Randomness Certification

It turns out that Bell tests also provide a useful primitive in cryptographic settings, where we may be very paranoid about being fooled. Here we will describe their use in generating **certifiable randomness**.

There are ample uses for random bits in both scientific and ordinary computing. But due to their effectively deterministic nature, our common computing devices rely on pseudorandomness. ("pseudo" means "fake")

Pseudorandom numbers output sequences of numbers that may look random according to some limited tests, but the sequence is deterministically generated from an initial seed (for which a common choice is the time).

In contrast, quantum mechanics appears to give us access to true randomness:

$$\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$$

Randomness Certification

Suppose you sell me a box labeled "**Quantum** RNG", and tell me it works by measuring these states:

$$\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$$

This is all I need for marketing purposes: I can put a picture of your fancy quantum logo on my banking website, everyone feels more secure, and my stock price marginally increases.

But lets say that I care about whether these random numbers are really being generated by a quantum process. I refuse to buy the boxes until you demonstrate this to me.

You say "fine, take these two boxes, A and B. Send one to your other bank branch in China so you are sure the boxes can't communicate. Box A has two input buttons x = 0 and x = 1, and two lights a = 0, a = 1. B is similar. Ask the boxes random questions (x,y). I've rigged them up to play the CHSH game. All those Bell pairs weren't cheap, so you agree to buy the boxes if the win rate is over 80%." Do you take the deal?

The essential idea is that the CHSH game can be used for a **device-independent** test of quantumness, and hence randomness (we don't need to know about the inner workings of the device to rule out classical cheating).

Cryptography and Encryption

The goal of **cryptography** is for **honest parties** to communicate securely in the presence of **adversaries**. The honest parties may be classical or quantum, as may the adversaries.

Cryptography generally relies on *encryption*: one honest party (Alice) wants to communicate a *plaintext* message to Bob securely by *encrypting* it using a *cipher* (algorithm) or *key* (bit string) to produce unintelligible *encoded ciphertext*. Then Bob must *decode* the ciphertext to obtain the original the message.

The only encryption scheme which yields perfect information-theoretic security is called a "one time pad" (OTP), and it was analyzed by Shannon in "Communication Theory of Secrecy Systems", 1949.

The idea of the OTP is that Alice and Bob share a private key $k \in \{0,1\}^n$ (which is the OTP). The key is chosen uniformly at random in advance, and we assume the adversary does not know it.

When Alice wants to send a message $m \in \{0,1\}^n$, she sends $e = m \oplus k$ to Bob. Bob can apply k to this message to recover $m = e \oplus k$.

Cryptography and Encryption

To see the perfect security of the OTP scheme, note that every key being equally likely also means that every ciphertext is equally likely. This is because XOR is a one-to-one function.

Shannon defined perfect security in terms of conditional entropy:

$$S(M) = S(M|E)$$

Where M is the plaintext and E is the ciphertext. This is the statement that knowledge of the ciphertext provides no information at all about the message.

Notice that the OTP scheme is even secure against brute-force attacks. Trying all possible keys will simply return all possible messages.

Cryptography and Encryption

But the OTP scheme also has several drawbacks:

1. The assumption that the key held by both Alice and Bob is truly private and not known to the adversary.

2. As Shannon discussed, each key can only be used once, since multiple uses allows for the adversary to start learning information about the key.

3. If Alice and Bob generate their key using pseudorandomness it could lead to an attack on the scheme.

4. For many applications, we want the messages we send today to remain secure for many years into the future. This means the keys need to be disposed of properly.

Cryptography and Encryption

Private key encryption schemes like the OTP, which require Alice and Bob to hold a secret key, yield information-theoretic security but are practically inconvenient.

Public key encryption schemes are more practical, and are ubiquitous in real-world internet security.

In a public key scheme, Bob generates a pair of keys $k_{\text{priv}}, k_{\text{pub}}$. He keeps the private key to himself, and publishes the public key for the world to see.

