proportion of zeros and one in the sequence. As you can guess in the truly random sequence the numbers of zeros and ones should be quite similar. It is worth noting, that according to the NIST standard, if this test fails then further tests should not be applied and sequence/RNG have to be discarded. The test works as follows.

1. For binary input sequence $s = s_1, \ldots, s_n$ calculate the value $S_n = \sum_{i=1}^{n}(2s_i - 1)$ that is the number of ones minus the number of zeroes in the sequence.
2. Calculate the test statistic $s_{\text{obs}} = |S_n|/\sqrt{n}$.
3. Compute the P-value that equals $\text{erfc}(s_{\text{obs}}/\sqrt{2})$.
4. If the P-value is smaller than 0.01 the test fails otherwise it passes.

Task 1: Perform frequency test for the sequences

- 1011010101
- 1100100100001111110110101010001000100001011010001100001000110100110001001100011001100010100001011100
- 1111110001000100000100101010001001100011001011011000010001101001110011010001000101000101100001110010

Assignment 5.2.3 (Frequency Test within a Block)

The frequency test with a block is a generalization of the previous test. We also are interested in the proportion of ones and zeroes but not in the whole sequence but in some shorter blocks of size M. The test works as follows.

1. Divide the binary input sequence $s = s_1, \ldots, s_n$ into $N = \lfloor n/M \rfloor$ segments of size M.
2. Calculate the proportion of π_i of ones in each M-bit block for $1 \leq i \leq N$ using formula $\pi_i = (\sum_{j=1}^{M} s_{(i-1)M+j})/M$.
3. Calculate the test statistic $s_{\text{obs}} = 4M \sum_{i=1}^{N} (\pi_i - 1/2)^2$.
4. Compute the P-value that equals igamc$(N/2, s_{\text{obs}}/2)$.
5. If the P-value is smaller than 0.01 the test fails otherwise it passes.

Task 1: Perform frequency test within a block for the sequences

- 0110011010, M = 3
- 11001001000011111101101010100010001000010110100011
  00001000110100110001001100011001100010100010111000, M = 10

Assignment 5.2.4 (The Runs Test)

The run test checks if oscillations between ones and zeroes are correct for random sequences. The run is an uninterrupted sequence of the same bit in the sequence. A run of length k is zero followed by k ones fo.lowed by zero or one followed by k zeros followed by one,

1. Calculate the proportion π of ones in the sequence $s = s_1, \ldots, s_n$ using formula $\pi = (\sum_{i=1}^{n} s_i)/n$.
2. Calculate the test statistic $s_{\text{obs}} = 1 + \sum_{i=1}^{n-1} r_k$, where $r_k = \begin{cases} 0 : s_k = s_{k+1} \\ 1 : s_k \neq s_{k+1} \end{cases}$.
3. Compute the P-value that equals erfc$((s_{\text{obs}} - 2n\pi(1-\pi))/(2\sqrt{2n}\pi(1-\pi)))$.
4. If the P-value is smaller than 0.01 the test fails otherwise it passes.

Task 1: Perform the run test for the sequences

- 1001101011
- 1100100100001111110110101010001000100001011010001100001000110100110001001100011001100010100001011100

Assignment 5.2.5 (Tests for the Longest-Run-of-Ones in a Block)

The test, as the name suggests, checks the number of consecutive ones in the block of size M. The test has a few versions that depend on the length of the sequence. We will focus on the most basic one that is applicable if $n \geq 128$. In this version $M = 8$ and we have two additional parameters $K = 3$ and $N = 16$. For the definition of the other version and its parameters see [4].

1. Divide the binary input sequence $s = s_1, \ldots, s_n$ into segments of size M.
2. Calculate the values v where v_0 is the number of blocks with the longest run of ones that have length 0 or 1, v_1 is the number of blocks with the longest run of ones that have length 2, v_2 is the number of blocks with the longest run of ones that have length 3, and v_4 is the number of blocks with the longest run of ones that have length 4 or more.
3. Calculate the test statistic $s_{\text{obs}} = \sum_{i=0}^{K}((v_i - N\pi_i)^2/(N\pi_i))$, where $\pi_0 = 0.2148$, $\pi_1 = 0.3672$, $\pi_2 = 0.2305$, and $\pi_3 = 0.1875$.
4. Compute the P-value that equals $\text{igamc}(K/2, s_{\text{obs}}/2)$.
5. If the P-value is smaller than 0.01 the test fails otherwise it passes.

Task 1: Perform the tests for the longest-run-of-ones in a block for the sequence

- 1100110000010101011011000100110011110000000000001001001101010100010001001111010110100000001101011111001100111001101101100010110010

Assignment 5.2.6 (Binary Matrix Rank Test)

The binary matrix rank test checks linear dependence between fixed-length blocks of the sequence. To do this we will write sub-strings in the form of the matrices and calculate their rank. In real-life applications, matrices of the size 32×32 are the most commonly used. Here, for the simplicity of the exercise, we will only use matrices of the size 3×3. For parameters and description of the test in bigger cases consult [4].

