



UNIVERSITY OF WEST ATTICA  
SCHOOL OF ENGINEERING  
DEPARTMENT OF INFORMATICS AND COMPUTER ENGINEERING

Master of Science in New Generation Communication  
Networks and Distributed Applications Environments

Master Thesis

# Quantum Technologies and the Integration of Security Principles on Data Communications

Author: Dimitris Foustanas  
S.N. 21003

Lecturer: Dr. Antonios Bogris, Professor



Master Thesis  
Quantum Technologies and the Integration of Security  
Principles on Data Communications

Dimitris Foustanas  
S.N. 21003

Examinations Committee:

Professor Antonios Bogris  
Professor Vasileios Mamalis  
Associate Professor Ioanna Kantzavelou

Examinations Date: May 17, 2023



## **ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

Ο κάτωθι υπογεγραμμένος Δημήτριος Φουστάνας του Αντωνίου, με αριθμό μητρώου 21003 φοιτητής του Προγράμματος Μεταπτυχιακών Σπουδών «Δίκτυα Επικοινωνιών Νέας Γενιάς και Καταναεμημένα Περιβάλλοντα Εφαρμογών» του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Είμαι συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Επιθυμώ την απαγόρευση πρόσβασης στο πλήρες κείμενο της εργασίας μου εκτός από ακαδημαϊκούς σκοπούς και έπειτα από αίτηση μου στη Βιβλιοθήκη και έγκριση του επιβλέποντα καθηγητή.

Ο Δηλών



Δημήτρης Φουστάνας



## ΕΥΧΑΡΙΣΤΙΕΣ

Όπως στην επιστήμη, έτσι και η πληροφορική, είναι ένας αναπτυσσόμενος κλάδος, που βασίζεται στις εξελίξεις του "σήμερα" για να εξηγήσει της εξελίξεις του "αύριο". Αν αυτό δεν είναι δυνατόν, τότε να διορθώσει τον τρόπο σκέψης του "χθες", και ουσιαστικά συμβάλλοντας σε ένα καλύτερο "μεθαύριο". Όλες οι εφευρέσεις, ανά τους αιώνες, ακολουθούν το ίδιο σκεπτικό.

Σε όλο αυτό το ποτάμι πληροφοριών και εξελίξεων, οι καθηγητές ενορχηστρώνουν τον ρόλο της καθοδήγησης του ανθρώπινου νου. Δημιουργώντας μια "πυξίδα" που ουσιαστικά πολώνει την σκέψη των φοιτητών τους σε νέους ορίζοντες, φλερτάροντας με νέες ιδέες και γνώσεις. Αν και στο σύνολο τους όλοι οι καθηγητές προσωποποιούν των ίδιο ρόλο, ο επιβλέπων καθηγητής μου, το διέπραξε σε μια τόσο κρίσιμη περίοδο, και για το μεγαλύτερο χρονικό διάστημα. Θέλω λοιπών να τον ευχαριστήσω για την προσπάθεια και την πολύτιμη καθοδήγηση του.

Στην προσπάθεια την επιτυχής ολοκλήρωσης των Μεταπτυχιακών μου σπουδών, έναν πολύ μεγάλο ρόλο είχε και η υποστήριξη της οικογένειας μου. Θέλω να ευχαριστήσω τους γονείς μου για την στήριξη τους, τον καιρό που χρειάστηκε για την φοίτηση μου.

# Acknowledgements

The path to Learning is never a straight line, and to that extent not a straightforward experience. The procedural call of Cognitive Learning comes at the cost of our Professors who spread ideas and shape our minds and souls. Even tirelessly, but nor our Professor or Teachers, even think about that. To them, it's all about teaching. And it does not matter if you are five or fifty years old.

Some Professors taught me the curiosity, or the evidence-based thinking. Others, like my latter ones, the Bayesian logic. To stop learning is simply not something that can be afforded, specially for a scientist. So the weight falls onto the shoulders of Professors and Teachers. With that said, I want to give a big "Thank you" to my dear Professor, to whom I couldn't complete this Thesis on my own without a helping and guiding hand.

In the words of Steven Paul Jobs: "You cannot connect the dots, by moving forward. You can only connect the dots by looking backwards." meaning that even as a pupil in high school (or even worse, in the Freshman years of University), ones come across those wonderful peoples, that with their best interest at heart, will even "punish" you with bad grades. In that moment of time the pupil cannot understand, and doesn't know any better to get angry at his Professors. But as time passes away a couple of years, he/she understands that the things sprinkled with anger, had in fact a nudging effect in the right direction.

So ... What would it take to live in a world without any sort of "knowledge transmission", deprived of Professors and Teachers? I wouldn't have gotten **any** of my degrees and **nor is anyone else**. No cars or any means of transportation, no light bulbs or electric appliances of any kind, not even a roof over our heads. Just plain nothing ... We all got the picture, and a grim one it is. So, I as most peoples, deserve to pay tribute for all that received knowledge. Humbleness is the place to start, and as we go on, we'll find the right way to proceed.





## ΠΕΡΙΛΗΨΗ

Ένα από τα μεγαλύτερα Τεχνολογικά επιτεύγματα του 19ου αιώνα είναι ο υπολογιστής. Από την στιγμή που πρωτοεμφανιστήκαν, γνώρισε τεράστιες βελτιώσεις. Αν και ο τρόπος λειτουργίας του, είναι σχετικά απλός, είναι ιδιοφυές. Λειτουργεί με την διέλευση του, από ηλεκτρικό ρεύμα. Αυτό κωδικοποιείται σε ένα λογικό επίπεδο, του "μηδέν" και του "ένα". Από εκείνη την στιγμή, γεννήθηκαν τα Ψηφιακά Ηλεκτρονικά, και με την σειρά τους, αυτά τα λογικά επίπεδα, ονομάστηκαν Bits. Από τον πρώτο υπολογιστικό σύστημα που φπαχτείτε, μέχρι σχεδόν την δεκαετία 1980, όλα τα υπολογιστικά συστήματα ήταν βασισμένα στο προαναφερθέν σκεπτικό. Και ομαδοποιώντας αυτά τα λογικά Bit σε οκτάδες, καθιερώθηκε σαν κανόνας, για την μετάδοση μιας ηλεκτρονικής πληροφορίας.

Αυτός ο κύκλος συνεχίστηκε για σχεδόν έναν αιώνα. Μέχρι ώστε η υπολογιστική ικανότητα αυτόν των μηχανημάτων, να μην είναι σε θέση να διαχειριστεί και να ολοκληρώσει τους υπολογισμούς που χρειάζεται να φέρει εις πέρας, στο χρονικό περιθώριο που πρέπει. Ήταν κοινή λογική ότι αυτή η τεχνολογία δεν θα έπρεπε να πάει χαμένη, αλλά να εξελιχθεί σε κάτι καλύτερο. Έτσι, οι Επιστήμονες και οι Μηχανικοί, στράφηκαν σε μια πολύπλοκη και περίεργη τεχνολογία. Μια που θα χρησιμοποιούσε δεσμίδες φωτός, αντί για ηλεκτρικό ρεύμα. Αν και θεωρητικά τουλάχιστον, δεν μπορεί να υπάρξει μεγαλύτερη ταχύτητα από εκείνη του φωτός. Αυτό ήταν πολύ καλά νέα, για την ταχύτητα επεξεργασίας, αλλά υπήρχε ένα πολύ μεγάλο πρόβλημα. Κανένας δεν κατανοούσε σε ικανοποιητικό βαθμό τα Κβάντα Φωτός, προκειμένου να προσδιορίσει με ακρίβεια της συμπεριφορές του. Για τον λόγο αυτόν εντάθηκαν οι προσπάθειες. Αν και οι "Κβαντικοί Υπολογιστές" είναι κάτι επιστημονικά εφικτό, γίνονται ακόμα έρευνες προκειμένου να κατανοήσουμε καλύτερα το Κβαντικό μικρόκοσμο. Εκτός αυτού, έχοντας ένα Υπολογιστικό σύστημα που λειτουργεί με δεσμίδες φωτός, περικλειόμενο ανάμεσα σε ηλεκτρονικά συστήματα, είναι και αυτό, ένα μείζων ζήτημα. Και ο κύριος λόγος που οι Κβαντικοί Υπολογιστές δεν έχουν βγει στο εμπόριο, αλλά είναι μόνο σε ερευνητικό επίπεδο.

Σε αυτήν την διπλωματική εργασία, θα αναλύσουμε της μεθόδους Κρυπτογραφίας, από την ίδρυση της έννιας αυτής, μέχρι την Κβαντική Κρυπτογραφία. Στον δρόμο της ανάλυσης αυτής, είναι ανάγκη να αναφερθούμε στον τρόπο λειτουργίας ενός Κβαντικού συστήματος, και στα φαινόμενα που θεσπίζουν την λειτουργία τους. Σαν Μηχανικός Πληροφορικής, θα αναφερθώ στην διασφάλιση μιας Ασφαλούς πληροφορίας.

# Abstract

One of the biggest Technological marvel of the 19th century, known as the Computer, has known huge upgrades since its first launch. The working principle was simple, yet ingenious as it turned out. The flow of electricity, or the lack of, translating it to a logical "one" or "zero". This simple principle was called as Digital Electronics and these values were (and still are) named Bits. From the first ever Computer ever made, till the 1980s, that was the most solid foundation, that all computers shared. The thought of "strapping" bits together in a sequence, became the norm of information exchange.

This cycle continued for nearly a century, until the processing power of these machines were not powerful enough to process all that information in time. Halting this *magical* piece of technology was not an option. In order to diminish the processing time needed, scientists turned to a, strange but quite promising, technology. This exotic concept used light instead of electricity. Nothing comes even closer to the speed of light, so this would turn out second to none. The problem is that, even for the smartest of human kind, *no one* can exactly understand and predict the light patterns. Even the behavior of the particles constituting it is still a mystery. As time unravels it does become clear, but we are certainly not there yet. Although "Quantum Computers" is in fact a reality, science has many mysteries to solve. Besides of a system requiring light to function properly, onto a cluster of electricity-functioning systems (the *traditional Computers*), is another very big issue that is braking harshly, its propagation.

In this thesis, I will try to analyze traditional Cryptographic methods, stretching into a *post-Quantum era*. By doing so, I'll have to tackle some inner workings of an Quantum computer. These goes without saying that I will try my best (as I am not a Physicist) at explaining this phenomenons. As a Computer Scientist, I will dip into the security of information exchange between all these systems. Not all information exchange needs to happened in a secure channel, but some of it should. Any information marked as "Confidential", such as communication data, and most importantly "keys" (system/devices, cryptographic...) are not to fall in the wrong hands, or in this specific example, on wrong eyeballs!

**Keywords:** Cryptography, PKI, Post-Quantum Cryptography, QPI, Computers

*“If you can’t explain it simply, you don’t understand it well enough.”*

Albert Einstein

# Contents

<b>1</b>	<b>Securing Information in the Pre-Quantum Era</b>	<b>1</b>
1.1	Early Cryptography . . . . .	1
1.1.1	Substitution Ciphers . . . . .	1
1.1.2	Transposition Cipher . . . . .	2
1.2	Modern Cryptography . . . . .	2
1.2.1	Cryptographic Hash Functions . . . . .	4
1.2.2	Cryptographic Keys . . . . .	4
1.3	Cryptographic Hash Functions . . . . .	4
1.3.1	Properties . . . . .	5
1.3.2	Usage . . . . .	5
1.3.3	Security Level of a CHF . . . . .	5
1.3.4	Merkle–Damgård Hash Construction . . . . .	6
1.3.5	Sponge Hash Construction . . . . .	6
1.3.6	HAIFA Hash Construction . . . . .	8
1.3.7	Wide-Pipe and Narrow-Pipe . . . . .	8
1.3.8	Common Cryptographic Hash Algorithms . . . . .	8
	MD5 . . . . .	8
	SHA-1 . . . . .	8
	RIPMD-160 . . . . .	8
	Whirlpool . . . . .	8
	SHA-2 . . . . .	9
	SHA-3 . . . . .	9
	BLAKE2 . . . . .	9
	BLAKE3 . . . . .	9
1.4	Cryptographic Keys . . . . .	10
1.4.1	Kerckhoffs’ Principle . . . . .	10
1.5	Symmetric-Key Algorithms . . . . .	10
1.5.1	Asymmetric Cryptography . . . . .	10
1.5.2	Asymmetric vs Symmetric Cryptography . . . . .	11
1.5.3	Long Term or Single Use . . . . .	12
1.5.4	Hybrid Cryptosystems . . . . .	12
	Diffie - Hellman Key Exchange . . . . .	12
1.6	Digital Signatures . . . . .	13
1.7	Public Key Infrastructure . . . . .	14
<b>2</b>	<b>Honorable Mention: Kerberos</b>	<b>15</b>
2.1	Introduction . . . . .	15
2.2	The Kerberos System . . . . .	15
2.2.1	Workings of a Kerberos System . . . . .	16
2.3	Issues with a Kerberos System . . . . .	20
2.4	Kerberos And SESAME . . . . .	20
2.4.1	SESAME . . . . .	20

2.5	Conclusion Regarding Kerberos . . . . .	21
<b>3</b>	<b>The Quantum Era</b>	<b>22</b>
3.1	The birth of Quantum Mechanics . . . . .	23
3.1.1	The Old Quantum Mechanics . . . . .	24
3.1.2	The New Quantum Mechanics . . . . .	24
3.2	The Bits and the Qbits . . . . .	24
3.2.1	Bits and Qbits . . . . .	24
3.3	Quantum Entanglement . . . . .	25
3.3.1	Quantum Teleportation . . . . .	26
3.3.2	Quantum Cloning . . . . .	28
3.4	Putting it all Together . . . . .	28
3.4.1	A State Of The Art Product for Quantum Key Distribution . . .	29
3.5	Heisenberg Uncertainty Principle . . . . .	29
<b>4</b>	<b>Building A Quantum Computer</b>	<b>31</b>
4.1	Types of Quantum Computers . . . . .	31
4.1.1	Quantum Turing Machine . . . . .	31
4.1.2	Quantum Annealing . . . . .	31
4.1.3	Adiabatic Quantum Computation . . . . .	32
4.1.4	Quantum Circuits . . . . .	32
4.2	Quantum Logic Gates . . . . .	32
4.2.1	Controlled NOT Gate . . . . .	33
4.3	Quantum Monte Carlo . . . . .	33
4.3.1	The Quantum Monte Carlo Methods . . . . .	34
	Zero-temperature (Only Ground State) . . . . .	34
	Finite-temperature (Thermodynamics) . . . . .	34
	Real-time Dynamics (Closed Quantum Systems) . . . . .	34
4.4	Quantum Requirements . . . . .	34
4.4.1	The Importance of Cooling . . . . .	34
4.4.2	Eliminating "Noise" . . . . .	35
4.5	State of The Art Computers . . . . .	35
<b>5</b>	<b>Securing Information in the Post-Quantum Era</b>	<b>37</b>
5.1	Cryptographic Problems . . . . .	37
5.1.1	Integer Factorization Problem . . . . .	38
	Fermat's Factorization Method . . . . .	38
5.1.2	Discrete Logarithm Problem . . . . .	38
	An Example of a Discrete Logarithm Problem . . . . .	39
5.1.3	Elliptic-curve Discrete Logarithm problem . . . . .	39
5.1.4	Quantum Requirements . . . . .	40
5.2	Protocols based On Heisenberg Uncertainty Principle . . . . .	40
5.2.1	The BB84 Protocol . . . . .	41
5.2.2	The B92 Protocol . . . . .	42
5.2.3	The Six-State Protocol . . . . .	43
5.2.4	The SARG04 Protocol . . . . .	43
5.3	Protocols based On Quantum Entanglement . . . . .	43
5.3.1	The E91 Protocol . . . . .	43
5.3.2	Entangled Versions of the BB84 Protocol . . . . .	44
5.3.3	Cryptography without Bell's Theorem . . . . .	44
5.4	Real World QKD Concerns . . . . .	44

