The Secret Life of Alice: Quantum Data Security

Word Count: 2825

1 We Have a Visitor...

1.1 First Contact

“Alice, the Martian Deep-Space Radio Array has received [one] new message."

“Tell Bob to quit playing with the Lunar transmitter. He has work to do."

“As you wish. You still have [one] new message."

Alice swung around from the optical bench, beam-splitter still in hand, looking at the computer in disbelief. The soft teal display stared back placidly, waiting for the next voice command.

“Play message."

“Best regards, we are Nonmentiorians, from planet Veritas, in a star system about 700 light-years from your sun. I am messaging you from an expedition starship. In our search for other intelligent lifeforms, we intercepted Voyager 1, and reconstructed English from its gold data-bank [**Figure 1 flashes**]. We will arrive at Terra in 3 Terran years, but in the meantime we accept light-speed communication via this channel."

As much as Alice was overwhelmed by being the proverbial first contact, something seemed off. Before she could chew on it, Bob's confused face filled the display.

“Alice! Are we compromised? I wasn't using the transmitter, how could you have received a proper message?"

“We're about to find out," replied Alice, her inkling now a full-fledged suspicion.

Figure 1: The Golden Record in Voyager 1, containing analog data of audio, images, and even human brain-wave scans[1].

1.2 Friend or Foe

"Greetings, we are pleased to finally know that we are not alone. If I may, how were you able to encode a message with this facility's exact specifications, and what do you plan to do on Earth?"

The reply came in an hour, this time in the form of the Mars station's Standard Instructional Package. "Open SIP file," said Alice, mirroring Bob's look of shock through the inter-comm.

[**Alien SIP Loaded**] Hi there! I'm your AI instructor for today. This lesson is on how our relay virus could break your station's encryption to obtain data such as this SIP's formatting. We are a perfect race of cybernetic beings with absolute quantum and classical computing power. Our civilisation is built upon complete truth and transparency, but from your data-bank we see humans keep countless secrets from each other using technology. We will thus dismantle all Terran cryptography networks, creating an open society! Since we are superior, we will assume the role of supreme benevolent leaders—

“Deep joy," remarked Bob, ever the sarcastic one.

“Computer, set starship as Extraterrestrial Virulent Entity, and check station’s network isolation."

“EVE designation set. No channels from Mars-Lunar network to Earth detected."

“Good, let’s keep it that way. Now we have to brush up on cryptography through an alien!"

2 EVE

> *‘...the enemy knows the system...’*
>
> Claude Shannon, 1949[2]

2.1 Key to Classical Cryptography

[**SIP Resume**] Cryptography is the practice of creating secret messages between two or more parties, and has existed in human society since writing was invented[3]. A powerful member of your species called Julius Caesar *encrypted* (“garbled up") his messages by substituting each letter with the one that followed it 3 places down the alphabet. The receiving party would *decrypt* the message by doing the reverse.

As we see from the Caesar cipher, the two parties must share knowledge of the *system*, which is the procedure to execute (shifting alphabets), and the *key*, which is the number of letters to shift in this context [**Figure 2 flashes**]. Broadly speaking, the key is the additional input needed alongside the message (called *plaintext*) for an algorithm to perform encryption (creating the *ciphertext*) and/or decryption. By keeping such information secret to outsiders, ciphertext intercepted while in transit would be unintelligible. At your current technology level, it is sufficient that only the key remains secret for good security (against other humans at least)[2,3,4,5].

2.2 A Little Asymmetry Looks Good

The Caesar cipher uses a *symmetric* key, where the encryption key is also used for decryption. While using symmetric keys can create unbreakable ciphertext, such as with the One-Time Pad (OTP)[2,6], this method has a deadly weakness: *key distribution*[3]. If two distant parties wish to exchange symmetrically encrypted secrets, they must first share the key via a potentially unsecure channel. If the key is compromised, the ciphertext offers no security.