1. Divide the binary input sequence $s = s_1, \ldots, s_n$ into segments of size $M = 3 \times 3 = 9$.

2. From each segment construct matrix of size 3×3 placing the first three bits from the block into the first row of the matrix, bits 4 to 6 into the second row, and the last 3 bits into the last row. We will obtain $N = \lfloor n/9 \rfloor$ matrices.

3. Compute the (binary) rank of each matrix.

4. Calculate the test statistic $s_{\text{obs}} = (F_3 - 0.2888N)^2/(0.2888N) + (F_2 - 0.5776N)^2/(0.5776N) + (F_{1,0} - 0.1336N)^2/(0.1336N)$, where F_3 is number of matrices with rank 3, F_2 is number of matrices with rank 2, and $F_{1,0}$ is number of matrices with rank 1 or 0.

5. Compute the P-value that equals $\text{igamc}(1, s_{\text{obs}}/2)$.

6. If the P-value is smaller than 0.01 the test fails otherwise it passes.

Task 1: Perform the binary matrix rank test for the sequence

- 01011001001010101101

Assignment 5.2.7 (Serial Test)

The serial test in its core is quite similar to the estimation of the ϵ parameter for SV Source described in the first assignment of this section. The main differences are that here the cyclic approach is used and pure frequencies (not conditional ones) are used. Because of this similarity and quiet "messy" formulas used in the test, we will not give any tasks in this assignment. Finally, also the approximate entropy test (that we will omit) is in some ways related to the estimation of the ϵ parameter for SV Source.

Assignment 5.2.8 (Cumulative Sums Test)

The cumulative sums (cusum) test examines the sequence of partial sums generated from the original sequence. It can be seen as checking the maximal excursion of the random walk constructed from the original tested sequence.

1. Change the binary input sequence $s = s_1, \ldots, s_n$ into normalized sequence $x = x_1, \ldots, x_n$ using the formula $x_i = 2s_i + 1$.

2. Calculate sequence $S = S_1, \ldots, S_n$ of partial sum where $S_i = \sum_{j=1}^{i} x_i$.

3. Calculate the test statistic $z = \max_{1 \leq i \leq n} |S_i|$.

4. Compute the P-value that equals

$$P = 1 - \sum_{k=\frac{\frac{-n}{z}+1}{4}}^{\frac{\frac{n}{z}-1}{4}} \left(\Phi\left(\frac{(4k+1)z}{\sqrt{n}} \right) + \Phi\left(\frac{(4k-1)z}{\sqrt{n}} \right) \right) + \sum_{k=\frac{\frac{-n}{z}-3}{4}}^{\frac{\frac{n}{z}-1}{4}} \left(\Phi\left(\frac{(4k+3)z}{\sqrt{n}} \right) + \Phi\left(\frac{(4k+1)z}{\sqrt{n}} \right) \right) \tag{5.5}$$

   where Φ is the Standard Normal Cumulative Probability Distribution Function.

5. If the P-value is smaller than 0.01 the test fails otherwise it passes.

Task 1: Perform the Cumulative Sums Test for the sequence for the sequence

- 1011010111
- 1100100100001111110110101010001000100001011010001100001000110100110001001100011001100010100001011100 0

Finally Remarks: We omitted some tests that have more complicated descriptions or are not suitable to perform on the piece of paper. We advise interested readers to read the original paper [4] and also try to perform all of the tests on much longer sequences using dedicated software.

Bibliography

[1] C. Abellán and et.al. Challenging local realism with human choices. *Nature*, 557(7704):212–216, May 2018.

[2] E. A. Aguilar, R. Ramanathan, J. Kofler, and M. Pawłowski. Completely device-independent quantum key distribution. *Physical Review A*, 94(2):022305, Aug. 2016.

[3] R. Arnon-Friedman, C. Portmann, and V. B. Scholz. Quantum-Proof Multi-Source Randomness Extractors in the Markov Model. In A. Broadbent, editor, *11th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2016)*, volume 61 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:34, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[4] L. E. Bassham, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks, N. A. Heckert, J. F. Dray, and S. Vo. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical Report 800-22, NIST, 2010.

[5] M. N. Bera, A. Acín, M. Kuś, M. W. Mitchell, and M. Lewenstein. Randomness in quantum mechanics: philosophy, physics and technology. *Reports on Progress in Physics*, 80(12):124001, Nov. 2017.

[6] M. É. Borel. Les probabilités dénombrables et leurs applications arithmétiques. *Rendiconti del Circolo Matematico di Palermo*, 27(1):247–271, Dec. 1909.

[7] J. Bouda, M. Pawłowski, M. Pivoluska, and M. Plesch. Device-independent randomness extraction from an arbitrarily weak min-entropy source. *Physical Review A*, 90(3), Sept. 2014.

[8] F. G. S. L. Brandão, R. Ramanathan, A. Grudka, K. Horodecki, M. Horodecki, P. Horodecki, T. Szarek, and H. Wojewódka. Realistic noise-tolerant randomness amplification using finite number of devices. *Nature Communications*, 7(1):11345, Apr. 2016.