5.4.1	QKD with Noisy Channels . . . . .	45
5.4.2	QKD with the problem with Privacy Amplification . . . . .	45
5.5	The Research . . . . .	45
5.5.1	Lattice-Based Cryptography . . . . .	46
5.5.2	Multivariate Cryptography . . . . .	46
5.5.3	Hash-Based Cryptography . . . . .	47
5.5.4	Code-Based Cryptography . . . . .	47
	McEliece Encryption Scheme . . . . .	47
	Niederreiter Encryption Algorithm . . . . .	47
5.5.5	Supersingular Elliptic Curve Isogeny Cryptography . . . . .	47
5.5.6	Symmetric Key Quantum Resistance . . . . .	48
5.6	Comparison of Algorithms in Post Quantum . . . . .	48
5.6.1	Security Reductions . . . . .	48
5.6.2	Lattice-based Cryptography - Ring-LWE Signature . . . . .	48
5.6.3	Lattice-based Cryptography - NTRU, BLISS . . . . .	49
5.6.4	Multivariate Cryptography - Unbalanced Oil and Vinegar . . . . .	49
5.6.5	Hash-based Cryptography - Merkle Signature Scheme . . . . .	49
5.6.6	Code-Based Cryptography - McEliece . . . . .	49
5.6.7	Code-Based Cryptography - Random Linear Code Encryption . . . . .	50
5.6.8	Supersingular Elliptic Curve Isogeny Cryptography . . . . .	50
5.7	Forward Secrecy . . . . .	50
5.8	Open Quantum Safe Project . . . . .	50
<b>6</b>	<b>Conclusions</b> . . . . .	<b>52</b>
6.1	Conclusion . . . . .	52
6.2	Further Research . . . . .	53
<b>A</b>	<b>Clarifications For Chapter 1: Securing Information in the Pre-Quantum Era</b> . . . . .	<b>55</b>
A.1	Pigeonhole Principle . . . . .	55
A.2	The X.509 Standards . . . . .	55
<b>B</b>	<b>Clarifications For Chapter 2: Honorable Mention: Kerberos</b> . . . . .	<b>56</b>
B.1	Needham-Schroeder Protocol . . . . .	56
B.2	Project Athena . . . . .	57
<b>C</b>	<b>Clarifications: Chapter 3: The Quantum Era</b> . . . . .	<b>59</b>
C.1	What quantify as Quantum Particle or not? . . . . .	59
C.2	What is a "Wave Function"? . . . . .	59
C.3	What is <i>really</i> <b>Superposition</b> in Quantum Mechanics? . . . . .	59
C.4	What happens to the original object after it has undergo teleportation? . . . . .	60
C.5	The Proof for Quantum Cloning . . . . .	60
C.6	What is a Bell State Analyzer? . . . . .	62
C.7	Looking into Heisenberg Uncertainty Principle . . . . .	62
<b>D</b>	<b>Clarifications: Chapter 4: Building A Quantum Computer</b> . . . . .	<b>64</b>
D.1	Quantum Parallelism . . . . .	64
D.2	Adiabatic and Diabatic theorems . . . . .	65
D.3	Penrose's Quantum Notation Scheme . . . . .	65
D.4	Quantum Logic Gates . . . . .	66

<b>E</b>	<b>Clarifications For Chapter 5: Securing Information in the Post-Quantum Era</b>	<b>67</b>
E.1	Shor's Algorithm . . . . .	67
E.2	What is a Cusp in a function? . . . . .	68
E.3	The <i>Lattice</i> terminology . . . . .	68
E.4	Bell's Theorem . . . . .	69



# List of Figures

1.1	The Scytale Example . . . . .	2
1.2	Visual Representation of Moore's Law . . . . .	3
1.3	The "Avalanche Effect" of Hashing Data . . . . .	4
1.4	The Merkle–Damgård Hash Construction . . . . .	6
1.5	The Sponge Construction . . . . .	7
1.6	Visual Representation of Symmetric Cryptography . . . . .	11
1.7	An Example of Asymmetric Cryptography . . . . .	11
1.8	Diffie - Hellman Key Exchange . . . . .	12
1.9	A Visual Representation of the Process of Digital Signatures . . . . .	13
1.10	A Visual Representation of the Process of a PKI . . . . .	14
2.1	The Kerberos System . . . . .	15
2.2	A Visualization of the Kerberos System . . . . .	17
2.3	A Visualization of the SESAME System . . . . .	21
3.1	The Double Slit Experiment . . . . .	22
3.2	Green Laser passing Two Slits 0.4mm Wide and 0.1mm Apart . . . . .	23
3.3	The Rutherford Experiment . . . . .	23
3.4	The Bloch Sphere . . . . .	26
3.5	A Quantum Entanglement Experiment . . . . .	27
3.6	A Quantum Teleportation Experiment . . . . .	28
3.7	A Device for QKD . . . . .	29
4.1	Quantum Circuit Picturing Quantum Teleportation . . . . .	32
4.2	The CNOT Quantum Gate . . . . .	33
4.3	A Quantum Computer ( 50 Qbit) . . . . .	36
4.4	The D-Wave Processor (128Qbit) . . . . .	36
5.1	Prime Decomposition of $n = 864$ as $2^5 * 3^3$ . . . . .	38
5.2	Visual Representation Of An Elliptic Curve . . . . .	39
5.3	Visual Representation Of A QKD . . . . .	41
5.4	Visual Representation Of Photon-Bit Polarization . . . . .	41
5.5	Visual Representation Of The BB84 Protocol . . . . .	42
5.6	Visual Representation Of The B94's 2-State Encoding . . . . .	43
5.7	Visual Representation Of A QKD . . . . .	44
5.8	Visual Representation Of The OSI Model (Reference) . . . . .	51
B.1	The Symmetric Needham–Schroeder Protocol . . . . .	56
B.2	The Project Athena . . . . .	57
C.1	Schrödinger's Equation (Time-Dependent) . . . . .	59
C.2	Graphical representation of a Complex Number . . . . .	60
C.3	A Schematic of a Bell State Analyzer . . . . .	62

D.1	Penrose's Quantum Notation Scheme . . . . .	65
D.2	Quantum Logic Gates . . . . .	66
E.1	Visual Representation Of A Cusp at $(0,0)$ . . . . .	68
E.2	Visual Representation Of A Lattice in Euclidean Plane . . . . .	68

# List of Tables

1.1	A Substitution based Algorithm . . . . .	1
1.2	SHA-512 Family CHF . . . . .	9
2.1	Releases of the Kerberos System . . . . .	16
3.1	Possibilities with two Bits . . . . .	25
5.1	<b>Post Quantum</b> Algorithms Comparison . . . . .	48
5.2	<b>Post Quantum</b> Algorithms Comparison ( <i>continued</i> ) . . . . .	49
5.3	Open Quantum Safe Algorithms . . . . .	51
C.1	Teleportation of Bob . . . . .	60
D.1	Diabatic and Adiabatic Theorems . . . . .	65

# List of Abbreviations

<b>AKA</b>	<b>Also Known As</b>
<b>WSF</b>	<b>What (it) Stands For</b>
<b>PT</b>	<b>Plain Text</b>
<b>CT</b>	<b>Cipher Text</b>
<b>HF</b>	<b>Hash Function</b>
<b>CHF</b>	<b>Cryptographic Hash Function</b>
<b>IE</b>	<b>Id Est (<i>Latin</i>)</b>
<b>IV</b>	<b>Initialization Vector</b>
<b>NIST</b>	<b>National I of S Institute Standards and Technology</b>
<b>CPU</b>	<b>Central Processing Unit</b>
<b>VS</b>	<b>VerSus</b>
<b>TLS</b>	<b>Transport Layer Security</b>
<b>SSH</b>	<b>Secret SHell</b>
<b>PKI</b>	<b>Public Key Infrastructure</b>
<b>CA</b>	<b>Certificate Authority</b>
<b>RA</b>	<b>Registration Authority</b>
<b>VA</b>	<b>Validation Authority</b>
<b>TM</b>	<b>Turing Machine</b>
<b>QTM</b>	<b>Quantum Turing Machine</b>
<b>ECC</b>	<b>Elliptic Curve Cryptography</b>
<b>FS</b>	<b>Forward Secrecy</b>
<b>PFS</b>	<b>Perfect Forward Secrecy</b>
<b>HSM</b>	<b>Hardware Security Modules</b>
<b>SSO</b>	<b>Single Sign On</b>
<b>SRP</b>	<b>Secure Remote Password Protocol</b>
<b>HIP</b>	<b>Host Identity Protocol</b>
<b>KDC</b>	<b>Key Distribution Center</b>
<b>HUP</b>	<b>Heisenberg Uncertainty Principle</b>
<b>SSP</b>	<b>Six State Protocol</b>

## Chapter 1

# Securing Information in the Pre-Quantum Era

Since the birth of the *Electronic Computer*, in the 1938 till somewhere around 1980-90 where scientist were starting to go *Quantum*, every computer shared the same old logic. The pass of electric current or the lack of, gave birth to the logic of bits ("*Zero*" and "*One*", also referred as "*Binary*" logic). From the early days, people found that it is not very practical to isolate thees computer systems on their own, but rather inter-connect them by wire, forming a worldwide cluster of computer systems. Rapidly, it became **vital** to ensure that some information, that was being passed around, were not broadly shareable with **everybody else** that was connected with the same wire. Scientists and Engineers were starting to ponder with the idea of keeping some information scrambled to the vast majority of users. Making it arduous for the unwanted or curious to *unscramble* and take knowledge. What we call Cryptography was being born.

### 1.1 Early Cryptography

The role of Cryptography was not an invention solely based on Computer Systems, but rather from the old days of analog computer systems. Some simple methods for encrypting data, date all the way back to 1467. The two biggest examples of "**Substitution Ciphers**" were the ciphers of Vigenère and Ceasar.

#### 1.1.1 Substitution Ciphers

Don't get me wrong, that specific category of Ciphers are quite easy to intercept the message, break through the encryption and finally take knowledge of some "private" information, that is passed around. But thees ciphers do need a glimpse of Respect, because without them, Cryptography may have not taken off, as it did. The premise is extremely easy to grasp and bet a hold-on. It relies on simple substitution of a character by some other one. As the visual representation 1.1 is illustrating it clearly. This basic concept is implemented in the *much modern* **ROT13** cipher.

TABLE 1.1: A Substitution based Algorithm

Plain	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
Cipher	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O

... and so on...

The generalized idea is, for encryption:  $E_n(x) = (x + n) \bmod 26$   
and as for decryption:  $D_n(x) = (x - n) \bmod 26$

The information provided, is indeed senseless:  
PT: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG  
CT: QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD

But with the increasing power of the computer systems. Fortunately, this information is not staying secure for long. But the reason comes later in the chapter.

### 1.1.2 Transposition Cipher

To date, only one method of featuring transposition cipher, has stood the test of time and can be found today. The method of Scytale, used by the ancient Greeks uses a parchment, that is wrapped around a stick (or cane). While still wrapped, the message is written onto the parchment and then, it is unwrapped and carried over. on the Decrypting end, it is wrapped again onto a similar radius stick and the message is revealed. As for an example purpose, let us imagine that the message is: "I am hurt very badly help".

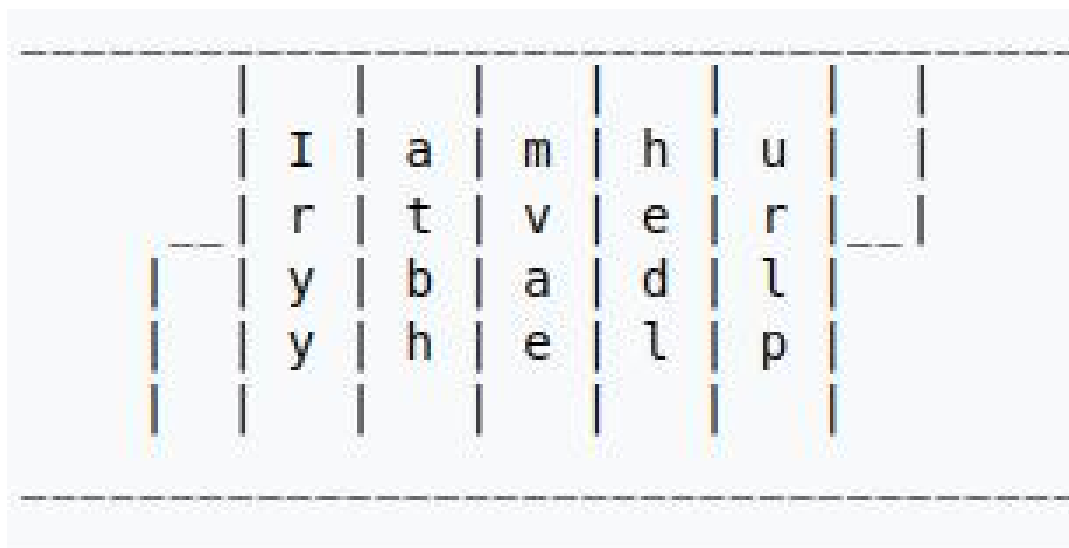


FIGURE 1.1: The Scytale Example  
(Source: <https://en.wikipedia.org/wiki/Scytale>)

After unwrapping the parchment, the CT is "Iryatbhmvaehedlurlp". That is **very** impressive, considering that this method was used all the way back in the 7th century **B.C!**

## 1.2 Modern Cryptography

Like we already said in the previous section, those methods of securing a message were not sufficient to secure the information from *modern computing systems*. Like we said on numerous occasions, the art of **Cryptography** was evolving according to the analogous of computing power, and by extension to *Moore's Law*. See in figure 1.2.



### 1.2.1 Cryptographic Hash Functions

This is the epitome of *Modern Cryptography*. As to date, the biggest majority of Cryptographic algorithms uses some kind of hash in order to properly advance. These functions are one-way, and it is impossible, for traditional binary-based computers, to inverse the function, and getting the PT from the CT. Much needs to be said about these functions, and we will discuss them in greater depth, onto an upcoming section.

### 1.2.2 Cryptographic Keys

Much like their *Locksmithing* analogous, a Cryptographic Key is a piece of data (most commonly a **string** of characters) that is given into a Cryptographic algorithm in order to proceed to a result (encryption or decryption). The making of the *Keys* is mostly **one-way**, but there afterwards usage certainly is *not*. Hence, the encryption strength, is directly related to the key being **secure** enough. Much like hashes, there is way more information to be covered, and a later section will be reserved to them.

## 1.3 Cryptographic Hash Functions

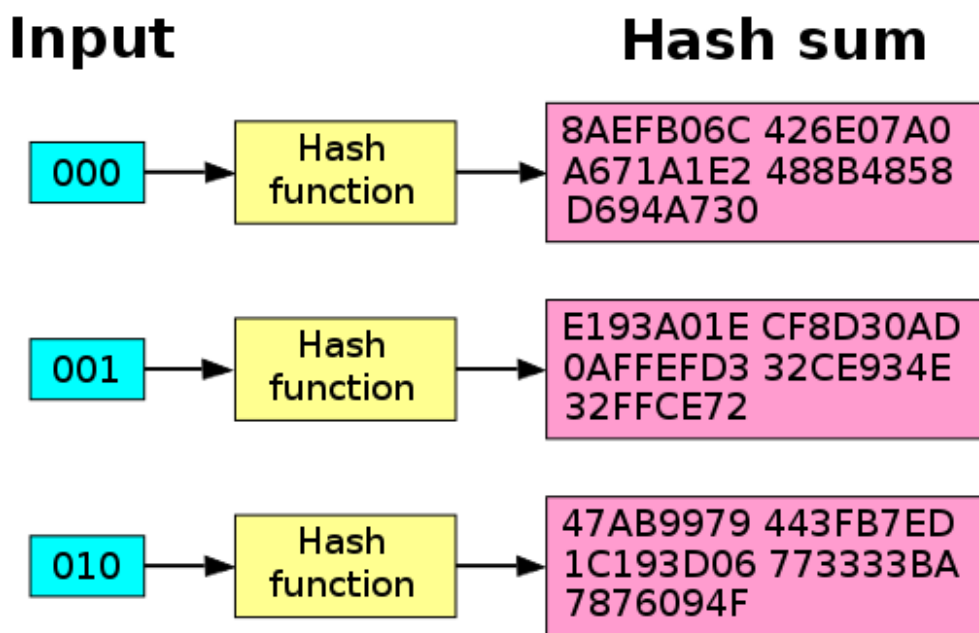


FIGURE 1.3: The "Avalanche Effect" of Hashing Data

(Source: [https://en.wikipedia.org/wiki/Avalanche\\_effect](https://en.wikipedia.org/wiki/Avalanche_effect))

Just to build upon the concept painted by a previous section, these **functions** are *one-way* and not easily *reversible*. Over the ages they were also known as *Message Digest Functions*. Something to note is that, despite the length of *Input* data, the function's *Output* is **constant** in size. This premise makes it harder to guess the size of the input message. [52].



### 1.3.1 Properties

Ideally, a CHF has to be deterministic, meaning that it must **always** result in the same *hash* (aka *fingerprint*). But there are some other requirements, that makes the computing life easier, and be extension, more secure.

A CHF has to be:

- Low on computing resources, resulting in quickening of hashing messages.
- As long as the CT provided is *unique*, its fingerprint must be unique as well.
- It **has** to be very difficult to forge a *different* PT that will have the same fingerprint.
- A *change*, no matter how small to the input message, **has** to reflect greatly on the resulting hash (*Avalanche Effect*).

### 1.3.2 Usage

The use of CHF is quite extensively used in Cryptography. They make the "glue" between *Information-Security* applications. They provide the solid foundation for *Digital Signatures* and *Message Authentication Codes* (aka: *MACs*). Their usage is limitless, and it is a complex task in finding all the uses that do benefit from CHF (or a simple hash function (HF), for that matter). Lengthwise, the output of those functions is between 128 and 1024 bits. One common use of HF, that do not need to be Cryptographic, is to detect duplicate data, or to check that the data is not accidentally tempered with (aka: *checksum*). [3]

### 1.3.3 Security Level of a CHF

#### Pre-Image Resistance

- **First Pre-Image Resistance:** It is computationally impracticable to pinpoint and predict the output hash, i.e for CT =  $y$  and PT =  $x$ , that  $h(x) = y$ , then finding  $x$ , given  $y$  must be very hard.
- **Second Pre-Image Resistance:** For a specified input, it is computationally infeasible to find some **other** input that produces the same output, i.e, given  $x$ , it is difficult to specify a second input  $x' \neq x$  such that  $h(x) = h(x')$ .