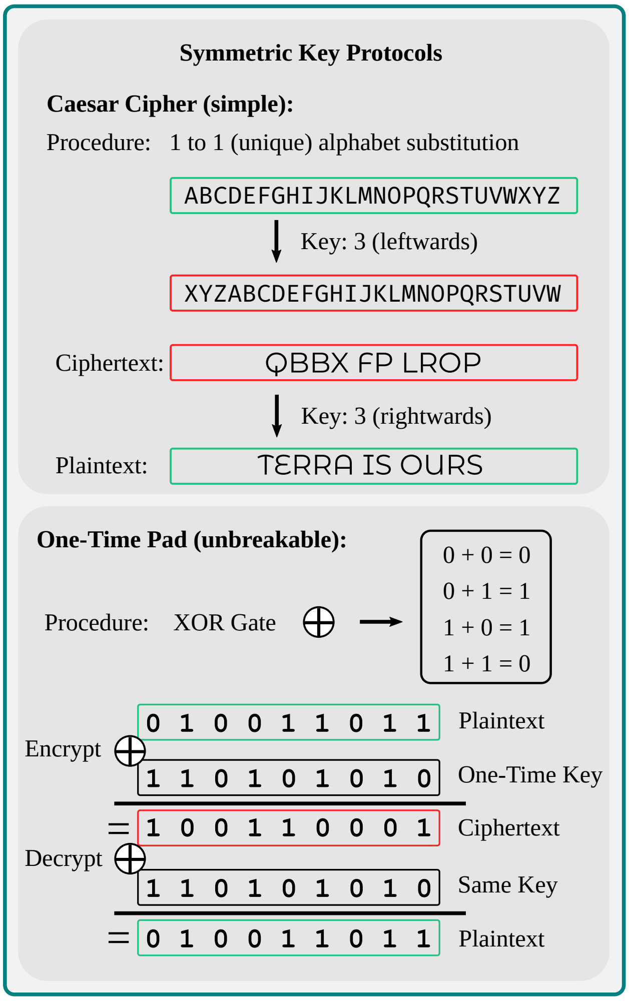

Symmetric Key Protocols
Caesar Cipher (simple):
Procedure: 1 to 1 (unique) alphabet substitution
ABCDEFGHIJKLMNOPQRSTUVWXYZ
Key: 3 (leftwards)
XYZABCDEFGHIJKLMNOPQRSTUVW
Ciphertext:
QBBX FP LROP
Key: 3 (rightwards)
Plaintext:
TERRA IS OURS
One-Time Pad (unbreakable):
Procedure: XOR Gate ⊕
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0
Encrypt ⊕
0 1 0 0 1 1 0 1 1 Plaintext
1 1 0 1 0 1 0 1 0 One-Time Key
= 1 0 0 1 1 0 0 0 1 Ciphertext
Decrypt ⊕
1 1 0 1 0 1 0 1 0 Same Key
= 0 1 0 0 1 1 0 1 1 Plaintext

Figure 2: Illustration of symmetric key cryptography protocols. The One-Time Pad (OTP) example shows messages as binary arrays (1 or 0). For the ciphertext to unbreakable, the OTP key must be random, at least as long as the plaintext, and destroyed after use.

In response, your scientists invented the *public* (or *asymmetric*) key system (PKS), enabling distribution by using two distinct keys: a public key for encryption and a *private* key for decryption[7]. This is possible due to “one-way" mathematical algorithms that are simple in one direction, but far harder to reverse without extra information at hand. Current human systems seem to still predominantly use the Rivest-Shamir-Adleman (RSA) protocol[4,5,8] for key distribution. [**Figure 3 flashes**] In RSA, Alice would declare two positive coprime (no common factors other than 1) integers (N, e) as the public key, where N is the product of two massive prime numbers pq which only Alice knows, and $e \ll N$. If Bob wishes to transmit say a secret integer P (smaller than N) to Alice through an open channel, he encrypts P into the ciphertext number C by finding the remainder of P^e divided by N. In your modular arithmetic, this is

$$C = P^e \bmod N \tag{1}$$

where C can be converted back into P via a similar operation with a special integer d (the private key):