[9] R. G. Brown, D. Eddelbuettel, and D. Bauer. Dieharder : A random number test suite, 2020.

[10] Chattopadhyay and Zuckerman. Explicit two-source extractors and resilient functions. *Annals of Mathematics*, 189(3):653, 2019.

[11] K.-M. Chung, Y. Shi, and X. Wu. Physical randomness extractors: Generating random numbers with minimal assumptions, 2014.

[12] R. Colbeck. *Quantum And Relativistic Protocols For Secure Multi-Party Computation*. PhD thesis, University of Cambridge, 2006.

[13] R. Colbeck and A. Kent. Private randomness expansion with untrusted devices. *Journal of Physics A: Mathematical and Theoretical*, 44(9):095305, Feb. 2011.

[14] R. Colbeck and R. Renner. Free randomness can be amplified. *Nature Physics*, 8(6):450–453, May 2012.

[15] R. Gallego, L. Masanes, G. D. L. Torre, C. Dhara, L. Aolita, and A. Acín. Full randomness from arbitrarily deterministic events. *Nature Communications*, 4(1):3654, Oct. 2013.

[16] A. Grudka, K. Horodecki, M. Horodecki, P. Horodecki, M. Pawłowski, and R. Ramanathan. Free randomness amplification using bipartite chain correlations. *Physical Review A*, 90(3):032322, Sept. 2014.

[17] K. Horodecki and M. Stankiewicz. Semi-device-independent quantum money. *New Journal of Physics*, 22(2):023007, Feb. 2020.

[18] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing - STOC '89*. ACM Press, 1989.

[19] M. Kessler and R. Arnon-Friedman. Device-independent randomness amplification and privatization. *IEEE Journal on Selected Areas in Information Theory*, 1(2):568–584, Aug. 2020.

[20] L. Lydersen, C. Wiechers, C. Wittmann, D. Elser, J. Skaar, and V. Makarov. Hacking commercial quantum cryptography systems by tailored bright illumination. *Nature Photonics*, 4(10):686–689, aug 2010.

[21] Z. Ma, M. Rambach, K. Goswami, S. S. Bhattacharya, M. Banik, and J. Romero. Randomness-free test of non-classicality: a proof of concept, 2023.

[22] V. Mannalath, S. Mishra, and A. Pathak. A comprehensive review of quantum random number generators: Concepts, classification and the origin of randomness, 2022.

[23] P. Mironowicz, R. Gallego, and M. Pawłowski. Robust amplification of santha-vazirani sources with three devices. *Physical Review A*, 91(3):032317, Mar. 2015.

[24] M. Pawłowski and N. Brunner. Semi-device-independent security of one-way quantum key distribution. *Physical Review A*, 84(1), July 2011.

[25] R. Ramanathan. Finite device-independent extraction of a block min-entropy source against quantum adversaries, 2023.

[26] R. Ramanathan, F. G. Brandão, K. Horodecki, M. Horodecki, P. Horodecki, and H. Wojewódka. Randomness amplification under minimal fundamental assumptions on the devices. *Physical Review Letters*, 117(23):230501, Nov. 2016.

[27] R. Raz. Extractors with weak random seeds. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing - STOC '05*. ACM Press, 2005.

[28] O. Sakarya, M. Winczewski, A. Rutkowski, and K. Horodecki. Hybrid quantum network design against unauthorized secret-key generation, and its memory cost. *Physical Review Research*, 2(4):043022, oct 2020.

[29] M. Santha and U. V. Vazirani. Generating quasi-random sequences from semi-random sources. *Journal of Computer and System Sciences*, 33(1):75–87, 1986.

[30] T. Satoh, S. Nagayama, S. Suzuki, T. Matsuo, M. Hajdusek, and R. V. Meter. Attacking the quantum internet. *IEEE Transactions on Quantum Engineering*, 2:1–17, 2021.

[31] M. Stankiewicz, K. Horodecki, O. Sakarya, and D. Makowiec. Private weakly-random sequences from human heart rate for quantum amplification. *Entropy*, 23(9), 2021.

[32] J. von Neumann. Various techniques used in connection with random digits. In A. Householder, G. Forsythe, and H. Germond, editors, *Monte Carlo Method*, pages 36–38. National Bureau of Standards Applied Mathematics Series, 12, Washington, D.C.: U.S. Government Printing Office, 1951.

[33] S. Wehner, D. Elkouss, and R. Hanson. Quantum internet: A vision for the road ahead. *Science*, 362(6412), 10 2018.

[34] H. Wojewódka, F. G. S. L. Brandão, A. Grudka, K. Horodecki, M. Horodecki, P. Horodecki, M. Pawłowski, R. Ramanathan, and M. Stankiewicz. Amplifying the randomness of weak sources correlated with devices. *IEEE Transactions on Information Theory*, 63(11):7592–7611, Nov. 2017.