**Collision Resistance** This principle is *very* similar to *Second Pre-Image Resistance*, earlier covered. We say that a CHF is collision-resistant, if (*and only if*) it is very hard to find two specific inputs, that give out the same output. ( $a$  and  $b$  are the input of the CHF). [54]

$$a \neq b, \text{ but } h(a) = h(b)$$

The "Pigeonhole Principle" (for further read, Appendix A.1) demonstrated that in any HF, where the inputs provided, numerically outnumber the outputs of the given function will necessarily have such collisions. The harder they are to find, the more Cryptographically secure the HF is.

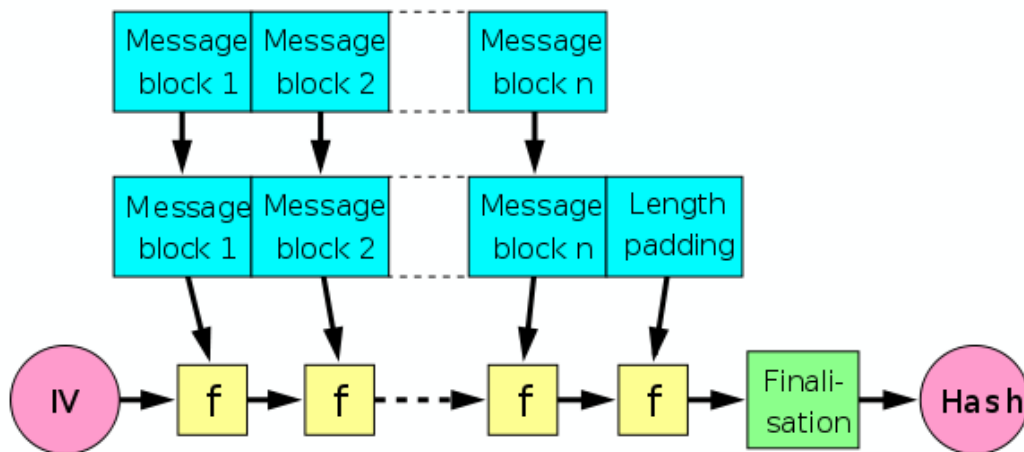


FIGURE 1.4: The Merkle–Damgård Hash Construction

(Source: [https://en.wikipedia.org/wiki/Merkle%E2%80%93Damg%C3%A5rd\\_construction](https://en.wikipedia.org/wiki/Merkle%E2%80%93Damg%C3%A5rd_construction))

### 1.3.4 Merkle–Damgård Hash Construction

This has construction model (as seen in Figure 1.4) was proposed by Ralph Merkle's Ph.D thesis back in 1979, while working with Ivan Damgård. Firstly, it was proposed as a padding scheme to offer collision-resistant in one-way data compression. They later came to realize that this could also be used for CHF, providing the latter properties. This design became popular by algorithms such as MD5, SHA-1 and SHA-2.

Firstly, for this function to work, the input message is divided into multiple "chunks" of bytes of *pre-fixed* number (*aka: MD-compliant padding*). The PT is most commonly spitted 512-bit ( $2^9$ ) blocks. This is known as "Length Padding" (*aka: Merkle-Damgård Strengthening*). This is a **vital** step for the algorithm, for the fact that is **cannot** handle inputs that are **not** a *pair* number. Needless to say, that this algorithm requires **two** distinct inputs, in order to work. To start computing the hash, the first padded block, gets compressed and hashed using the IV as the second input, it is an application-specific fixed number. After the first round, the IV is useless, since, the output of the previous round is fed as input with the next block of PT. This rounds are looped until all the blocks of the PT are used. After the last result, there is no guarantee, that it is the correct size, so, it gets *padded* with zeros, as needed. To harden the security of the hash even further, the (freshly padded) last result is sometimes fed through a "Finalization Function". It has several purposes, for example, to compress a sometimes, big result into a smaller, more manageable, hash size. More often than not, this so called Finalization function, is just a for compressing.

### 1.3.5 Sponge Hash Construction

The construction method described in the previous part, has a flaw, that can become pretty severe. Supporting only some specific length, and having to pad (or *inflate*) the inputs to size is problematic. Scientists got together and developed an algorithm that could take as input a *stream* of bits, by definition, any length of bits. Likewise, providing an output of any desired length. The *Sponge Construction* is used in many aspect of Cryptography, ranging from Pseudo-Random Number Generators, all the way to stream ciphers. (As seen in Figure 1.5).

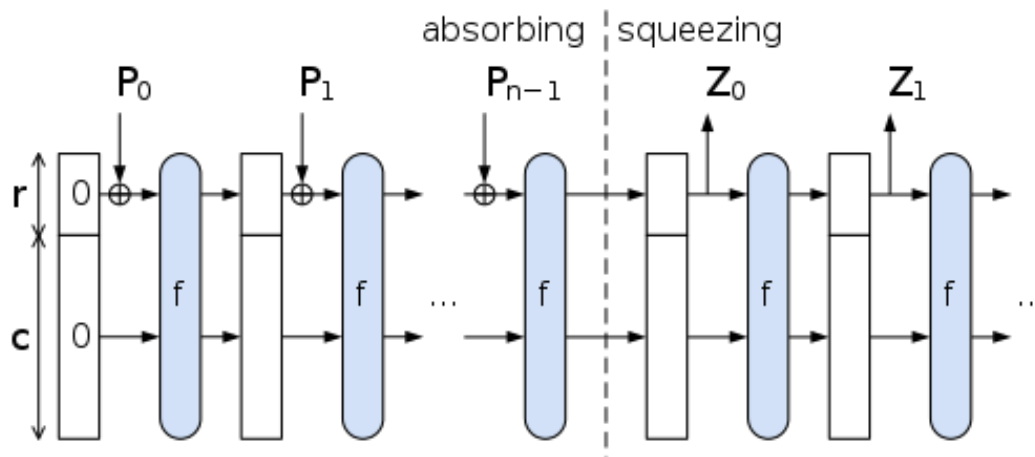


FIGURE 1.5: The Sponge Construction

(Source: [https://en.wikipedia.org/wiki/Sponge\\_function](https://en.wikipedia.org/wiki/Sponge_function))

In the figure 1.5,  $P_i$  are blocks of the input string, and  $Z_i$  are the hashed output blocks. A Sponge function is built from *three* components:

- A state memory, denoted as  $S$ , containing  $b$  bits.
- A function denoted  $f$ , such as  $\{0,1\}^b \rightarrow \{0,1\}^b$ .
- A padding function, denoted here as  $P$ .

In order for the Sponge function to work:

- $S$  is divided into two sections. Denoted as  $r$  (**the bitrate**) and  $c$  (**the capacity**).
- $f$  produces a *Pseudorandom Permutation* of the  $2^b$  states from  $S$ .
- $P$  appends enough bits to the input string, so that it is a *whole multiple* of  $r$ . By doing so, the input string is segmented into  $r$  sized blocks.

The *Operation* order is as follows:

- $S$  is initialized to Zero.
- For each  $r$ -bit block, denoted  $B$ :
  - $R$  is replaced with  $R$  XOR  $B$  **with the use of bitwise XOR**.
  - $S$  is replaced by  $f(S)$ .
- This steps are *repeated*, until all the of a padded input string are processed ("*aka: absorption phases*"). Then, the Sponge function output is ready to be produced.

"Squeezed Out" phase:

- Output the  $R$  portion of  $S$ .
- $S$  is replaced by  $f(S)$  unless output is filled up.
- If less than  $r$  bits remain, then  $R$  will be truncated. Meaning that only part of  $R$  will be outputed.

The *Sponge metaphor* is mostly used [34]. But another metaphor describes the state of memory as an "entropy pool", with the inputs "being poured into the pool", and the function ( $f$ ) referred to as "stirring the entropy pool" [62].

### 1.3.6 HAIFA Hash Construction

HAIFA Hash Construction (*wsf: Hash Interactive Framework*) is another **modern** design for the creation of HF. It stands as an alternative to Merkle-Damgård construction, providing cover from *length extension attacks* and some other weaknesses. It was originally designed by *Eli Biham* and *Orr Dunkelman* in 2007. Common applications of HAIFA, are HF like **BLAKE** and **ECHO** algorithms. [21]

### 1.3.7 Wide-Pipe and Narrow-Pipe

Building an application with the Merkle-Damgård construction, results **always** in a *narrow-pipe* hash design as the size of the *hash* output is the same as the size of *internal state*. Some of the problems of this design is *length-extension*, *multicollisions*, prone to *long message attacks* and *generate-and-paste attacks*. It is also worth noting that those applications cannot be parallelized. Scientists came up with the evolved design named *wide-pipe construction*, named after the large internal state size. Such designs are the *Sponge* and the *HAIFA* construction.

### 1.3.8 Common Cryptographic Hash Algorithms

#### MD5

Ronald Rivest in 1991 designed this algorithm. And as a direct replacement to MD4. Later, in 1992, MD5 became known with the name *RFC1321*. While MD5 produce a fingerprint (*aka: digest*) of a mere 128 bits, while using 16 bytes as *block size*. A collision against MD5 can be calculated with modern hardware, in matter of seconds, this algorithm is not suited for Cryptography.

#### SHA-1

SHA-1 was designed as part of the U.S. Government's Capstone project. Although the original specification, recently under the common name of SHA-0 was published in 1993 under the name *FIPS PUB 180* in Secure Hash Standard, by U.S. Governmental Agency of National Institute of Standards and Technology (*aka: NIST*). Shortly after, it was withdrawn by the National Security Agency (*aka: NSA*). Publishing an improved version, named *FIPS PUB 180-1*, that became commonly known as *SHA-1* in 1995. It produced a digest of 160 bit long, while using 20 bytes as *block size*. Collisions in SHA-1 were produced using *Shattered attack*, so like the previous one, it is not very secure.

#### RIPEND-160

This is a **family** of CHF (*wsf: RACE Integrity Primitives Evaluation Message Digest*). They were developed in 1996 by Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. It was based from MD4, but had similar performance with the SHA-1. While sharing the **same** properties with SHA-1 (160 bits/20 bytes), to this day, it has **not** been broken.

#### Whirlpool

In 2000, Vincent Rijmen and Paulo S. L. M. Barreto designed a CHF based on a much *modified version* of **Advanced Encryption Standard** (*aka: AES*). As for it's output, it produces a 512-bit long hash digest, while using 60 bytes as the size of its blocks.

## SHA-2

The Secure Hash Algorithm 2 (*aka: SHA-2*) is a CHF designed by the NSA and published in 2001. It maintains the the Merkle - Damgård structure. It consists of two hash algorithms: SHA-256 and SHA-512. SHA-224 is a variant of SHA-256 with a different starting values and a truncated output. The SHA-384, the less known SHA-512/224 and the SHA-512/256, are all in the SHA-512 family. SHA-512 is more secure than SHA-256 and is faster than SHA-256 on 64-bit coputers. Their *Output* and *Block* size are displayed in the table below. See 1.2.

TABLE 1.2: SHA-512 Family CHF

Algorithm	SHA-224	SHA-256	SHA-384	SHA-512
Output size ( <i>in bits</i> )	224	256	384	512
Block size ( <i>in bytes</i> )	28	32	48	64

## SHA-3

The Secure Hash Algorithm 3 (*aka SHA-3*) was released by the NIST in 2015. The SHA-3 is a member of the Keccak family of algorithms. They were originally designed by Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. It uses the Sponge construction. The SHA-3 does provide the same output sizes as the previous SHA-2, but unlike this previous SHA-2, it offers the ability to provide a configurable output size. See 1.2.

## BLAKE2

The BLAKE2 algorithm is a direct improved version of the original BLAKE. It was announced in 2012 by Jean-Philippe Aumasson, Samuel Neves, Zooko Wilcox-O’Hearn, and Christian Winnerlein. It’s goal was to completely replace the widely used but broken MD5 and SHA-1 algorithms. When run on 64-bit (x64) and ARM architectures system, BLAKE2 is faster than SHA-3, SHA-2, SHA-1, and MD5 algorithms. Although, BLAKE and BLAKE2 have not been standardized as *SHA-3* has, BLAKE2 has been used in many protocols including the Argon2 password hash, for the high efficiency that it offers on modern CPUs. As BLAKE was a candidate for replacing SHA-3, the BLAKE and BLAKE2 algorithms both offer the **same output sizes** as SHA-3, including a configurable output size. See 1.2 as the results are much similar.

## BLAKE3

The BLAKE3 algorithm, the improved version of BLAKE2, was announced in 2020, created by Jack O’Connor, Jean-Philippe Aumasson, Samuel Neves, and Zooko Wilcox-O’Hearn. BLAKE3 is a single algorithm, in contrast to BLAKE and BLAKE2, which are algorithm families with multiple variants. The BLAKE3 is really a tweaked BLAKE2. Difference being that the number of rounds is reduced from 10 to 7. Internally, BLAKE3 is just a Merkle tree, and it supports higher degrees of parallelism than BLAKE2 does not.

## 1.4 Cryptographic Keys

In cryptography, a *Key* is a string of data that is used to **lock** or **unlock** a *HF* or a *CHF*. Imagine it as password (or a *passphrase*). The length of the key is directly related to how secure it is. This is the major reason that a very large majority of *certified* internet sites, require a certain length as a password. This is indeed a *double-edged sword*, because a *malicious* user, who wants to break a password, knows the lower bound of the *input* size. Luckily, the website creators prompted the *legitimate* user to include in its password, some special characters and capital letters. The whole point is to make the guessing of the password, as difficult and time-consuming task as possible. If a user's passwords *breaks* in a million of computing years, it will **not matter** a thing, as the human life span is much smaller.

### 1.4.1 Kerckhoffs' Principle

The receiver of a message must know two things, in order to decode the message passed:

1. The **Algorithm** used for the encryption.
2. The **Key** in order to unlock the function.

Kerckhoffs' Principle states that: "The security of the *encryption scheme* must depend only on the secrecy of the Key, and **not** on the secrecy of the Algorithm". The reasoning being, that algorithms are not subject to change. They are even built on hardware, making them much more difficult to upgrade. No one is going to build a *cryptographic system* for just two (or *even three* users. Every algorithm is used by **millions** of users worldwide. That does not mean that the more users a specific system has, the more insecure it become. This is **madness!** To add to that, the same *old* algorithm is subject to be used for a **really** long time. So, the single point that provides the security, is indeed the secrecy of the **Key**.

## 1.5 Symmetric-Key Algorithms

Leaving the fancy words aside, this is often called as *"*. This is the simplest of concept. Having a **single** encryption (and decryption) key and sending through the channel. The two parties have *the same* key, hence the **Symmetric**, and it is given to them at the start of the transaction by a *third* party (*aka: CA*). This latter one is the single point of failure, as to whether it is a **trustworthy, non-compromised** member, and does not interfere with the transaction other than the simple task of "**Key Distribution**". (As seen in Figure 1.6).

### 1.5.1 Asymmetric Cryptography

For a number of years and up to the mid-1970s, a "Symmetric" encryption model was more than enough. But in order to keep up with the rise of the Computer's crude power, computer scientists concluded that in order to ensure the privacy of the communication, and by extension, the transmitted information, it was time to develop another more complicated model. A *Key pair* is used instead of a **single** key. The message is encrypted with one key, and on the other side, decrypted with the other, by the second party of the transaction. (As seen in Figure 1.7).

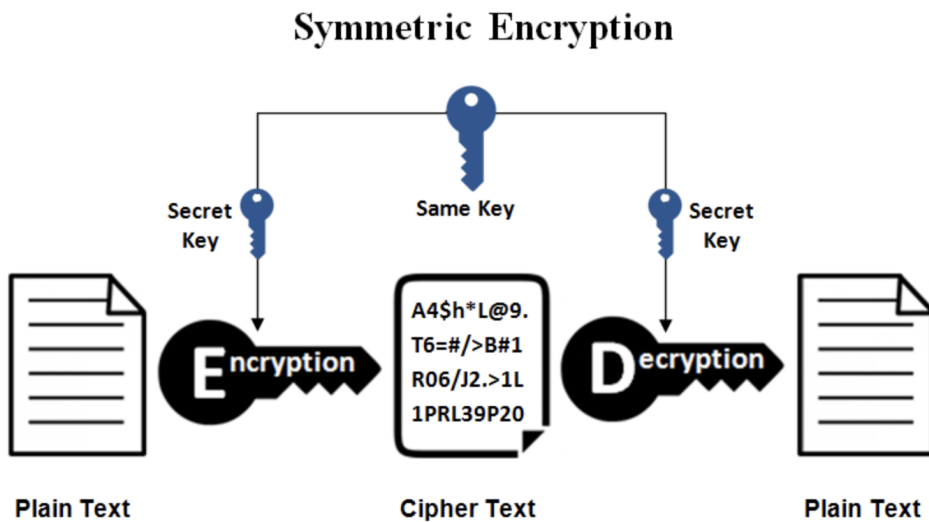


FIGURE 1.6: Visual Representation of Symmetric Cryptography  
(Source: [https://miro.medium.com/max/3372/1\\*bbCyiW35hBU3GiaiF4Qcmw.png](https://miro.medium.com/max/3372/1*bbCyiW35hBU3GiaiF4Qcmw.png))

(Note: In some books or Papers, like [52], they do *not* use the term of **Assymmetric**, but rather **Public-Key Cryptography**).