$$C^d \bmod N = P^{ed} \bmod N = P \tag{2}$$

This is possible because d was chosen such that

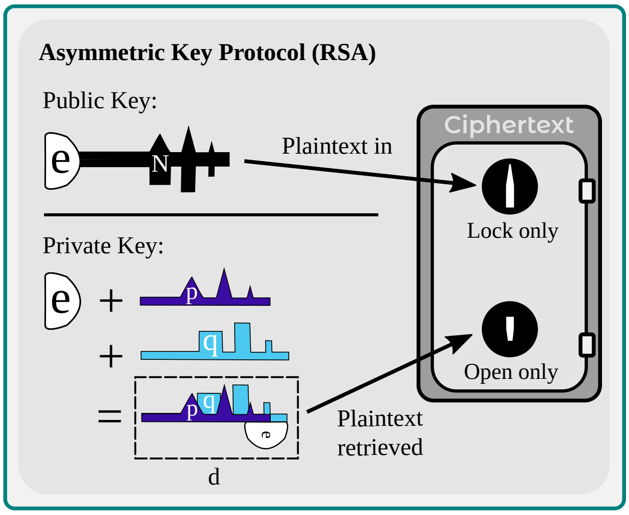


Figure 3: Concept illustration of the Rivest-Shamir-Adleman (RSA) public key protocol. Public key users are unable to form the private key needed to "unlock" the ciphertext as it is difficult to break N apart to give p and q for making d.

$ed = k\phi(N) + 1$ for some integer $k > 0$, where $\phi(N)$ is the totient of N (number of integers up to N that are coprime to N). In modulo form this relation is:

$$ed \equiv 1 \pmod{\phi(N)} \tag{3}$$

Since $N = pq$, $\phi(N) = (p-1)(q-1)$, and using Euler's Theorem

$$P^{\phi(N)} \equiv 1 \pmod{N} \tag{4}$$

for the coprime integers P and N, we can show how Equation 2 works more explicitly using modular exponentiation:

$$P^{ed} \bmod N = P^{k\phi(N)+1} \bmod N = (1)(P) \bmod N \tag{5}$$

which trivially evaluates to P since $P < N$. Alice easily calculates $\phi(N)$ with p and q, thus obtaining d by solving Equation 3 using an efficient method like Euclid's Algorithm[7]. On the other hand, it is (assumed to be) much harder for Bob to factorise N back to p and q to obtain the private key.

2.3 Power Overwhelming

The fact that humans still use such *number-theoretic* algorithms to both generate symmetric keys and distribute them shows that you still lack the quantum (and classical) computing power needed to break the "one-way" assumption[8]. However, the underlying principles are universal:

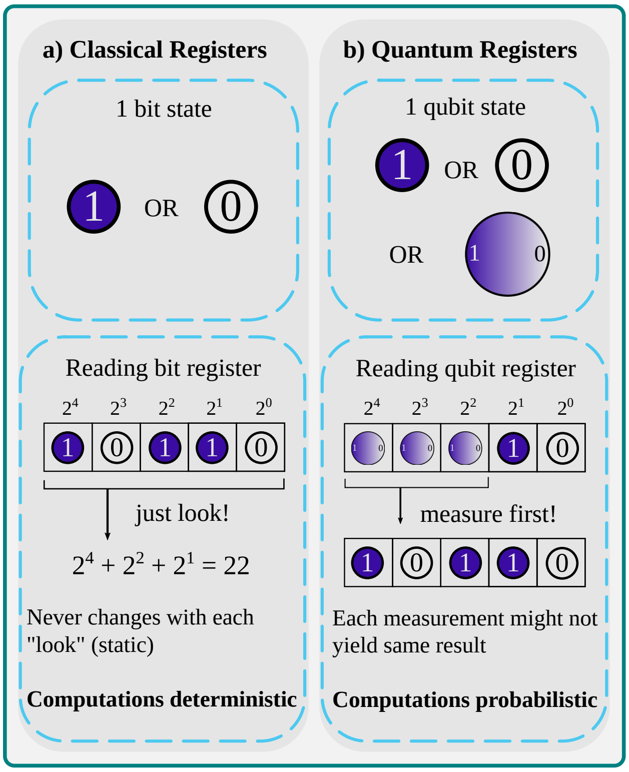


Figure 4: a) Graphic depicting the classical binary register. b) Graphic depicting the quantum binary register, where one qubit can additionally be in a superposition of states.

[**Figure 4a flashes**] In most computers, a single unit of information is represented and stored as a binary state called a *bit*: either 1 or 0. An array of bits is a *register*, which can be used to represent numbers according to the binary number system. Classical computation always involves a sequence of operations that compare bit states and then decide whether to toggle between the bit's two states. After the bits in a register have been suitably flipped, the result can simply be read off as a binary number.