When Alice has a message m to send to Bob, she encrypts using an encoding function $e = \mathcal{E}(m, k_{\text{pub}})$ that acts on the message and Bob's public key. Then she sends *e* to Bob and he decodes with a function

$$m = \mathcal{D}(e, k_{\text{priv}})$$

The security of this scheme is based on the computational assumption that decoding e without knowledge of Bob's private key is hard (and knowing the public key doesn't make it any easier).

Cryptography and Encryption

Computational security is weaker than information-theoretic security, because the former is only based on our belief that a task is hard, and also with enough time we know the task could be done.

The standard example, which is still one of the most common public key encryption schemes, is the RSA (Rivest, Shamir, and Adleman) scheme. It is based on the computational hardness of integer factoring.

The invention of the efficient quantum factoring algorithm in 1994 broke the computational hardness assumption underlying the security of RSA, at least for adversaries that have a quantum computer.

Although we are still several years away from having a QC that can factor large integers, this breakthrough led to an interest in post-quantum cryptography that allows for secure classical communication in the QC era.

Cryptography and Encryption

Understanding RSA requires some background in number theory, particularly modular arithmetic.

Given any positive integers x and n, there is an integer remainder r in the range 0,…,n-1 such that

$$x = kn + r$$

and this is denoted $r = x \mod n$. Modular arithmetic focuses on remainders.

The set of integers $\mathbb{Z}_n = \{0, ..., n-1\}$, with the operations of addition and multiplication both taken to modular by n, form an algebraic structure called a ring.

In this ring, the multiplicative identity is 1. Therefore the multiplicative inverse of $a \in \mathbb{Z}_n$ is $b \in \mathbb{Z}_n$ with

$$ab \mod n = 1$$

If such a b exists (there is not such a b for every a, which is why the ring is not a field).

Cryptography and Encryption

The greatest common divisor (gcd) of a and b is the largest integer that divides both a and b.

The integer a has a multiplicative inverse mod n iff gcd(a,n) = 1. We say a and n are co-prime.

The Euler totient function $\varphi(n)$ is the number of integers less than n that are co-prime with n.

If p is prime then all the numbers less than p are co-prime to p, so $\varphi(p) = p - 1$.

The following result is called “Euler’s generalization of Fermat’s little theorem”. If a is co-prime to n, then

$$a^{\varphi(n)} \mod n = 1$$

Cryptography and Encryption

Now we are prepared to describe RSA. Alice chooses two primes p and q, and computes n = p q.

She chooses her encryption key e randomly to be co-prime to $\varphi(n) = (p-1)(q-1)$

Compute d, the multiplicative inverse of e, modulo $\varphi(n)$. [done efficiently using Euclid's algorithm]

Alice's public key is (e,n), and her private key is (d,n). To send a message m to Alice, you send $c^d \mod n$, and to decrypt she computes $c = m^e \mod n$. This works if m is co-prime with n because

$$c^d \mod n = m^{ed} \mod n = m^{1+k\varphi(n)} \mod n = m \mod n$$

Using the fact that $m^{k\varphi(n)} \mod n = 1$. It also turns out $c^d \mod n = m \mod n$ in general.

Cryptography and Encryption

To use RSA, we first need to generate the random primes p and q. This is done by uniformly generating large random numbers and using an efficient primality test.

Computing $\varphi(n)$ is immediate if we know p and q, but we don't know how to do it efficiently knowing just n. Since $n, \varphi(n)$ would let us solve for p,q we need to keep the number of co-primes less than n a secret.

In practice, the public key e is often chosen to be 65537, which is prime. Using a standard public key is fine, as long as it is relatively prime to $\varphi(n)$, which it will be with high probability.

To encrypt and decrypt efficiently, Alice and Bob need to perform fast modular exponentiation.

This fast modular exponentiation is done by repeatedly squaring m modulo n, so it runs in time proportional to the number of binary digits in d.

Cryptography and Encryption

While quantum computation enables an efficient factoring algorithm that breaks RSA security. Fortunately, what QM takes away with one hand, it repays with the other.

In 1984, Bennet and Brassard invented a protocol which uses quantum information to distribute keys (e.g. OTPs) in an unconditionally secure way. ("secure according to the laws of physics")

This task is called quantum key distribution (QKD), and the protocol is called BB84.