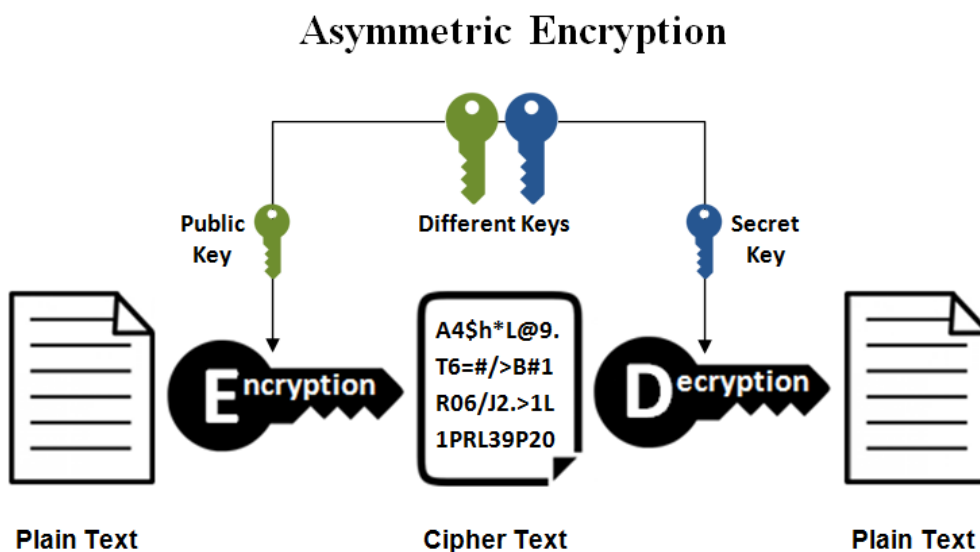


FIGURE 1.7: An Example of Asymmetric Cryptography  
(Source: <https://www.ssl2buy.com/wiki/wp-content/uploads/2015/12/Asymmetric-Encryption.png>)

## 1.5.2 Asymmetric vs Symmetric Cryptography

As already noticed, Asymmetric and Symmetric Cryptographic *models* use keys differently. A single key for encryption **and** decryption, for Symmetric models, versus two **separate** keys, for those two purposes, in Asymmetric ones. Right off, the Asymmetric model produces more **secure** algorithm than Symmetric model does.

But because of her *more complicated* nature, it is **slower** than a good Symmetric encryption algorithm. Sometimes *way to slow* for **many** applications. For that reason, many of today's Cryptosystems (TLS, SSH, *etc*), use both encryption models. The most often scenario is to use *asymmetric* model for securely exchange a secret *key*, that is will then be used for the *symmetric* model.

### 1.5.3 Long Term or Single Use

Besides the use of a Key, one other property they share, is for how long a specific Key will remain valid. A long-term key is referred to as static or archived, while others are used for a **single** session, and is called *Ephemeral*. Most *key types* are designed to last for long **crypto-periods** of about one to two years. For further read [59].

### 1.5.4 Hybrid Cryptosystems

Some things are not black or white, and a mix is possible. Cryptography is one of the many. As stated before, it is common to use an asymmetric *key-exchange algorithm* to encrypt (*and exchange*) a **symmetric key**. Hence, hybrid.

#### Diffie - Hellman Key Exchange

This scheme was published by Whitfield Diffie and Martin Hellman in 1976. It combines the two keys of the asymmetric model, to form a new key, and then transmit the information. Traditionally, secure (*and encrypted*) communications between two parties, would require at the start to exchange keys by some trusted means. This algorithm enables us to exchange keys securely, through an unsecured communication channel.

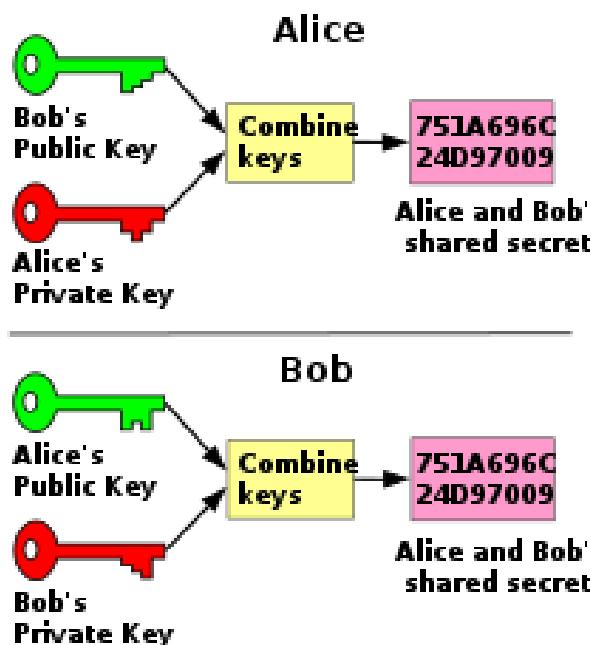


FIGURE 1.8: Diffie - Hellman Key Exchange

(Source: [https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman\\_key\\_exchange](https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange))

In 2002, Hellman suggested the algorithm be called Diffie-Hellman-Merkle key exchange in recognition of Ralph Merkle's contribution to the invention of Public-Key



Cryptography, writing:

"The system...has since become known as Diffie–Hellman key exchange. While that system was first described in a paper by Diffie and me, it is a public key distribution system, a concept developed by Merkle, and hence should be called 'Diffie–Hellman–Merkle key exchange' if names are to be associated with it. I hope this small pulpit might help in that endeavor to recognize Merkle's equal contribution to the invention of public key cryptography." [37]

## 1.6 Digital Signatures

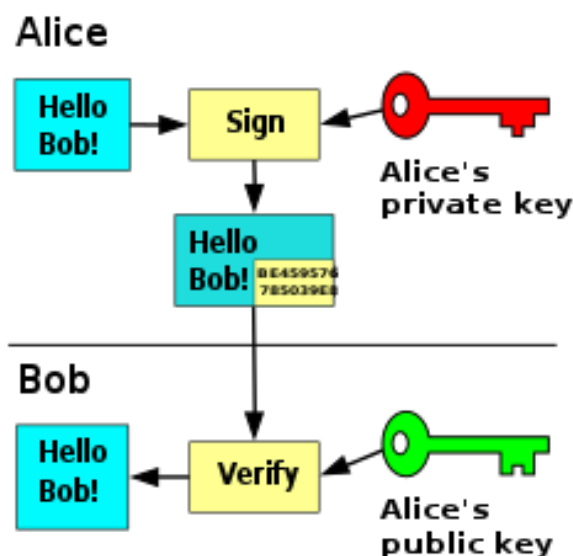


FIGURE 1.9: A Visual Representation of the Process of Digital Signatures

(Source: [https://en.wikipedia.org/wiki/Digital\\_signature](https://en.wikipedia.org/wiki/Digital_signature))

As its name implies, those are signatures that attach to a *electronic* document or message, to verify its authenticity. They are also known under the name of *Electronic Signatures*. They provide two main roles:

- The message (or document) was indeed created by the one who signed it (Authenticity).
- The signed message or document was **not** altered in transit, through the *Communication* channel (Integrity).

This concept is a little old. it dates back from 1976 where Whitfield Diffie and Martin Hellman first described the notion of a **Digital Signature Scheme**. Very soon after, Ronald Rivast, Adi Shamir and Len Adleman invented the *RSA* algorithm, that was later used for Digital Signatures. (See figure 1.9)

## 1.7 Public Key Infrastructure

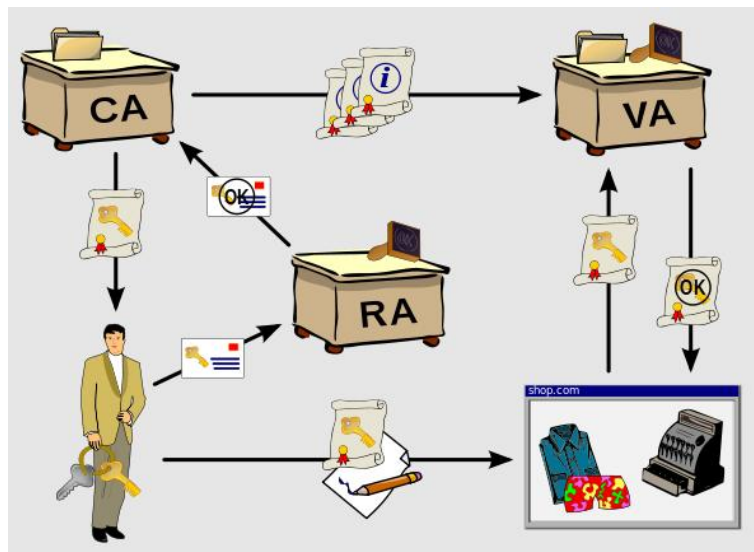


FIGURE 1.10: A Visual Representation of the Process of a PKI  
(Source: [https://en.wikipedia.org/wiki/Public\\_key\\_infrastructure](https://en.wikipedia.org/wiki/Public_key_infrastructure))

Public Key Infrastructure (*aka: PKI*) is mostly a set of roles, policies, needed to create and manage *Digital Certificates* and by extension public-key encryption. Most often than not, even hardware and software are included in a **PKI scheme**. Its main purpose is to facilitate the Electronic transfer of secure information. Some common usages for PKI are:

- E-commerce
- Online Banking
- Confidential E-mail

But there is no limit to its usage, and it is quite a task, to pinpoint all the use a PKI has (or *potentially has*). In essence, the use of a PKI is to *bind* public keys with the identities of entities (*like Peoples or Organizations*). This process is performed by the CA, but in order to finally *certify* an entity, the latter one must make a formal request in the local RA, and if it does approves, it communicates with the CA and, the entity is now certified. All that is left to do, is to send the *certificate* over to the entity, in order to proceed with the transaction. In the meantime, the CA send to the VA, a *batch* of approved certificates. As he entity is authenticated and proves that authentication by sending his own certificate to the server. On the other side, the server request a valuation by sending the certificate over to the VA for validation. If everything goes according to plan, and the entity is legitimate, and not malicious, forging a certificate, the transaction (and the *axiomatic*) request is accepted. Since a picture is worth a thousand words, the **whole** process is illustrated in figure 1.10.

*Note: The X.509 Standards (See appendix A.2 for more information) define the most commonly used format for Public Key Certificates.*

## Chapter 2

# Honorable Mention: Kerberos

## 2.1 Introduction

We owe as Computer Scientists, to invite into the spotlight, a system that held our security needs with an *iron fist*, for such a long time. (Of course we designed every bit of it, but this is not the *point*!) From a *Network Security point of view*, it was (and most importantly, **will be**) the **cornerstone** of Network Security, and for that reason, it started a movement for the computer community to develop many *other systems* from its footsteps, like *SSO, SRP, HIP, ... etc.* Many companies use this system to secure their network and computer systems. The first Kerberos was written in 1988 by the *Massachusetts Institute of Technology (aka: MIT)*. It was primarily developed to secure a campus-wide distributed computing environment that was used solely for educational needs. This became afterwards as **Project Athena**. Despite its *advance age*, it is used by companies even to this day. At the time of writing this *M.Sc thesis*, there are only five implementation versions available. It would be such a waste not to find a **Quantum Safe** way to *upgrade and finally implement* this technology of *computer-network authentication protocol* to the new era of Quantum Computers.



FIGURE 2.1: The Kerberos System  
(Source: <https://web.mit.edu/kerberos/>)

## 2.2 The Kerberos System

As we all ready cited in the previous section, this is a computer-network authentication protocol that allows *network nodes* to communicate securely with each other over a non-secure network. The way it accomplishes this task, is by using a *symmetric-key*

*cryptographic model* (and requiring a third party in doing so). Providing the *legitimate user with a ticket*. It was designed as a *client-server model* as it provided an error free way for **mutual authentication** of both, user and server. Currently, there are five releases of the Kerberos System (*just counting the **stable releases***). They are as follows:

TABLE 2.1: Releases of the Kerberos System

<b>Kerberos Version</b>	<b>Usage</b>	<b>Availability</b>	<b>Developer</b>
Kerberos Version 1	Project Athena	Only Inside MIT	MIT
Kerberos Version 2	Project Athena	Only Inside MIT	MIT
Kerberos Version 3	Project Athena	Only Inside MIT	MIT
Kerberos Version 4	Project Athena	Worldwide	Steve Miller/Clifford Neuman
Kerberos Version 5	Unknown	Worldwide	Clifford Neuman/John Kohl

*P.S: Versions of Kerberos one to three are based on a protocol named **Needham-Schroeder Symmetric Key Protocol**.*

There is very little to be known about the first three versions of Kerberos other than they are exclusive for MIT use only. Even their release year are not accurate. This may be related to the fact that those three versions were not widely available and existed solely in MIT. By February 1988, we were in the *fourth alteration of Kerberos (Version 4)* so, when was the original (Version 1) of this system was made? According to MIT, Kerberos was indeed developed in 1988. So we speculate that this system was developed in January of 1988? The original release date, is of no importance in the real world, other than from a purely Academic standpoint. The fifth release of Kerberos dates back to 1993, and it is getting upgraded to this day. The most recent (*stable*) release (in the time of writing this thesis) is *Version 5, Release 1.20* that took place in 26 May 2022. In 2005 Kerberos Version 5 became obsolete, but after the upgrades Internet Engineering Task Force (*aka: IETF*) done, it became secure once more. The same institution, published a draft in 2022, and stated that Kerberos is Quantum safe. [58]

### 2.2.1 Workings of a Kerberos System

As we saw in Figure 2.2, there are two distinct systems that come into play into a single *Key Distribution Center (KDC)*. We have:

- The Authentication Server (*aka: AS*)
- The Ticket-Granting Service (*aka: TGS*)

Before we dive deeper to what these systems do, let us first examine the course of action of a *client's login* to the system without **any Kerberos interaction**:

1. Firstly, the *user must provide a username and a password* on the *client's machine*, with all the security issues it implies (*keyloggers, sniffers... etc*). In reality, the client **transforms** the provided password into a **Key** for the symmetric cipher in usage. The most common techniques used are *one-way hash* or most of the time, the built-in *key scheduling*.
2. On the receiving end, the server **compares** the information provided, with a database copy. If there is a **match**, the login is a success, and if not the server sends out a reply to repeat the login process.

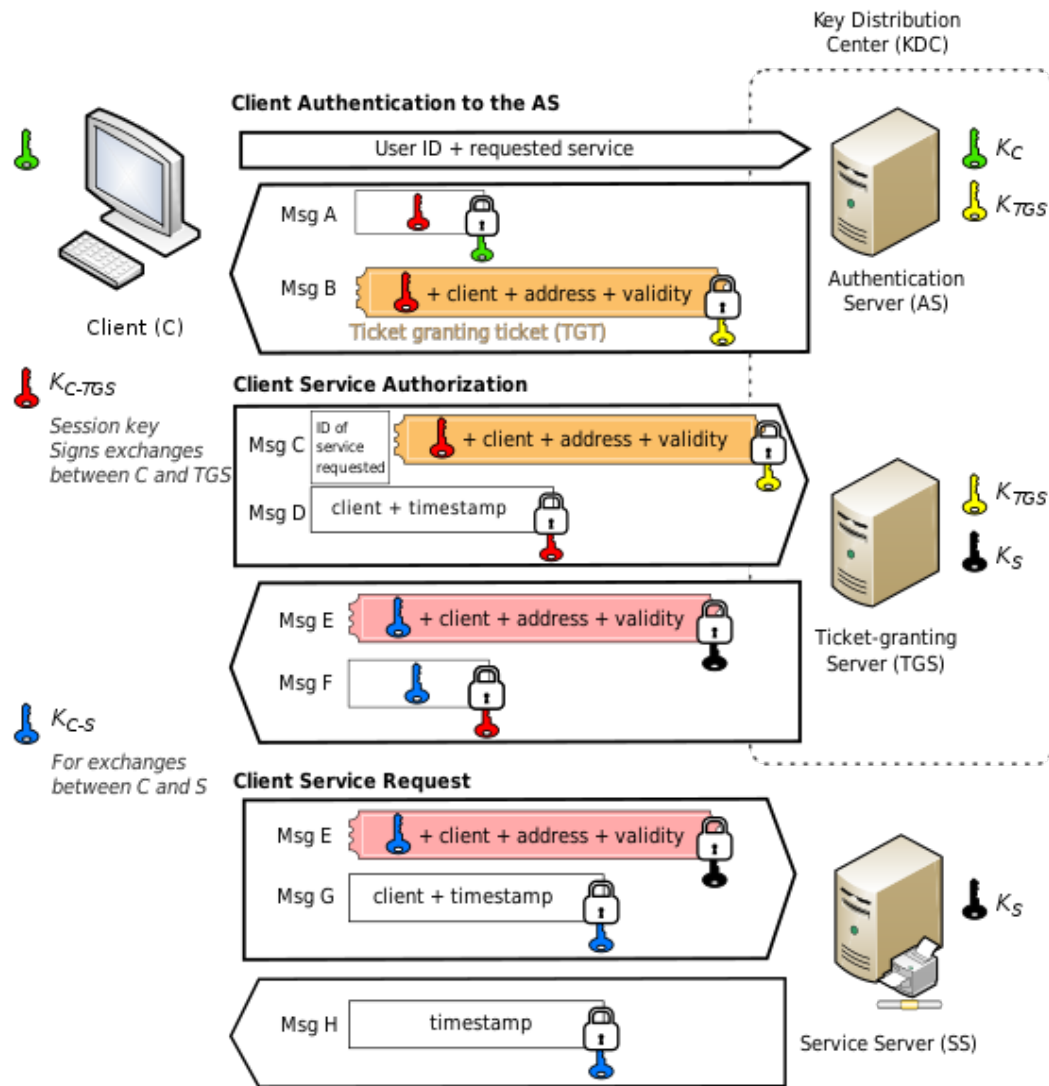


FIGURE 2.2: A Visualization of the Kerberos System  
(Source: [https://en.wikipedia.org/wiki/Kerberos\\_\(protocol\)](https://en.wikipedia.org/wiki/Kerberos_(protocol)))

Note: It is worth noting that the security relies that those two systems **are not compromised by any sort**, and the message is received **exactly** as it was sent.