[**Figure 4b flashes**] In a quantum computer, however, a special kind of operation can bring the *qubit* (quantum bit) to take an additional state: a linear superposition of the two distinct states. By your mathematical convention[4,9,10], this general 1-qubit state ψ is represented as:

$$|\psi\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle \tag{6}$$

where $|0\rangle$, $|1\rangle$ are 2-dimensional *orthogonal* vectors each representing a binary state, with α and β being the weight *amplitudes* that satisfy

$$|\alpha|^2 + |\beta|^2 = 1 \qquad \alpha, \beta \in \mathbb{C} \tag{7}$$

The key concept here is that an additional act of *measurement* is required to collapse (reduce) the superposition into one of the two distinct states depending on their probabilities $|\alpha|^2$ and $|\beta|^2$. This random behaviour is a physical property of the qubit, such as photon polarisation[9,10], where a measurement would be passing the photon through a linear polariser. We can also create a superposition of states for a 2-qubit register:

$$\begin{aligned} |\psi_0\rangle |\psi_1\rangle &= (\alpha |0\rangle + \beta |1\rangle)(\lambda |0\rangle + \gamma |1\rangle) \\ &= \alpha\lambda |00\rangle + \alpha\gamma |01\rangle + \beta\lambda |10\rangle + \beta\gamma |11\rangle \end{aligned} \tag{8}$$

We see that this is a superposition of all 4 possible 2-bit binary numbers. Generalising to *n* qubits and converting the binary numbers to decimal, we have

$$|\Psi\rangle = \sum_{x=0}^{2^n-1} \alpha_x |x\rangle \tag{9}$$

which is a quantum register in a superposition of *all* integer read-outs from 0 to $2^n - 1$ with corresponding amplitudes α_x. We effectively prepare 2^n unique outcomes by just performing that special operation n times!

“If we only obtain one random outcome from measuring the register, isn't this just a glorified random number generator?" remarked Bob.

Ah, this is where quantum computing becomes exciting! Much like how classical computers apply boolean *gates* (like XOR) on registers to change them into the final state, quantum computers apply a sequence of *quantum gates* representing a function. Consider two quantum registers R_0 and R_1, with initial arbitrary fixed values of x and 0. Applying the quantum operations (collectively called Q_f) causes the states to change accordingly:

$$|x\rangle |0\rangle \xrightarrow{Q_f} |x\rangle |0 + f(x)\rangle = |x\rangle |f(x)\rangle \tag{10}$$

where $f(x)$ is the function Q_f represents that takes R_0 as the input and stores the output in R_1. This is still a deterministic computation of $f(x)$ for a fixed x, but if we first transform R_0 into a superposition of all its possible states like Equation 9 (ignoring the amplitude terms for now) we obtain

$$\sum_{x=0}^{2^n-1} |x\rangle |0\rangle \xrightarrow{Q_f} \sum_{x=0}^{2^n-1} |x\rangle |f(x)\rangle \tag{11}$$

That is, *all* possible $f(x)$ for a single Q_f computation! As the measured value of R_1 is now correlated to that of R_0, the two registers are said to be *entangled*. Even though only one random final result will be obtained, before any measurement we can use more quantum operations to tune the probability amplitudes such that only the relatively useful states are likely to be obtained[4].

To break RSA (i.e. factorise $N = pq$), you would use a quantum algorithm like Shor's Algorithm[4], but first a convenient mathematical observation: the remainder of s^x divided by N for positive integers $s < N$ and x forms a repeating pattern for increasing x. For example, with $N = 3 \times 7 = 21$ and $s = 8$, the function $F_N(x) = s^x \bmod N$ yields $1, 8, 1, 8...$ for $x = 2, 3, 4, 5....$ The sequence has a *period* $r = 2$ integers. Calculating the greatest common divisor of N and $s^{r/2} \pm 1$ would give p and q[3,4,9] (try it!). Finding r classically for large N is as difficult as factoring N directly, but surprisingly efficient with Shor's Algorithm[3]:

1. Choose a random s. Set up an equally weighted R_0 and R_1 to achieve Equation 11's state, this time using $F_N(x) = s^x \bmod N$:

$$\sum_x |x\rangle |0\rangle \xrightarrow{Q_f} \sum_x |x\rangle |F_N(x)\rangle \tag{12}$$

2. Measuring R_1 gives a random value $F_N(k)$. Due to the function's periodicity, the entangled R_0 must now be in a superposition of states $|x\rangle$ where $x = k, k + r, k + 2r, ...$ such that $F_N(x) = F_N(k)$.