We imagine that Alice and Bob share an insecure quantum channel, which they will use to send (unentangled) qubits. (insecure means that the adversary Eve can potentially intercept the qubits)

They also share an authenticated, but insecure classical channel. This means that Eve can eavesdrop on the classical messages but cannot tamper with them.

Cryptography and Encryption

To generate a key, Alice begins with random strings $x, y \in \{0,1\}^n$.

If $y_i = 0$ she sends $|x_i\rangle$, otherwise she sends $H|x_i\rangle$. So she always sends one of $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$.

Bob generates his own random string $y' \in \{0,1\}^n$. He measures the i-th qubit in the computational basis when $y'_i = 0$, and the conjugate basis when $y'_i = 1$.

After the measurements are performed, A and B announce their strings y, y'. They discard all entries with

$$y_i \neq y'_i$$

For all the remaining entries, we have that Bob measured in the same basis that Alice used to encode the information. Therefore both A and B know the bit x_i , without either of them having to announce it publicly.

Cryptography and Encryption

Next A and B take half of their secret bits (where the encoding and measurement basis agreed) and announce them publicly to serve as “check bits.” If too many check bits disagree they abort the protocol.

Finally they may perform some postprocessing on their remaining shared bits to get from a “mostly secret” key to a fully secret key (privacy amplification using hash functions).

The security of this protocol is based on the no cloning theorem and the fact that incompatible measurements disturb quantum states.

If Eve tries to eavesdrop by measuring the qubits in a random basis, she will succeed in being undetected 50% of the time. But the rest of the time she causes discrepancies, which are caught with the check bits.

Cryptography and Encryption

The security of BB84 relies on the fact that Eve cannot learn any information about the signal.

Theorem: in any attempt to distinguish two non-orthogonal states, information gain is only possible at the expense of disturbing the state.

Let the two non-orthogonal states be $|\psi\rangle, |\phi\rangle$. The most general thing Eve can do without disturbing the states is to start from a state $|u\rangle$ and apply a joint unitary

$$|\psi\rangle|u\rangle \rightarrow |\psi\rangle|v\rangle$$
$$|\phi\rangle|u\rangle \rightarrow |\phi\rangle|v'\rangle$$

But since unitaries preserve inner products, we have

$$\langle\psi|\phi\rangle\langle v|v'\rangle = \langle\psi|\phi\rangle \implies \langle v|v'\rangle = 1$$

And so v,v' are identical.