Now let us see the changes that require in this simple procedure in order to **fully secure the transaction** using the **Kerberos System**. [66]. Firstly, a Kerberos is depicted as having three heads. This is not *just a mythological fact!* It is also to remind ourselves that there are three *systems* involved. (In reality, it has only two systems. but if we take to account the **secure server** the client is about to log in, then this last one is the **third and final system**). This is the reason it is depicted with *three heads* and finally a **three system** protocol. For this reason, we can now analyze its system functions:

### 1. Client Authentication

- (a) The client sends out a CT message, providing **only the user identification** (*aka: User ID*) to the AS of the Kerberos System.  
No password is sent in this phase!

- (b) The server that plays the role of AS, after receiving the message, makes a *query* in its database to find the *specific user that has this User ID*. If the User ID is **found**, then the AS **generates** the secret key by hashing the password value it found in the database. This is often handled by the **Active Directory** for Windows Servers. Likewise, the system sends out the following **two messages** to the client:
- Message A: It contains the **key** that will be used by the client to contact the TGS server. This is a simple Session Key. It is constructed by an amalgamation of the secret key for that specific user.
  - Message B: A **Ticket-Granting Ticket** (*aka: TGT*). It includes the following information:
    - Client's ID
    - Client's *Network Address*
    - The Client/TGS **Session Key**
    - The ticket's *validity period*
- All this message is **encrypted using the secret key of the TGS**.
- (c) After providing proof, (thanks to *TCP/IP*), that the *Client* has received **both messages**, the latter one attempts to decrypt *Message A* with the User's secret key that the User has on storage. If the *User's secret key* matches the *User's secret key* that the AS sent, then, he can successfully decrypt *Message A* and obtain the **Session Key** that will be used to communicate with the TGS. It is needless to say, that if the password the client provided **does not match** the password in the AS database, the decryption of *Message A* will fail, since *Message B* is encrypted with the TGS's secret key.

## 2. Client Service Authentication

- (a) Following the flow of the request, it is only logical that the Client requests a service from the TGS. In order to achieve such a feat, he sends out the following messages:
- Message C: This takes the full length of the depicted as Message B, adding the ID of the *User* and the requested service that he tries to accomplish. *Note that the Message B in the encrypted version and not the decrypted one!*
  - The second message, the *Client* sends out, depicted here as **Message D**, is composed solely of the *Client's ID* and a timestamp. It is worth noting that this message is encrypted using the *Session Key* from **Message A**.
- (b) After successfully receiving the **Messages C and D**, the TGS separates Message B from the rest of Message C. Since Message B is encrypted using its own secret key. Using the acquired the **Session Key**, it attempts to decrypt message D as well. It can now compare the **Client ID** of messages B and D, ensuring that they do match. If it does, then it is time for the request to pass on. The TGS sends the following messages:
- The first message the TGS sends out is called as a **Client-to-Server Ticket**. It is depicted here as **Message E**. This Message is encrypted using the requested server's **Secret Key** and it includes:

- Client ID
- Client's Network Address
- The Client-Server Session Key
- It's validity period
- The Message F contains the Client-Server Session Key and it is encrypted with the Client-TGS Session Key. A thing to note is that the legitimate Client has all the keys necessary for the decryption of this messages.

### 3. Client Service Request

- (a) After receiving **both** messages from the TGS. He has all the information necessary to contact the Service Server (*aka: SS*), and proceed with the secure transaction with the server that he requested. There are still three steps separating himself from the service the SS provide. This time, the **Client** sends two messages. They are:
  - Firstly, he sends back the Message E as is. The receiver this time is the SS.
  - Like the Message D but instead of encrypting it with the Session Key for the TGS, it does it with the Session key for the Client-Server one.
- (b) In normal circumstances, the SS can successfully decrypt Message E that is encrypted with its own **secret key**. Using this newly acquired session key, the SS can now decrypt messages E and G. It looks for a match in the **Client ID**. If they do match, and normally they do, The SS sends a Message (depicted as H in the Figure 2.2) to the client that confirmed the client's identity and that the system is willing to serve the client's request. So the message composed solely with a *timestamp*, but is encoded with the Client-Server Session key.
- (c) The Client decrypts the confirmation message (Message H) and checks if the *timestamp* sent by the SS is indeed correct. If it is, then it's time to start issuing service requests to the SS.

This is a much longer process compared to the one that does not require the use of Kerberos. But this is what it takes to communicate securely over an insecure network. This elaborate procedure is not without vulnerabilities. First of all, the encryption algorithm that is used, is DES (*aka: Digital Encryption Standards*). This is a very outdated cryptographic scheme, and it is **not** very safe anymore, specially for the computer systems we currently have [2]. Not to even mention Quantum computers! The reason the **Kerberos** system is using DES, and not something newer like AES for example, is for compatibility reasons with many legacy systems. But this is by not means a deal-breaking vulnerability. By the way, we all ready cited that there is a bright sky for Kerberos. It has all ready been upgraded to meet the new standards. Another vulnerability that is all ready taken care of, is that there is an exploitable vulnerability that was possible for doing what is known as privilege escalation. In other words, this vulnerability allowed users to *elevate and abuse* their privilege up to a *Domain* level. But this was patched by *Microsoft* in November 2014. The patch name was **MS14-068**. A thing worth noting is that most operation systems other than *Microsoft* fixed the issue in a matter of days, or even less. But in *MS Windows* case, it took nearly **six months** to push a patch to her operation system.

## 2.3 Issues with a Kerberos System

- The first issue that needs attending is the fact that Kerberos is quite strict with its time requirements. This is analogous to say that **Both clocks** (client and server) have to be fully synchronized. If **both** clocks are not, the entire Authentication will fail. It is said that clocks just **cannot** be more than five minutes difference between those. In a real world scenario, Kerberos uses a set of NTP (*aka: Network Time Protocol*) daemons in order to have all clocks synchronized.
- Since all the Authentications happen inside a **Key Distribution Center** (*aka: KDC*), it is the one point of if an malicious user attacks the system. In that case the attacker can successfully impersonate **any** user of the system.
- Every network service provided, requiring a *different* host name, will require separate set of Kerberos keys. This makes visualizations and clustering, very complicated procedures.
- There is not a standardized way for providing administration on Kerberos systems. This varies between server implementations as a byproduct.
- The users that do use Kerberos **must trust** the Kerberos Distribution Server.

## 2.4 Kerberos And SESAME

I decided to put those two systems together under the same title, because they are not that different. These two security systems allow users to securely connect to a network node, requiring a single authentication to be made. They then grant the user a **ticket** that is used to access other applications on the network. There are two obvious differences:

1. SESAME is European based, and not American.
2. SESAME ticket contains Access Rights.

These systems are compatible with each other and they both use GSS-API interface.

### 2.4.1 SESAME

SESAME is an acronym for *Secure European System For Applications in a Multi-vendor Environment*. It is rather an improved version of Kerberos. In contrast to the Kerberos system, it provides:

- It uses an Asymmetrical Cryptographic scheme (*aka: Public-Key Cryptography*).
- It has a Role-based access control.
- It has **Separate** Authentication and Confidentiality Keys.

As the figure 2.3 illustrates, the *User* starts by connecting to a *User Sponsor*. This latter one is entrusted with the task of **Authenticating** the user to the *Authentication Server* (*aka: AS*) via *Authentication Privilege Attribute* (*aka: APA*). As soon as the US authenticates the user, it contacts the *Privilege Attribute Server* (*aka: PAS*) and if successful, the US receives a *Privilege Attribute Certificate* (*aka: PAC*). The PAC contains the user privileges allowing or forbidding specific applications to be executed. When a user



is ready to run an application, the US that he authenticated with, will contact the *Secure Association Context Manager* (aka: SACM). When that is the case, the client-side SACM will contact in tandem, the server-side SACM to exchange **User credentials**. The final step of the transaction is done by the server-side SACM that requests the *User's PAC*. If the user is permitted to request that specific service from the **Application Server**, then, the request is **granted**. In contrast to Kerberos, there is no tickets involved, but for compatibility sake with a Kerberos system, SESAME can be configured to run with tickets.

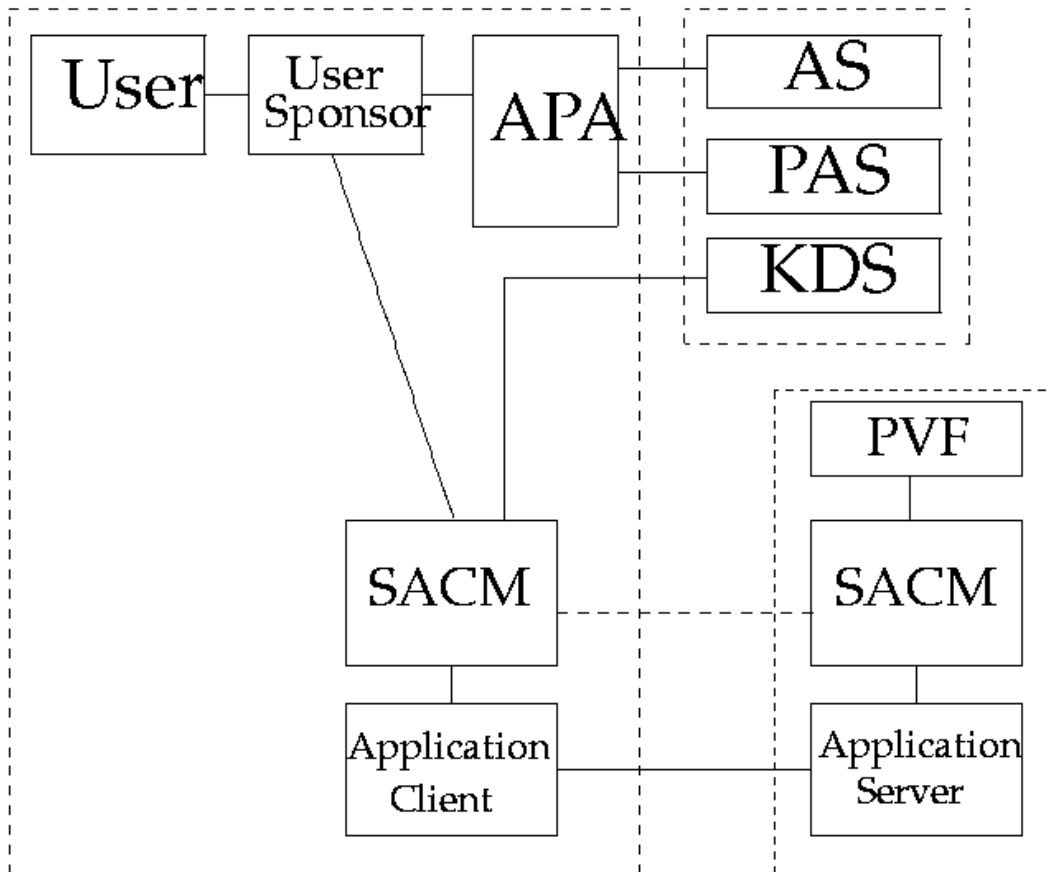


FIGURE 2.3: A Visualization of the SESAME System

(Source: <https://www.cosic.esat.kuleuven.be/sesame/matsulf/kerbses.html>)

This system is **not** very widely known according to my research, but who knows what the future will hold.

## 2.5 Conclusion Regarding Kerberos

Despite its problems, the Kerberos is one of the most used systems to this day [55]. And one of the reasons why, is the fact that it can work perfectly with symmetrical as well as asymmetrical cryptography. With its upgraded version, it is even Quantum safe. This is wonderful news, because there are many applications that require security over an insecure network. One such example is **cloud computing**.

## Chapter 3

# The Quantum Era

Since the famous "Double Slit Experiment" (see figure 3.1) in 1802 by Thomas Young, light have shown its peculiar nature. Scientists and, in particular, Physicists have partially found an explanation for this bizarre phenomenon. But in who hundred years, a plethora of things are still a mystery to be shed light on. Science is a self-correcting field, and, as new Scientists emerge, they search (and find) evidences that **partially** correct the initial hypothesis. The other way around is also possible, as it is a perfectly viable solution. Sometimes, the initial hypothesis gets shut down by the evidences or the lack of. This is a perfectly healthy outcome. Even way, the beliefs are being modified as we inch ourself in, to the truth. We have a fancy name for these occurring actions, it is simply called "Bayesian Reasoning". This is a **vital** part of the process, as the *human knowledge* evolves by moving forward, so does *Science*.

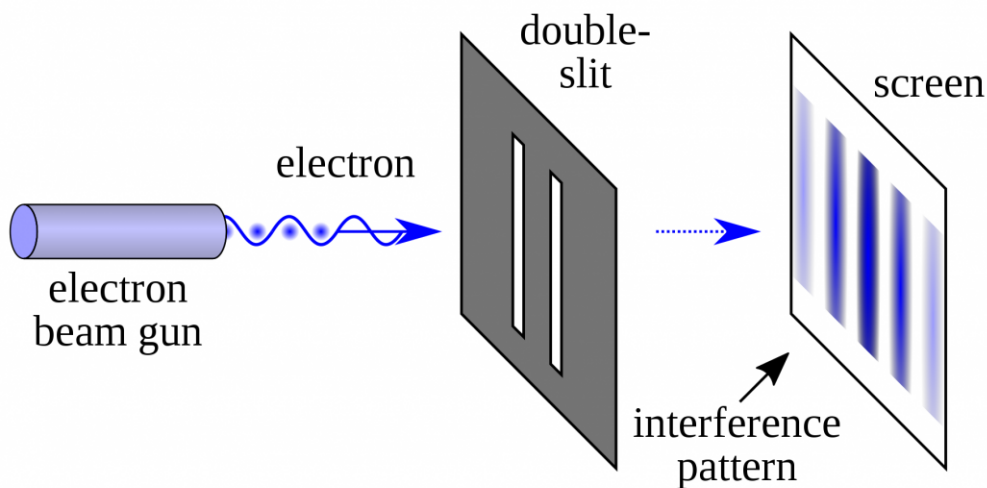


FIGURE 3.1: The Double Slit Experiment  
(Source: <https://worthknowingthat.com/wp-content/uploads/2019/04/double-slit-experiment-worth-knowing-that-e1555188078449.png>)

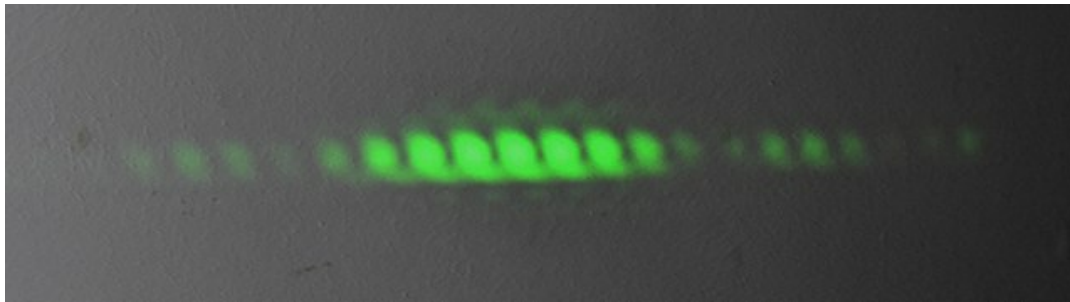


FIGURE 3.2: Green Laser passing Two Slits 0.4mm Wide and 0.1mm Apart

(Source: [https://en.wikipedia.org/wiki/Double-slit\\_experiment](https://en.wikipedia.org/wiki/Double-slit_experiment))

### 3.1 The birth of Quantum Mechanics

All this mystical fog of unanswered question, baffled scientist for over two hundred years. In 1911, Ernest Rutherford proposed another experiment that raised more eyebrows. It became known as "The Rutherford Experiment" and it was performed by two brilliant minds. That of Hans Geiger and Ernest Marsden (see figure 3.3). In this experiment, monochromatic light passed through a narrow passage and slammed a single piece of thin Gold Sheet foil secluded all around a circular screen. As light passed through, multiples patches of light was observed onto the screen. Logic suggested that just a single patch of light has to be observed, but the evidences showed that it did not. Ernest Rutherford couldn't be the father of Quantum Mechanics. A hundred years ago, Thomas Young did.

(For more information give the Appendix C a read.)