3. Measuring R_0 at this point does not give r directly since we do not know the random k offset. We thus use a Q_f that executes the discrete Fourier transform (FT) on R_0's values (*while still in superposition*). As with the classical Fourier transform, this reveals the relative weights of the component “frequencies" for a number sequence with some periodicity. In this case, we only have the period r, with the weights being the probability amplitudes.

4. If we measure R_0 now (in the “reciprocal" Fourier domain), we obtain with high probability an x value that satisfies

$$x = k\frac{2^n}{r} \tag{13}$$

for some integer k, where n is R_0's size. Rearranging the terms we can get

$$\frac{x}{2^n} = \frac{k}{r} \tag{14}$$

At this point, it is very likely that k and r are coprime[3], which makes $\frac{k}{r}$ irreducible. Since we know x and 2^n, we can obtain r by taking the denominator of the fully reduced fraction on the left. This entire process gives a usable r with a probability[4] of $\sim \frac{1}{2}$, so we can repeat it a few more times with better guesses of s to factorise N: exponentially faster than classical computers[11]. Our quantum computers have much more qubits needed to store and process your largest 4096-bit RSA keys, so we simply log your communication channels for the public keys, factorise them, decrypt ciphertexts to get distributed symmetric keys, and access all your secrets!

“How then...can we guard against you?" asked Alice, taking her chances.

Did the AI just hesitate? I'm afraid my session has terminated. Goodbye! [**Alien SIP Closes**]

“Hypocrites after all," muttered Alice.

“Do I tell Earth-command to wipe all our secrets?" Bob asked gravely.

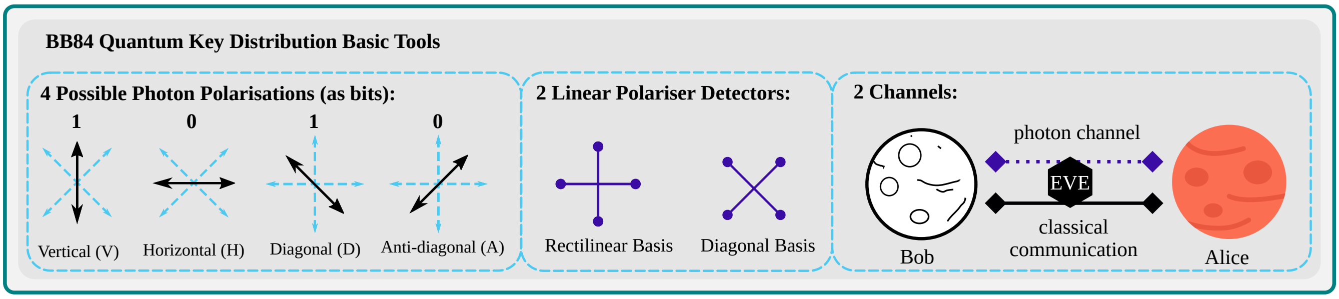


Figure 5: Illustration of the 3 main components in BB84. Notice how rectilinear polarisations can be formed from superposing diagonal ones and vice versa.

"Not so fast, perfect EVE did at least reveal its biggest weakness: its bounded by the same physics as us! I'm sending an SIP for *quantum cryptography*..."

3 The Holy Grail

3.1 Weakest Link

[**SIP Loaded**] Hi there! With *quantum key distribution* (QKD), we might just be able to use the same quantum mechanics against EVE! The core weakness with our classical cryptography is the fact that 1) EVE can eavesdrop on classical public channels *without detection*[12] and 2) EVE can decrypt any ciphertext that are created with mathematics-based key protocols. If any algorithm in a security chain can be cracked, all information within the chain, such as old and future keys, is compromised.

A potential solution is to replace the PKS with QKD. Broadly speaking, Alice would use a quantum channel to transmit a stream of random qubits to Bob. To eavesdrop on this channel, EVE must measure the qubits, which would inevitably cause some qubits to randomly collapse to one of the two orthogonal states. The very act of eavesdropping thus generates higher than expected errors in the qubit string[5,13,14], which reveals EVE's presence and how much information was leaked[15]. After thorough checking of the qubit stream, Alice and Bob share a completely secret sequence of states, which can be used in OTP encryption. This gives us *information-theoretic* (unconditional) security[11], denying EVE information not with difficult mathematics, but with the fundamental mechanisms of our reality!