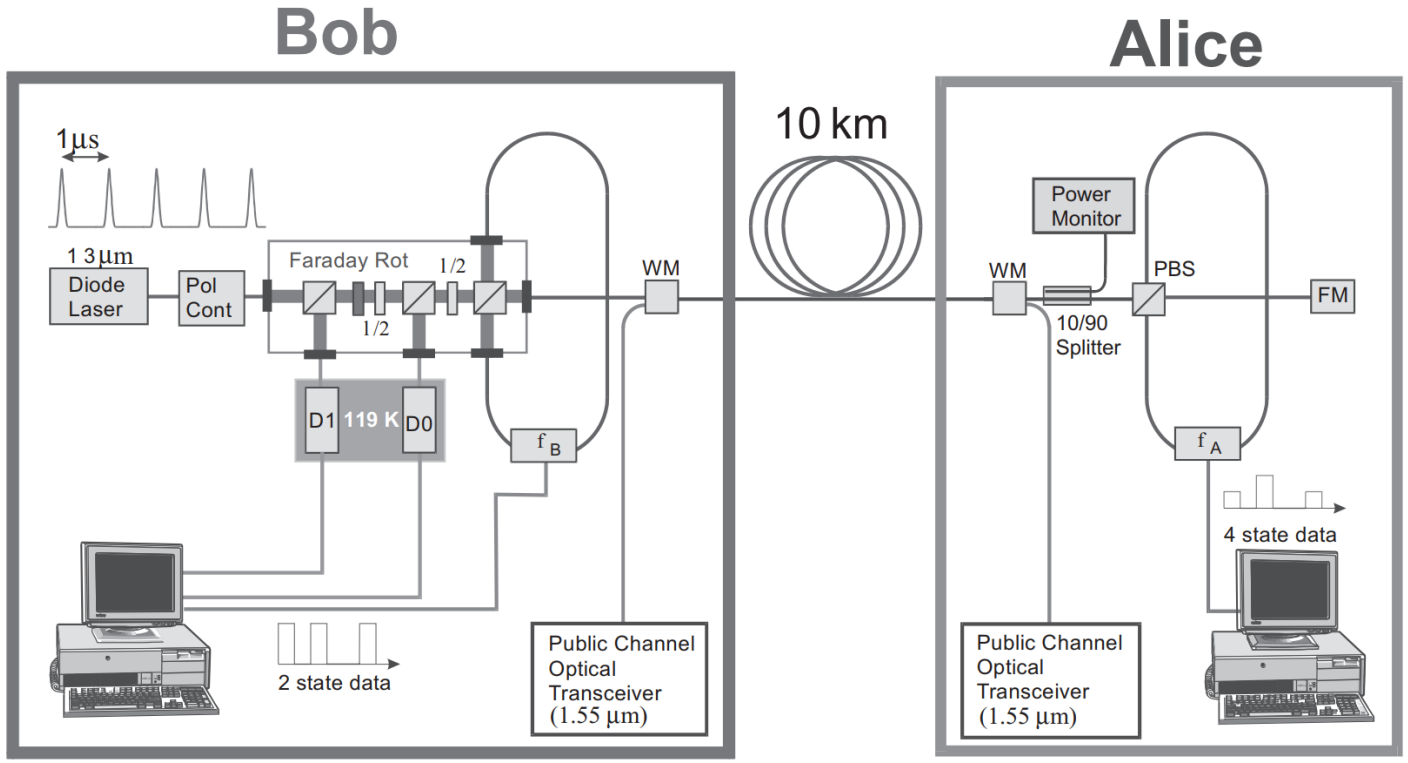
Cryptography and Encryption

In the real world, there are many issues which challenge the perfect security of BB84.

For example, photon loss caused by imperfect detectors may be mistaken for tampering by Eve.

Reliable on-demand generation of single photons is very difficult. If Alice seconds packets containing multiple photons then Eve can conceivably siphon off part of each packet without being detected.

Finally, the privacy of the key is only as good as our trust in the devices and the random number generators. This is particularly an issue for commercial applications.

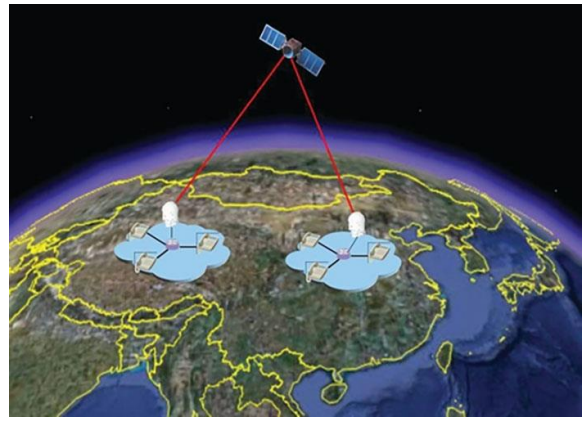


Cryptography and Encryption

Due to its potential impact on secure commercial and military communications, as well as the relative simplicity of the technology (compared to building a quantum computer), QKD has attracted billions of dollars in investment.

The QKD industry leader for many years has been a Swiss company called ID Quantique, which has a continuously operating QKD system in Geneva, and contracts with the US government.

China has also boldly invested in QKD technology, developing a connected 2000Km distribution network on their west coast, and also developing satellites for long-distance QKD.



Other major players in QKD technology include the Japan, the Netherlands, Austria, and more.

Cryptography and Encryption

We can also describe a more symmetrical version of QKD which is based on distributing Bell pairs.

Here Alice and Bob are each sent one of the qubits from $|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, which they measure in either the X or Z basis. After the protocol they can report their measurement choices and reconcile which bits they believe are private, just as in BB84.