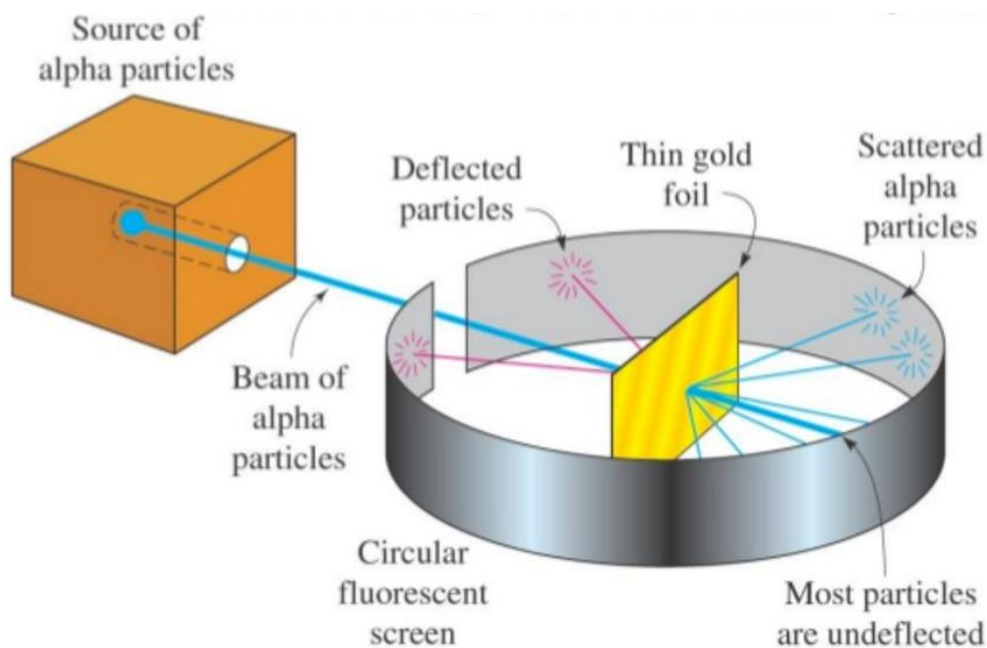


FIGURE 3.3: The Rutherford Experiment  
(Source: <https://classnotes.org.in/wp-content/uploads/Rutherfords-Experiment.png>)

### 3.1.1 The Old Quantum Mechanics

Prior to 1900, Quantum Mechanics was pretty much a black box sector. There was not enough knowledge in Physics, as a whole, to test **any** kind of hypothesis. The inner workings of Light was a unsolvable mystery. Until someone could get hold conclusive proof of the absorption and emission in a some solid body. That someone was Max Plank and his discovery known as the **Planck's Law**. He introduced us with the so called "**Quantum of Action**" (aka: The **Planck Constant**).

### 3.1.2 The New Quantum Mechanics

As you all understand, the field Quantum Mechanics was a bumpy road. All that was due to the nature of this rapidly evolving field. The fact that it was based on microscopic interactions, didn't help ether. In 1925, Erwin Schrödinger did a breakthrough discovery that turned the field around on it's head. A year later, he published his theory. It was a landmark discovery that, signaled the dawn of a brand new understanding of quantum states and phenomena. This discovery was so great that in 1933, his work won him the **Nobel Prize** in Physics. This discovery has a Mathematical form (aka **Schrödinger Equation**), in simple terms, is a *linear partial differential* equation that describes the **wave function** (see Appendix C.2) into a quantum-mechanical system.

Since I'm not a Physician, I will no even try explain the Mathematics that lurk around. I really believe that explaining the Physics behind it is beyond the scope of a **Computer Scientist** to explain. But...if someone want (and is interested) to look up all the Mathematics and Physics behind it all, he/she more than welcome to dig into. I humbly recommend the book [65].

## 3.2 The Bits and the Qbits

The inner working of Physics are quite intriguing, but, let's zoom out and give focus on the information passing Data between systems. The Quantum theory apply to them as well. In non-Quantum Computing Systems, the tiniest piece of information, is called a Bit. As already said, they hold a value of logical zero or one. In the quantum world, there are Bits too (called Qbits) but unlike there electrical analogous, they can be have one of the possible three values rather than two. The value of *zero* or *one* make the Qbits behave like a normal *Bit*. But the interesting part is the third possible value. It's an amalgamation **between** one and two. This might sounds like crazy talk, but in fact there is a completely logical explanation. A Qbit is not two things in one. It's a single particle and this so called *third* value, is as a matter of fact, just the probability of both the above two. This latter quantum state is called **Superposition**, and it is mind blowing, but not illogical. (see Appendix C.3 for more) [33]

### 3.2.1 Bits and Qbits

As we all ready cited, a Qbit can be in one of three values. For the sake of example, let us see what happens when we combine two bits.

TABLE 3.1: Possibilities with two Bits

Possibilities	Value of Bit 1	Value of Bit 2	Number
1)	0	0	0
2)	0	1	1
3)	1	0	2
4)	1	1	3

As we saw in the table 3.1, we have only four possibilities with two Bits. No matter if there **binary bits** or **quantum Qbits**. Generally speaking, for a Bit, The possibilities it can have are de-numbered by  $2^n$ . And in general, the maximum number it can represent is  $2^n - 1$ , for  $n$  represents the amount of binary bits we have in our disposal and the number two represents the bit's base. In this example we have  $2^2 = 4$  possibilities and the maximal number for representation is  $2^2 - 1 = 3$ . So, one's might question the advantages of a **Qbit** versus the **traditional** ones.

The advantages lies in the way *binary* bits are used. They can hold a value of either *One* or *Zero*, and **nothing** in between. Whereas a Qbit can be used as a regular bit, but it can hold **every** possible value **between** One and Zero. Mathematicians say that **Infinity** is found *between* One and Zero.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

(in this formulae  $\alpha$  and  $\beta$  are complex numbers and represents the probability of "1" and "0").

In the picture 3.4, we can see how **Superposition** is possible. For further read, I suggest giving [14] a read.

### 3.3 Quantum Entanglement

One of the most mind blowing feature of a Quantum object, is the phenomenon called "**Entanglement**". No ones *fully* understand the reason it functions. We only find solid evidences this happens. Even some very smart individuals tried to fully explain the phenomenon, but it still remains a paradox in the Quantum realm. The first Physicist that made the observation of Quantum objects behaving the specific way, was Erwin Schrödinger. After him Albert Einstein, Boris Podolsky and Nathan Rosen, have tried to give an explanation. To this day, the explanation given is not complete, thus it remains a paradox. The EPR paradox (short for the three Scientist: Einstein, Podolsky and Rosen) is that when two (or more particles) "link up" in a way, no matter how far apart they are, their states remain linked. It means they share a common and unified quantum state. In practice, by observation one of the particles (that is Entangled), we are able to can predict information about the other particles, regardless of the distance between them. As a consequence, any action to one of these two particles will invariably impact the others one in the Entangled system. (See figure 3.5). [17]

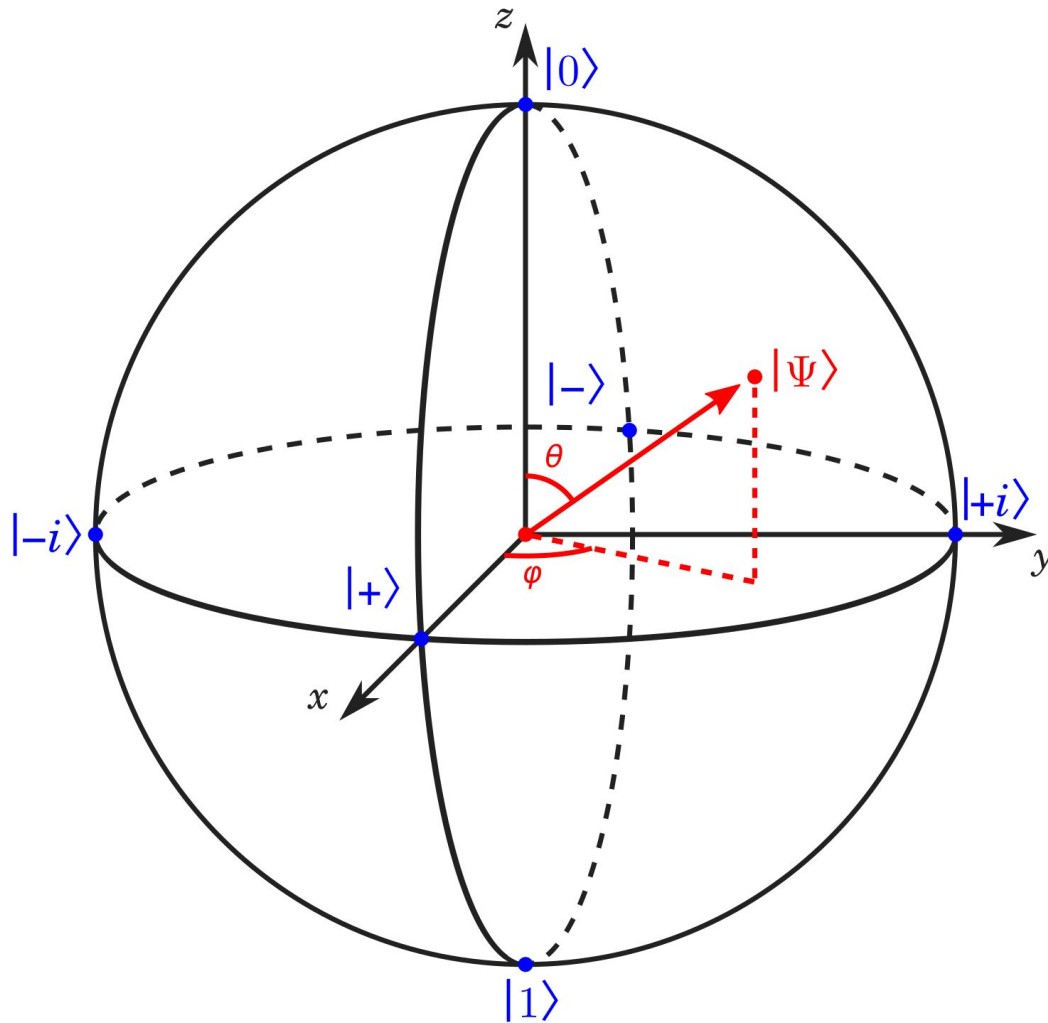


FIGURE 3.4: The Bloch Sphere

(Source: [https://en.wikipedia.org/wiki/Bloch\\_sphere](https://en.wikipedia.org/wiki/Bloch_sphere))

### 3.3.1 Quantum Teleportation

Earlier, we talked about **Entanglement**. This section is entitled *Teleportation*. I assure you, those, quite complicated, two things have indeed something in common. In-fact in order to achieve a successful **teleportation** of **any** sort, the **Source** and **Destination** particles **have** to be entangled with each other. [56]

There are three types of teleportation:

- Teleport the object instantly from one location to another, by a "loophole" in Space-Time.
- Teleportation that requires a "disassembly" of the object, send the "pieces" over and "reassemble" them afterwards.
- Teleportation by scanning the object and transmit *only* the **instructions** on location. And using **different** molecules and atoms, putting back together the object.

Out of the *three kind* of teleportation, only the last two are possible. The *first* one requires a hefty bit of magic, and by today's knowledge basis, **it is not possible!**

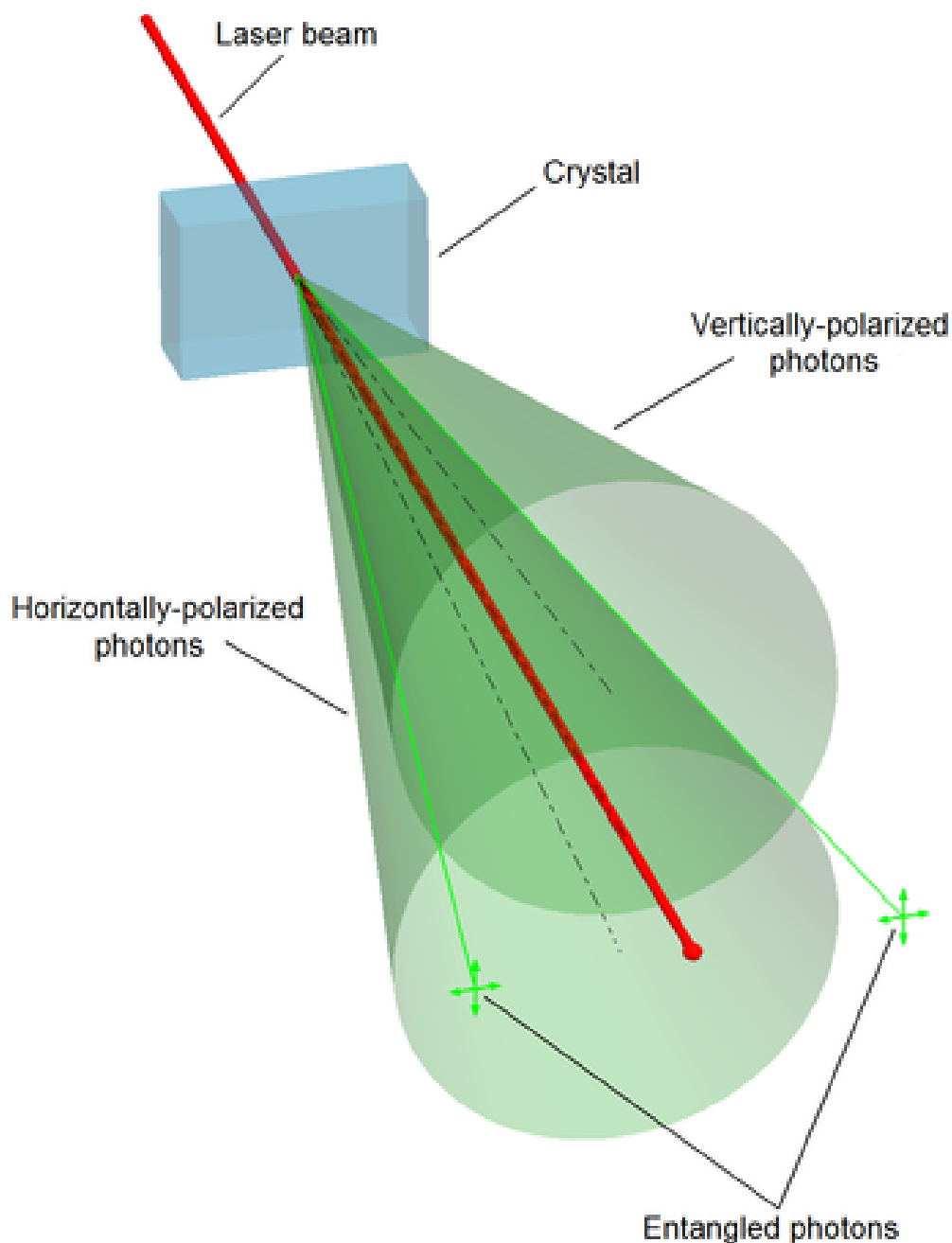


FIGURE 3.5: A Quantum Entanglement Experiment  
(Source: [https://en.wikipedia.org/wiki/Quantum\\_entanglement](https://en.wikipedia.org/wiki/Quantum_entanglement))

Currently we have successfully managed to **Teleport** quantum particles (mostly photons and electrons), over a distance of a hundred Kilometers [67]. This is a **really** big deal, since it is **extremely hard** to keep a pair of entangled particles long enough. So basically, *Star Trek* kind of teleportation, is still totally Science-Fiction.

A completely logical question someone might ask himself is "What happens to the **original** object that undergo teleportation?", for that specific question I suggest reading Appendix C.4 a read. For further examination of the "how" and "why" questions, please read [8].

PS: The third kind of teleportation does indeed sound like cloning.

### 3.3.2 Quantum Cloning

In Physics, we define as cloning the procedure that succeeds at copying the **exact** state of every molecule, atom or even electrons for that matter. Since we do **not** know the exactly the position, spin and momentum of every particle inside the object that we are trying to clone, and it is **impossible** to know all this things at every given moment, **Perfect Cloning is Not Possible** [46]. (For the some of the proof regarding Quantum Cloning, please see Appendix C.5)

But all is not lost. Researches suggests that a *Non Perfect Clone* can in fact exist [13].