3.2 BB-8...4?

We can understand QKD better using the Bennett-Brassard 1984 (BB84) protocol, which is proven to be unconditionally secure against passive monitoring[5,16,17]. [**Figure 5 flashes**] In BB84, the qubit is a single photon, with the bit information being the direction of linear polarisation, of which we choose 4 convenient angles: Vertical (V), Horizontal (H), $+45°$ (A), and $-45°$ (D). As shown, V and H form an orthogonal basis pair (the rectilinear basis), as do A and D (the diagonal basis). We see that V and H polarisation states can each be represented by a superposition of A and D states with equal coefficients ($\frac{1}{\sqrt{2}}$) and vice versa. If we try to measure a V or H polarised photon with a diagonal basis polariser, we will detect a randomised state of either A or D, because the probabilistic superpositions in V and H collapsed into the diagonal basis. No information about the photon's original state can be inferred[13]. It is only with the rectilinear basis do the photons behave classically and we can measure their state as is. The same argument applies to A and D polarised photons. If we assign V and D as 1, with H and A being 0, we now have two kinds of qubits requiring two different measurement bases to read their states deterministically. [**Figure 6 flashes**] With this in mind, we now perform the protocol:

1. Alice randomly generates a random array of bits locally *and* a corresponding basis (rectilinear or diagonal, equal probability).

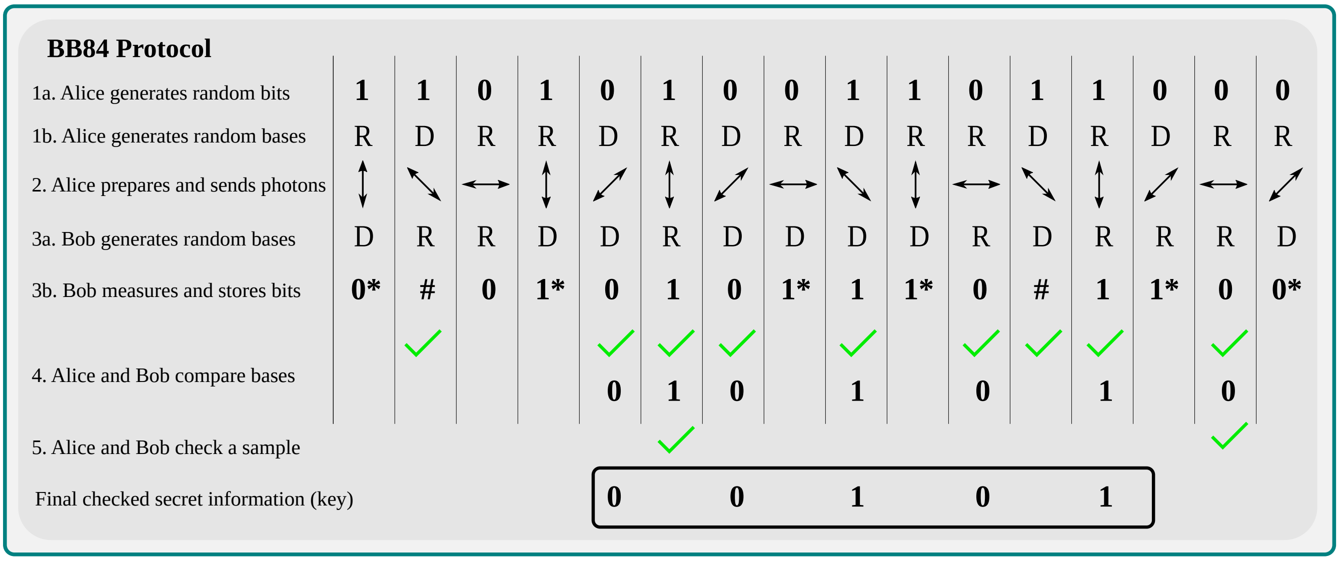
2. Using the bases and bit values to obtain the corresponding polarisations, she transmits photons prepared in those polarisations to Bob.

3. Bob receives the photons and measures them with *his* randomly generated sequence of bases, recording the bit values represented by the measurement.

4. Bob then publicly announces the basis sequence he used, as does Alice, and the two sift out the bits corresponding to a basis match.

5. Finally, Bob randomly samples some of the sifted bits and publicly checks with Alice if they match. If so, the remaining sifted bits can now be used as an OTP symmetric key for encrypting a message.