This version of the protocol has the advantage that A and B are using true randomness, and the behavior of the entangled Bell state will also reveal tampering by the eavesdropper.

In fact, A and B could purchase black-box devices and test their ability to play the CHSH game. This is the idea behind device-independent proofs of QKD security.

However, the downside of this enhanced version of the protocol is that generating and communicating entangled photons is more technologically difficult than sending polarization states of single photons.

Cryptography and Encryption

The BB84 protocol provides an elegant example of the way in which QM can be used in cryptographic applications. But it was actually not the first cryptographic idea based on quantum information.

That credit goes to Wiesner, who in 1976 proposed a scheme for “Quantum money.” His work was not published until much later, but is considered foundational in quantum information theory.

Classical money (i.e. bank notes) can always be counterfeited, given enough resources and incentive.

The no-cloning theorem opens the exciting (for some people…) possibility of producing bank notes that cannot be copied in principle.

But for money to be useful for exchanges, we must have a practical way to verify its legitimacy and value.

Cryptography and Encryption

Therefore in a general quantum money scheme we want (1) no-cloning and (2) verifiability.

Wiesner's quantum money scheme is based on a central bank, which not only issues the notes but is also charged with verifying them (a rather inconvenient limitation).

For each note, the bank distributes $(s, |\psi_{f(s)}\rangle)$ where s is an n-bit classical serial number, and the quantum state is based on a random function (which is fixed and known only to the bank)

$$f : \{0,1\}^n \rightarrow \{1,2,3,4\}^n$$

The quantum money state is $|\psi_{f(s)}\rangle = |\psi_1(s)\rangle \otimes ... \otimes |\psi_n(s)\rangle$, where $|\psi_i(s)\rangle \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$, such that the function f chooses one of these 4 possibilities at random.

Cryptography and Encryption

Suppose Alice brings a quantum bank note $(s, |\psi_{f(s)}\rangle)$ to Bob. To verify it, he brings it to the bank.

The bank checks the serial number, and uses their secret function f to know what the state $|\psi_{f(s)}\rangle$ should be.

Next they measure in the appropriate bases to verify that the state Bob brought in really is $|\psi_{f(s)}\rangle$. If Alice was honest, then all the measurements pass with 100% probability and the note remains intact.

If Alice was trying to submit a counterfeit note, then since she does not know f she will have to guess each of the qubit states in $|\psi_{f(s)}\rangle$, but she only gets this right with probability 2^{-n} .

Any ideas for attacking this scheme?

Cryptography and Encryption

The scheme is susceptible to an iterative attack. If Alice is allowed to submit an unlimited number of notes for verification, without any penalty, then she can learn the qubit states one by one.

Specifically, she starts with a valid note and replaces one of the qubits with a test qubit in a state she guesses. If the bank accepts the note then she knows the state of the qubit she tested, otherwise she guesses again.

To fix this, the bank limits the number of times an individual can verify a note with the same serial number.

Another drawback of this scheme is that the bank is required to store an exponentially large databased of serial numbers. They can mitigate this by using a pseudorandom function for f, but this creates additional security concerns. Still, the largest drawback is the inconvenience of using a central bank to verify notes.

Cryptography and Encryption

To allow individuals to verify notes without sending them into the central bank, we could imagine a public key quantum money scheme as follows.

The bank generates a private key and a public key, $k_{\text{pub}}, k_{\text{priv}}$. The public key is known to everyone, and the private key never leaves the bank. The bank produces money states of the general form

$$(s, |\psi_{s,k_{\text{priv}}}\rangle)$$

And also distributes a verification algorithm that anyone can use to verify a quantum note, along with the serial number and the public key,

$$V(s, k_{\text{pub}}, |\psi\rangle)$$

Which returns 1 if $|\psi\rangle = |\psi_{s,k_{\text{priv}}}\rangle$. There have been many proposals for public key quantum money, but all schemes have either been broken, or require strong unproven assumptions for security. Therefore designing a good public key quantum money scheme remains an active area of research.