## 3.4 Putting it all Together

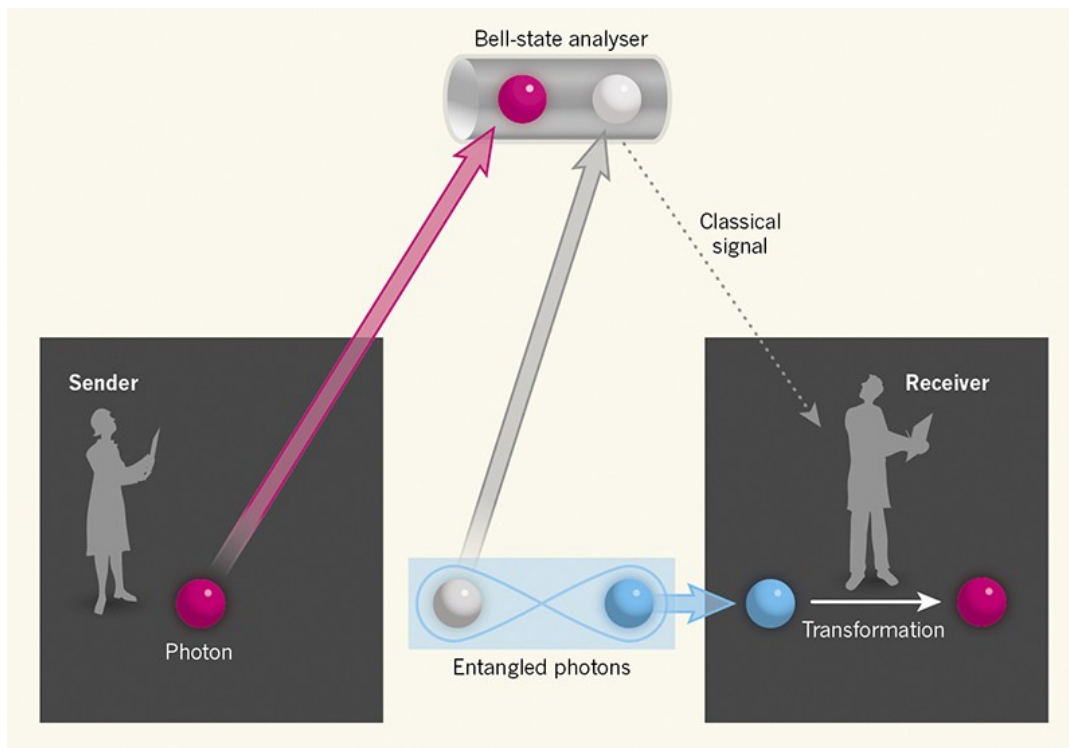


FIGURE 3.6: A Quantum Teleportation Experiment

(Source: <https://media.nature.com/lw800/magazine-assets/>)

I know what is the question out there. How it is a *teleportation* if the red-ish partible is not transferred over? To answer it simply: "It can't be transferred over!". As the figure 3.6 illustrated and for the sake of simplicity:

1. Firstly, we start by **Entangling** *two* Quantum particles. In this illustration, Photons are used. (*And mostly Photons are widely used.*)
2. We route **one** of the Photons, to a device together with one of the "senders" Qbit.



3. The device *copies* the state of the Qbit, onto the "blank canvas" particle. And since it is **entangled** with **another** Photon, every state change is copied over, without looking at the second Photon (as this will result the *collapse of the Wave function, and the destruction of the Superposition!*)
4. In the mean time, the **Second** Photon is transmitted to the *Destination*. (The Blue Photon in the the illustration 3.6)
5. Since the *two* Photons are entangled. The state of the "white" Photon, will be transferred to the "blue" one.

Ones might ask what happens to the "white" and "red" Photons after teleportation, just read Appendix C.4.

### 3.4.1 A State Of The Art Product for Quantum Key Distribution

In the *previous section*, we began to *analyze* the concept of achieving a secure communication between *two parties*. In *theory*, the *illustration 3.6* paints the picture of "How QKD should be performed". This is **hardly enough**, since we like **tangible** and **palpable** devices, we could *connect* our computers *onto* and start the *actual process*. This is the **essence** of what is called a **Bell State Analyzer**, and the picture below (*figure 3.7*) depicts the devices, that "out-of-the-box" achieves it.



FIGURE 3.7: A Device for QKD

(Source: <https://www.idquantique.com/quantum-safe-security/products/clavis-xg-qkd-system/>)

*PS: I'm sure this device is not just a **Bell State Analyzer**, but an array of other systems that are required for a secure Quantum Key Distribution.*

## 3.5 Heisenberg Uncertainty Principle

Besides using a *Bell State Analyzer* and a pair of **entangled Photons** in order to *securely communicate*, such a task **could** be achieved using plain **polarized photons** and making use of the *Heisenberg Uncertainty Principle* (*aka: HUP*). In 1927, the German physicist **Werner Heisenberg** made an **outstanding** discovery. This is greatly related to **wave-particle duality**, that explains the concept that particles (*quantum or not!*) behave like a *wave* in one moment and as a *particle* in the next. This principle,

combining with the *Non Cloning Theorem* that was stated above, has given us some interesting protocols that will be stated in a later section. (*For further reading on the HUP, please refer to appendix C.7 for further read.*)

## Chapter 4

# Building A Quantum Computer

In the previous chapter, we tried to demystify some of the *key* principles that come into play, in a **Quantum Computer**, as well as what it really takes to use a secure encryption scheme, like the ones we used with Traditional computing systems. In this one, we will try to describe the inner working of this marvelous system and some of it's building process.

### 4.1 Types of Quantum Computers

To everyone's understanding, the usage of *Quantum Mechanics* into a system, *did not* gave out a single implementation of a *Computer System*. It is only logical that a **single** design outgrow all the rest, but to begin with, there were some computational devices that tried to implement this technology in a *slightly different manner* and gave out some specialized computer systems that were or were *not* build.

#### 4.1.1 Quantum Turing Machine

A Quantum Turing Machine (*aka: Universal Quantum Computer*) is an **abstract machine**, that models the use of Quantum Computers. This model suggests that **any** quantum algorithm can be formally expressed by a QTM. Needless to say that, this machine shares common ground with the Turing Machine. At this point, it would be appropriate to point out that it is a purely *Mathematical* model. The explanation of the inner working of a QTM (or even a TM) is beyond the scope of this thesis. For the curious, I recommend giving [20]. Lance Fortnow is the *creator* of the QTM model with the publishing of his paper [28].

#### 4.1.2 Quantum Annealing

Quantum Annealing is more of an optimization process, rather than a full-blown system. In simple terms, it is a method of finding the maxima and minima of a mathematical function, with the use of quantum fluctuations. It is mainly used on problems where there is a discrete (or finite) search space. As an example is combinatorial optimization problems [27]. The process starts with a single Qbit in *superposition* of all possible *quantum-mechanical* states (*aka: candidates states*) with equal *weights*, and it evolves according to the time-dependent Schrödinger's equation. This is the **natural** evolution process for a physical system. As the time passes on, the amplitude of all candidates states changes according to the **time-dependent** strengths of the transverse field, effectively making quantum parallelism (D.1). This act simply causes a quantum tunneling **between states**, and effectively canceling some of the states out.

### 4.1.3 Adiabatic Quantum Computation

The Adiabatic Quantum Computation is a form of quantum computing that makes usage of the Adiabatic Theorem (see D.2). It is closely related to the one in the previous section. The only difference with *Quantum Annealing* is that, it requires that the changes in the *transverse field* are **slow** enough that the whole quantum system stays **really** close to the ground state of the instantaneous *Hamiltonian*. For further read, please consult [24].

### 4.1.4 Quantum Circuits

I saved the last place for Quantum Circuits, as it is the most common usage in Quantum Computers. It's analogous to classical circuits, used by traditional systems. This methodology is used to construct Quantum Logic Gates, similar to classic logic gates, with the sole implementation difference that they **must** be a reversible process. Richard Feynman used the notion of Quantum Circuitry in 1986 [26].

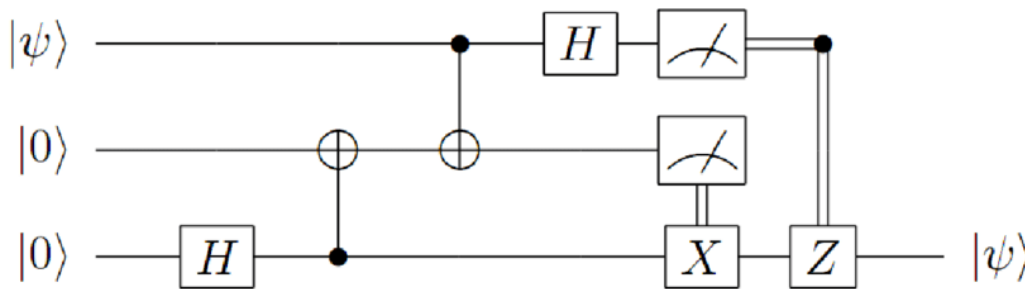


FIGURE 4.1: Quantum Circuit Picturing Quantum Teleportation  
 (Source: [https://www.researchgate.net/figure/Quantum-Circuit-for-Quantum-Teleportation\\_fig6\\_281376174](https://www.researchgate.net/figure/Quantum-Circuit-for-Quantum-Teleportation_fig6_281376174))

Note: *X* and *Y* are classically controlled Quantum gates.

Quantum Circuits follow the *Penrose Graphical Notation*. So, a quick refresh is found at Appendix D.3.

## 4.2 Quantum Logic Gates

As cited in the previous section, these are analogous to the well known logic gates, we all know. But the problem is that logic gates apply to binary bits that have a distinctive value of 0 or 1. Two bits in, one bit out. What will happen if instead of a bit, we try to pass through a Qbit in a superposition of four states (the equivalent of two bits into one). We cannot control its output, for the simple fact that we do **not** know out of the four states, the one that is responsible for the gate's output. Moreover, a Qbit in superposition is described as a vector with a complex number, with a *real* and an *imaginary* part. For that reason, the gate's output is a *matrix* and **not** a single number. (See Appendix D.4 for more).

Note: A number of Qbits in an enclosed space, is effectively a Quantum Register.

### 4.2.1 Controlled NOT Gate

Out of all the other Quantum gates, one gate is more usable than the others, and we encounter her **way** more. It is the *Controlled NOT Gate* (aka: *CNOT*). According to the results this gate gives, it resembles the *binary XOR* (Exclusive OR) gate, and it plays a **crucial** role to the Quantum Entanglement (and De-Entanglement):

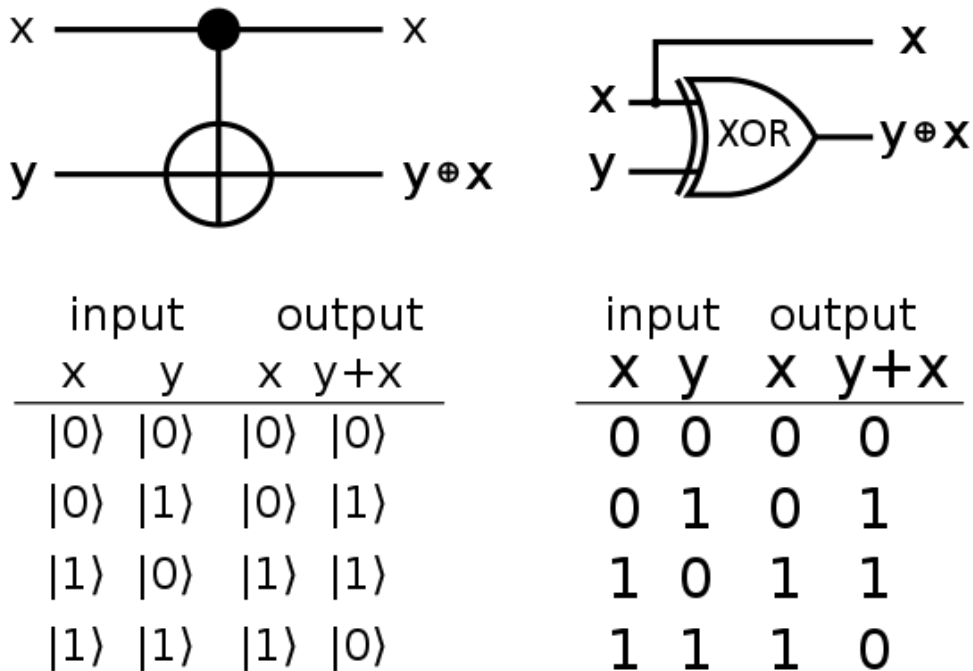


FIGURE 4.2: The CNOT Quantum Gate

(Source: [https://en.wikipedia.org/wiki/Controlled\\_NOT\\_gate](https://en.wikipedia.org/wiki/Controlled_NOT_gate))

Other name for the CNOT gate are:

- controlled X gate
- controlled bit-flip gate
- Feynman gate
- Pauli-X

### 4.3 Quantum Monte Carlo

The Quantum Monte Carlo is a large family of methods that are used in the creation of Quantum computer systems. All Monte Carlo methods share the same concept of handling multi-dimensional integrals, from a diversity of formulations in a many-body problem. All physical Quantum systems, as long as they are **not** moving at a speed that is close to the speed of light ( $c \approx 299,792,458$  m/s) can be effectively described by the many-body Schrödinger equation. But the problem lies that we

**cannot** solve the equation, because we do **not** know the many-body Wave function in a many-body Hilbert space. All Monte Carlo methods, allow us to study the complex (and complicated) many-body effects that are encoded into Wave function.

### 4.3.1 The Quantum Monte Carlo Methods

There are a few methods, that use Quantum Monte Carlo in a slightly different manner, in order to solve the many-body problem:

#### Zero-temperature (Only Ground State)

- Variational Monte Carlo
- Diffusion Monte Carlo
- Reptation Monte Carlo
- Gaussian Quantum Monte Carlo
- Path Integral Ground State

#### Finite-temperature (Thermodynamics)

- Auxiliary-field Monte Carlo
- Continuous-time Monte Carlo
- Reptation Monte Carlo
- Determinant Quantum Monte Carlo (or Hirsch-Fye Quantum Monte Carlo)
- Hybrid Quantum Monte Carlo
- Path Integral Monte Carlo
- Stochastic Green Function Algorithm
- World-line Quantum Monte Carlo

#### Real-time Dynamics (Closed Quantum Systems)

- Time-dependent Variational Monte Carlo

## 4.4 Quantum Requirements

### 4.4.1 The Importance of Cooling

Normally, when someone talks about a Quantum Computer, typically they refer to the *Quantum Processor*. But it is **hardly** all there is. For *Qbits* to be **able** to exist, ones **must** cool the system down to near **absolute zero**. Since the *Temperature* scale has to *tip* in the **negative** values so much, the *Science* community ditched the much older *Celsius* scale, for something more appropriate. Fixing the Boltzmann constant (denoted as  $k$ ) to be exactly  $1.380649 \times 10^{-23} \text{ J} \cdot \text{K}^{-1}$ . We concluded that one **Kelvin** results in a change of  $1.380649 \times 10^{-23} \text{ J}$  in the *thermal energy*. Since most humans, are wired to think of temperatures in *Celsius*, Scientists established a *compatibility* between these two scales. So,  $0\text{K}$  (aka: "Absolute Zero") is equal to  $-273.15^\circ\text{C}$ . Back on

tracks, let me point out some *typical values* for the *cooling requirements* for *Quantum computers* to function properly, and most importantly do the *computational tasks* that the Scientists and Engineers, want to execute.

In order not to *destroy*, the *Qbits* stored **inside** the registers, the machine is kept in a temperature of **four Kelvin** ( $-269.15^{\circ}\text{C}$ ). As this machine *radiates* its **cold** temperature, it renders the room completely **useless**. When the *Quantum Processor* is ready to run, the machine is **hot** for any use. This is where the pumps come in. They most likely pump in **Liquid Helium** and reduce the system's *Temperature* to approximately ten milliKelvin ( $-273.25^{\circ}\text{C}$ ). The temperature has to be monitored **very** closely. Or else, the *Qbits* will be destroyed. In the worst case scenario, the *Quantum Processor* will be destroyed.

#### 4.4.2 Eliminating "Noise"

Besides this issue of *cooling*. The computer **must** be enclosed in a (*big*) enclosure capable to **act as a shield**, and eliminate **all external Electromagnetic Radiation**. This is easier said, than done. The unwanted "noise" is **Electrical, Magnetic, and even Thermal**. In Figure 4.3 we can see a *Quantum Computer*. The wires are specially designed to transport *RF-frequency signals* without extra external noise to the processor, depicted at the bottom of the machine.

*Note: A Faraday Cage does the trick, but is not easy to implement such a enclosure.*

And Voila! (see Figure 4.3)

### 4.5 State of The Art Computers

In 2011, a company named D-Wave Systems made commercially available a Quantum Computer System, that works with the *quantum annealing* methodology and it was named **D-Wave One**. It had an 128 Qbits processor. [41] (See figure 4.4)

In May 2013, Google, NASA Ames and the USRA (Universities of Space Research Association) purchased an Adiabatic Quantum Computer System with a 512 Qbits processor.

In December 2015, Google has announced the D-Wave 2X. A computer that performed better than **any previous** system. It was said that it executed *Simulated Annealing* as well as *Monte Carlo* way faster. A speed up factor of 100.000.000 times. The test results were noted in using a set of hard optimization problems.