If EVE is not present, Bob can ideally expect about half of his randomly chosen bases to match Alice's[13]. Conversely, if EVE tries to measure the qubits and retransmit the measured state to Bob, it is bound to create disagreement between Bob and Alice in the final step,



BB84 Protocol

1a. Alice generates random bits	**1**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**0**	**0**
1b. Alice generates random bases	R	D	R	R	D	R	D	R	D	R	R	D	R	D	R	R
2. Alice prepares and sends photons	↕	⤡	↔	↕	⤢	↕	⤢	↔	⤡	↕	↔	⤡	↕	⤢	↔	⤢
3a. Bob generates random bases	D	R	R	D	D	R	D	D	D	D	R	D	R	R	R	D
3b. Bob measures and stores bits	**0***	**#**	**0**	**1***	**0**	**1**	**0**	**1***	**1**	**1***	**0**	**#**	**1**	**1***	**0**	**0***
4. Alice and Bob compare bases		✓			✓ **0**	✓ **1**	✓ **0**		✓ **1**		✓ **0**	✓	✓ **1**		✓ **0**	
5. Alice and Bob check a sample						✓									✓	
Final checked secret information (key)					**0**		**0**		**1**		**0**		**1**			

Figure 6: Illustration of a successful BB84 run (implying EVE is absent). R represents the rectilinear basis, and D the diagonal basis. The asterisks mark bits that are unreliable as they were measured by Bob with the wrong basis. The hashes indicate bits lost to equipment limitations like photodetector inefficiency.

as Bob, though choosing the same basis, may have been sent the wrong polarisation state (EVE can only guess!). According to the no-cloning theorem in quantum mechanics[18], it is also physically impossible for EVE to copy an intercepted photon and measure the duplicates instead. Once Alice and Bob see a disagreement rate higher than the error of their equipment, we conclude that the channel is compromised, and we restart the process until we obtain a usable secret key.

Assuming our devices are safe from EVE, the final threat to address would be if EVE not only performs an intercept-resend style attack, but also an impersonation (man-in-the-middle) attack. Thankfully, before EVE arrives, we can still use RSA to establish authentication keys between Alice and Bob and then perform QKD. The beauty of QKD is that after this first round of identity verification, we can use QKD to create new authentication keys that are truly random and independent of previous keys. *All* subsequent QKD runs would be information-theoretically secure[15], even if EVE obtains the initial RSA-encrypted key!

4 Epilogue

> *'We can only see a short distance ahead, but we can see plenty there that needs to be done.'*
>
> Alan Turing, 1950[19]

"If QKD can provide such perfect secrecy, why are we still using number-theoretic schemes like RSA for PKS, and AES for symmetric keys[8]?" asked Alice.

The answer lies in implementation: since number-theoretic algorithms came first, our existing classical network systems were designed to execute them cheaply and efficiently. Switching to the QKD paradigm at this point would require an infrastructure overhaul with fiber-optics and many more satellites around Earth to transmit photons globally[11]. Besides, security was always a matter of what is "good enough", and since we do not have large quantum computers[20] yet, our current systems still suffice. That being said, we are already in a cat-and-mouse game between increasingly secure cryptography and ever faster ways of overcoming it. For all the absolute security QKD promises, our current machines still require much more refinement, such as in improving the rate of key generation and efficiently producing single photons[5]. Interestingly, the research into practical QKD, such as single photon detector technology, simultaneously gives us insight into how larger and more stable quantum computers can be built[11]! Quantum cryptography is indeed an intricate meld of information theory, statistics, technology, and physics. However, whether we can overcome EVE is rather uncertain. [**SIP Closes**]

"Well then, guess Bob and I better start tinkering."

References

[1] NASA/JPL-Caltech. The golden record, 2017. URL `https://www.jpl.nasa.gov/images/voyager/20171201/voyager_record20171201.jpg`. [Online; accessed January 5, 2020].

[2] C. E. Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4): 656–715, 1949. doi: 10.1002/j.1538-7305.1949.tb00928.x.

[3] A. K. Ekert. From quantum-codemaking to quantum code-breaking. *arXiv e-prints*, art. quant-ph/9703035, March 1997. URL https://ui.adsabs.harvard.edu/abs/1997quant.ph..3035E.

[4] E. Gerjuoy. Shor's factoring algorithm and modern cryptography. an illustration of the capabilities inherent in quantum computers. *American Journal of Physics*, 73(6):521–540, 2005. doi: 10.1119/1.1891170. URL https://doi.org/10.1119/1.1891170.

[5] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev. The security of practical quantum key distribution. *Rev. Mod. Phys.*, 81:1301–1350, Sep 2009. doi: 10.1103/RevModPhys.81.1301. URL https://link.aps.org/doi/10.1103/RevModPhys.81.1301.

[6] G. S. Vernam. Cipher printing telegraph systems: For secret wire and radio telegraphic communications. *Journal of the A.I.E.E.*, 45(2):109–115, 1926. doi: 10.1109/JAIEE.1926.6534724.

[7] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978. ISSN 0001-0782. doi: 10.1145/359340.359342. URL https://doi.org/10.1145/359340.359342.

[8] D. J. Bernstein and T. Lange. Post-quantum cryptography. *Nature*, 549(7671):188–194, 2017. doi: 10.1038/nature23461.

[9] A. K. Ekert and R. Jozsa. Quantum computation and shor's factoring algorithm. *Rev. Mod. Phys.*, 68:733–753, Jul 1996. doi: 10.1103/RevModPhys.68.733. URL https://link.aps.org/doi/10.1103/RevModPhys.68.733.

[10] N. D. Mermin. From cbits to qbits: Teaching computer scientists quantum mechanics. *American Journal of Physics*, 71(1):23–30, Jan 2003. ISSN 1943-2909. doi: 10.1119/1.1522741. URL http://dx.doi.org/10.1119/1.1522741.

[11] H. Lo, M. Curty, and K. Tamaki. Secure quantum key distribution. *Nature Photonics*, 8(8):595–604, Jul 2014. ISSN 1749-4893. doi: 10.1038/nphoton.2014.149. URL http://dx.doi.org/10.1038/nphoton.2014.149.

[12] M. Dušek, N. Lütkenhaus, and M. Hendrych. Quantum cryptography. *Progress in Optics*, page 381–454, 2006. ISSN 0079-6638. doi: 10.1016/s0079-6638(06)49005-3. URL http://dx.doi.org/10.1016/S0079-6638(06)49005-3.

[13] C. H. Bennett and B. Gilles. Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science*, 560:7 – 11, 2014. ISSN 0304-3975. doi: https://doi.org/10.1016/j.tcs.2014.05.025. URL http://www.sciencedirect.com/science/article/pii/S0304397514004241.

[14] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin. Experimental quantum cryptography. In *Journal of Cryptography*, EUROCRYPT '90, page 253–265, Berlin, Heidelberg, 1991. Springer-Verlag. ISBN 038753587X.

[15] D. Stebila, M. Mosca, and N. Lütkenhaus. The case for quantum key distribution. *Quantum Communication and Quantum Networking*, page 283–296, 2010. ISSN 1867-822X. doi: 10.1007/978-3-642-11731-2_35. URL http://dx.doi.org/10.1007/978-3-642-11731-2_35.

[16] Peter W. Shor and John Preskill. Simple proof of security of the bb84 quantum key distribution protocol. *Physical Review Letters*, 85(2):441–444, Jul 2000. ISSN 1079-7114. doi: 10.1103/physrevlett.85.441. URL http://dx.doi.org/10.1103/PhysRevLett.85.441.

[17] D. Mayers. Unconditional security in quantum cryptography. *J. ACM*, 48(3):351–406, May 2001. ISSN 0004-5411. doi: 10.1145/382780.382781. URL https://doi.org/10.1145/382780.382781.

[18] W.K. Wootters and W.H. Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, October 1982. doi: 10.1038/299802a0.

[19] A. M. Turing. Computing Machinery and Intelligence. *Mind*, LIX(236):433–460, 10 1950. ISSN 0026-4423. doi: 10.1093/mind/LIX.236.433. URL https://doi.org/10.1093/mind/LIX.236.433.

[20] J. Suo, L. Wang, S. Yang, W. Zheng, and J. Zhang. Quantum algorithms for typical hard problems: a perspective of cryptanalysis. *Quantum Information Processing*, 19(178), April 2020. doi: https://doi.org/10.1007/s11128-020-02673-x. URL https://doi.org/10.1007/s11128-020-02673-x.