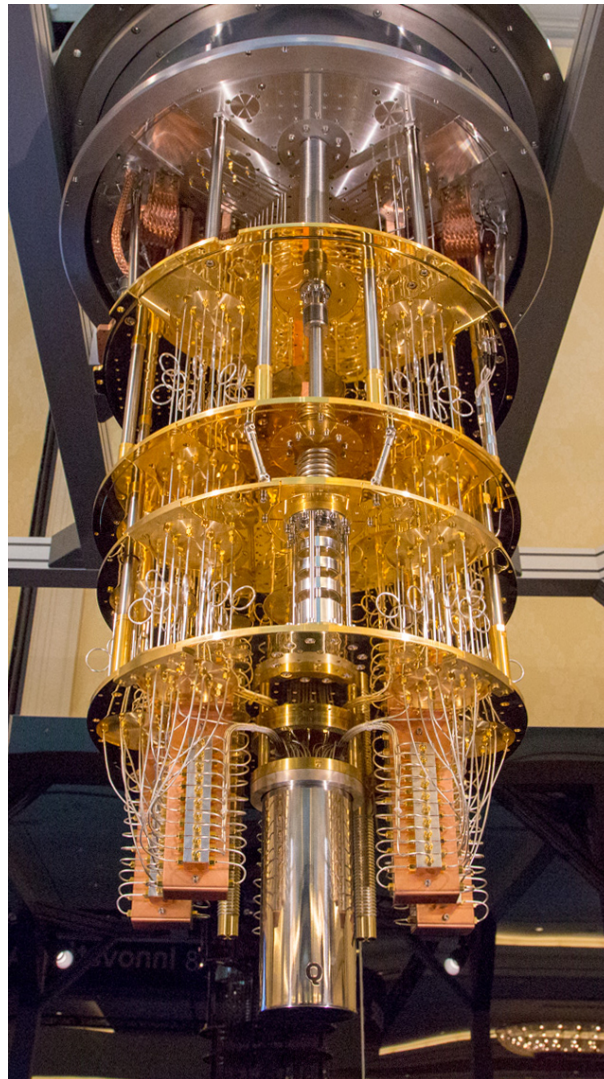


FIGURE 4.3: A Quantum Computer ( 50 Qbit)

(Source: <https://www.engadget.com/2018-01-09-this-is-what-a-50-qubit-quantum-computer-looks-like.html>)

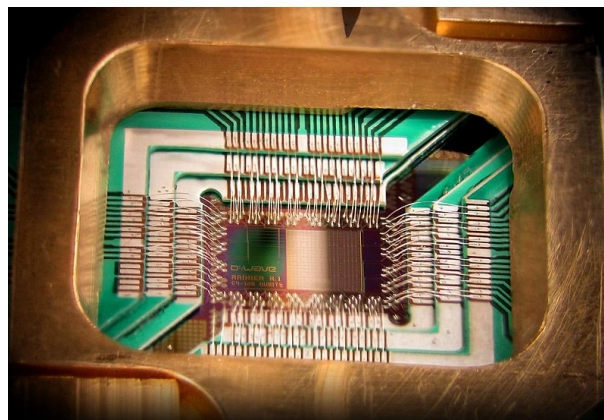


FIGURE 4.4: The D-Wave Processor (128Qbit)

(Source: [https://en.wikipedia.org/wiki/Quantum\\_annealing](https://en.wikipedia.org/wiki/Quantum_annealing))



## Chapter 5

# Securing Information in the Post-Quantum Era

Up to the point where **Traditional Computer Systems**, in conjunction with regular **Cryptography**, was used, the Information that circulated around what pretty *secure*. But *as soon as* Quantum Computers began to be used, the *cryptographic algorithms* were **unable** to provide adequate security, and by extension, keeping an information *private*, if it should be. This was a **major** problem, and it was the time to *upgrade* the cryptographic algorithms, so that they can provide the much needed security, despite the usage of lightning (...quite literally) fast computers. In this chapter, we will dig through the concept of **Cryptography** with the groundbreaking background of the Quantum computers.

### 5.1 Cryptographic Problems

Engineers and Scientist, started to analyze and sketch out what was going wrong with the *cryptology* as it is, and what can be improved upon. Finding out that **all** Cryptographic algorithms share three concept ideas in common, and improving those can make *hard enough* for Quantum Computers to break the *encryption scheme* and being able to provide privacy for private information. Cryptography, relies of three hard to compute, **Mathematical** problems. They are:

- The Integer Factorization problem
- The Discrete Logarithm problem
- The Elliptic-curve Discrete Logarithm problem

All of this problems could easily be solved while running **Shor's Algorithm** (see Appendix E.1) on a sufficiently powerful enough *Quantum Computer*. Thus breaking **Public-Key Cryptography Schemes**, such as:

- The **RSA** scheme
- The Finite-Field **Diffie-Hellman** Key Exchange
- The Elliptic-curve **Diffie-Hellman** Key Exchange

This was indeed a powerful motivator into researching and building new *Cryptosystems*, resistant even to Quantum Computers. This, took the name of **Post-Quantum Cryptography**.

In 2001, *Shor's algorithm* was used by a group in IBM to factor the number 15 into  $3 * 5$  using seven Q-bits. Eleven years later, in 2012, the successful factorization of 21 was achieved. And no one knows what the future holds... Large numbers have been factored by Quantum Computers, using different algorithms. On the bright side, these algorithms are quite similar to the classical *brute-force checking of factors*, and unlike *Shor's Algorithm*, they are **not** expected to perform better than classical algorithms that factor Integers [42].

### 5.1.1 Integer Factorization Problem

Integer Factorization or the difficult task of providing one, is one of the major building blocks for many cryptographic algorithms, such as RSA public-key encryption and RSA digital signatures. As seen in Figure 5.1, this concept is easier to illustrate rather than explain with words.

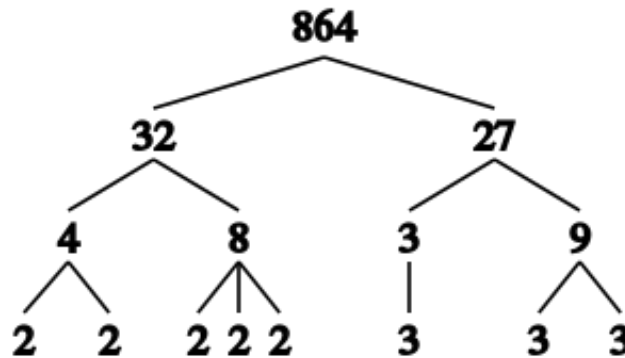


FIGURE 5.1: Prime Decomposition of  $n = 864$  as  $2^5 * 3^3$   
 (Source: [https://en.wikipedia.org/wiki/Integer\\_factorization](https://en.wikipedia.org/wiki/Integer_factorization))

Not all numbers of a given length are equally hard to factor. A hard problem is with Semiprimes (the product of two primes numbers).

### Fermat's Factorization Method

It's a factorization method, named after the French Mathematician *Pierre de Fermat*. It is based around of an odd integer, and thinking it about the difference of two square numbers.

$$N = a^2 - b^2 \equiv (a + b)(a - b)$$

### 5.1.2 Discrete Logarithm Problem

One of the three building blocks of a modern Cryptosystem. Discrete Logarithm are used in El Gamal-based systems.

Given any real number,  $a$  and  $b$ , we denote the logarithm  $\log_b a$  as  $x$ . This is equivalent to say, that  $b^x = a$ . Mathematical logic dictates that  $b$ , has to multiply itself  $x$  times in order to produce  $a$ . But, given  $a$  we do not know how many times, we have to multiply  $b$ . This is a difficult problem, that is bound to take many computing years to solve, unless we use a Quantum Computer.

### An Example of a Discrete Logarithm Problem

The power of 10 is:

$$\dots, 10^{-3} = 0.001, 10^{-2} = 0.01, 10^{-1} = 0.1, 10^0 = 1, 10^1 = 10, 10^2 = 100, \dots$$

Now,  $\log_{10}100 = 2$  but other numbers that what happens for other base-10 logarithms:  $\log_{10}53 = 1.724276\dots$  it means that  $10^{1.724276\dots}$  must be equal to 53. While some integer exponents can be defined in any group, with the usage of *products* and *inverses*, other arbitrary real exponents, like 1.724276, require other concepts to compute.

### 5.1.3 Elliptic-curve Discrete Logarithm problem

Elliptic-curve Discrete Logarithm problem (*aka: ECDLP*) is basically the same principle of a *Discrete Logarithm Problem*, but the only difference is, that instead of a static number, we make usage of the *Algebraic Structures of Elliptic curves* over a finite field (*aka: Galois Field*) (Seen in Figure 5.2). Elliptic curves can be used in Key Agreement, Digital Signatures and Pseudo-Random Number Generators. Neal Koblitz and Victor S. Miller, in 1985 presented a paper, that demonstrated their usage [44]. The theory behind them, is that elliptic curves are plane curves over a finite field. All points in the curve **must** satisfy:

$$y^2 = x^3 + ax + b$$

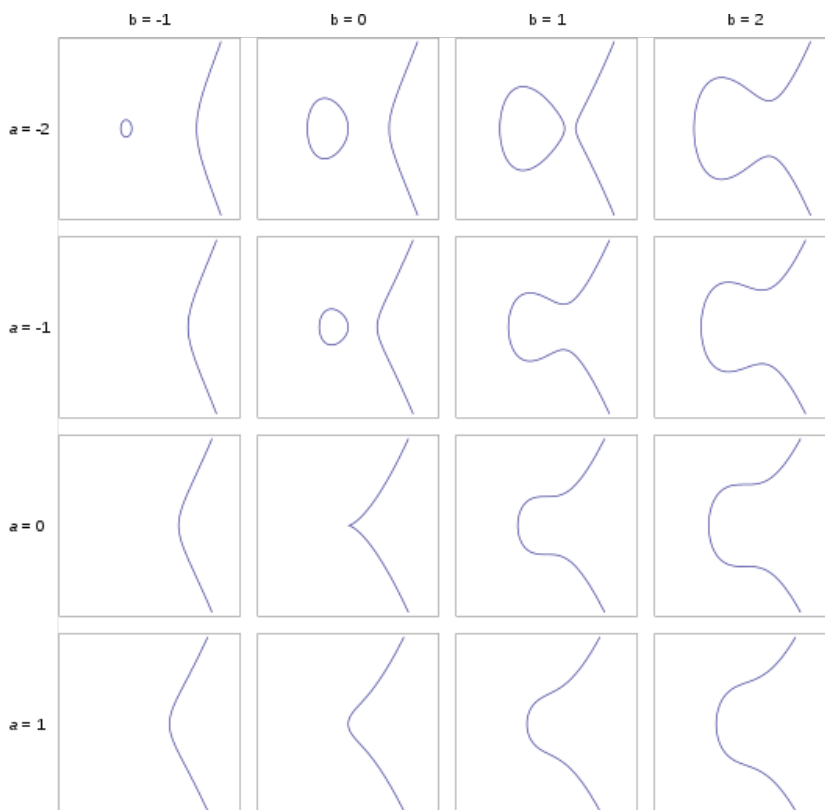


FIGURE 5.2: Visual Representation Of An Elliptic Curve

(Source: [https://en.wikipedia.org/wiki/Elliptic\\_curve](https://en.wikipedia.org/wiki/Elliptic_curve))

Note: The curve is defined that for  $x, y \in [-3, 3]$ . For  $(a, b) = (0, 0)$ , the function is not smooth, and this is not an elliptic curve.

Like all functions, an *Elliptic Curve* **must** be *non-singular*. Being *non-singular* is just the scientific term to just say that, the curve must not have any self-intersections or cusps (see Appendix E.2). In the previous figure, we demonstrated that a change in  $a$  and  $b$  coefficients, can result in a **massive** change in the final curve form.

There is a method to factorize an Elliptic-curve, called *Lenstra elliptic-curve factorization*. It is named after a Dutch Mathematician named Hendrik Lenstra.

### 5.1.4 Quantum Requirements

According to a many Scientists, a Quantum Computer running *Shor's Algorithm* can break the most advanced Cryptographic systems [4]. But what does it really take to render obsolete such schemes?

According to Scientists and Engineers, the last estimate for breaking a Elliptic Curve of 256-bit modulus, one of the most advanced Cryptographic techniques, is 2330 Qbits going through 126 billion Quantum gates (*aka: Toffoli gates*) [64]. For just a binary ECC, it will take in comparison 906 Qbits. Thees curves are both **128-bit security level**. In contrast to *elliptic curves*, Scientist can predict, that in order to successfully break the RSA algorithm, it would require 4098 Qbits and 5.2 trillion gates, if a key of 2048-bit is used. By deduction, ones can see that the *Quantum Computers* can be most successful at breaking the ECC rather than the RSA. In general, when increasing the *key length* in any algorithm, the slope of the *security strength* increases slower than its analogous key length. While performance cost increase much faster than the slope of the increased *key length* [39].

All is not lost, soon after the potential *fall* of the ECC [69], Scientist came up with a **Post-Quantum** alternate for ECC called *Supersingular Isogeny Diffie-Hellman Key Exchange* (*aka: SIDH*). It uses a secure form of ECC by using *Isogenies* to successfully exchange *Diffie - Hellman* keys. It was created a collaboration between two Scientist, named *David Jao* and **Luca De Feo**, in 2011 [40]. It was a start, but not a long lasting one, but it was quite a **spark**. Enough that, four years later, in August 2015, the NSA announced:

*"Unfortunately, the growth of elliptic curve use has bumped up against the fact of continued progress in the research on quantum computing, necessitating a re-evaluation of our cryptographic strategy."* [18]

A paper published by **Damien Robert** in 2022, shattered *SIDH*, and by extension ECC [63]. This is a good time to remind the reader of the **Bayesian Logic**. Science is a *self-correcting* process. With that said, an upgraded version of ECC, or even *SIDH*, may be *around the clock*.

## 5.2 Protocols based On Heisenberg Uncertainty Principle

In an earlier section we discussed the *Heisenberg Uncertainty Principle* in some details. As the convention of *all Information theory, Networking and Computer Sciences* lectures, states that the *two communicating parties* are named *Bob and Alice*. Here it is for the general QKD.

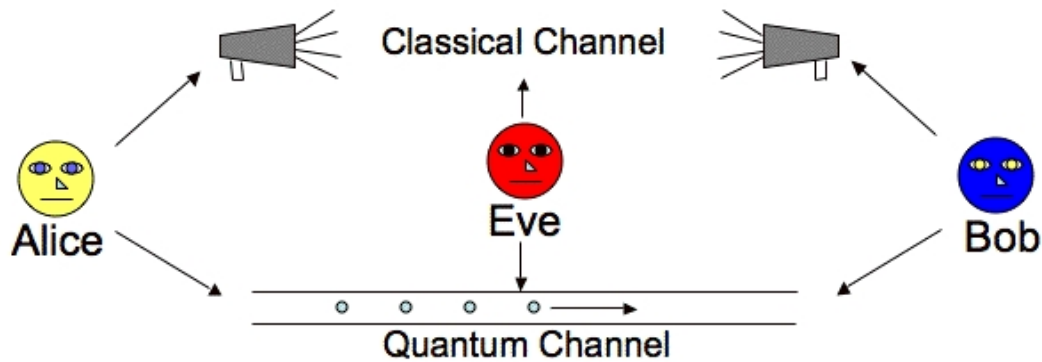


FIGURE 5.3: Visual Representation Of A QKD  
 (Source: <https://www.cse.wustl.edu/~jain/cse571-07/ftp/quantum/>)

### 5.2.1 The BB84 Protocol

One of the first *Quantum Protocols* for secure communications, that was ever created. It was from Charles Bennett and Gilles Brassard in 1984. It relies on **polarized photons** in order to achieve *secure communication* [6]. Before **any** communication can take place, *Binary digits* are **mapped** to values of polarization in photons, in **both** Basis: *Rectilinear* and *Diagonal*.

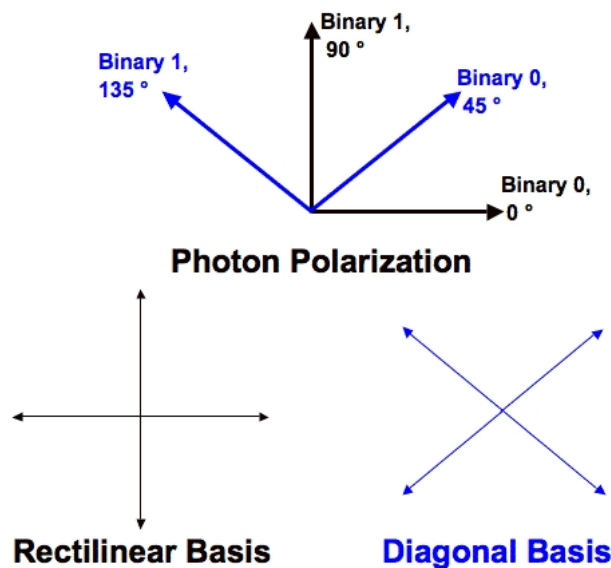


FIGURE 5.4: Visual Representation Of Photon-Bit Polarization  
 (Source: <https://www.cse.wustl.edu/~jain/cse571-07/ftp/quantum/>)

This protocol is divided into two distinct phases:

1. The **first** step in this *protocol* is for *Alice* to choose a *random string of Bits*. For each of the Bit chosen, she randomly chooses a Basis (Rectilinear or Diagonal) by which she'll **encode** the *Bit* and send it over to *Bob*. These actions are looped through **until Alice has sent all the Bits needed**. At the receiving end, *Bob* after **randomly** choosing a Basis, he measures the **photon's polarization**. If *Bob* have chosen the correct Basis, then **in theory** this specific *Bit* is **the same** that *Alice* have transmitted. If the Basis *Bob* chose for that particular *Bit* **doesn't** matches *Alice's*, then *Bob* would get a *random* value.

2. In the second phase, the error correcting process of this protocol. *Bob* will send to *Alice*, over an **insecure channel**, the Basis that he chose in the measurement of **each** photon. *Alice* will **only** notify him if the Basis was **wrong**. *Bob* will not try to **fix anything**. If the Bit is wrong, he'll discard it. Finally, before the communication is finished, **both** parties agree on the remaining **correct** Bits. This is known as a **Sifted Key**.

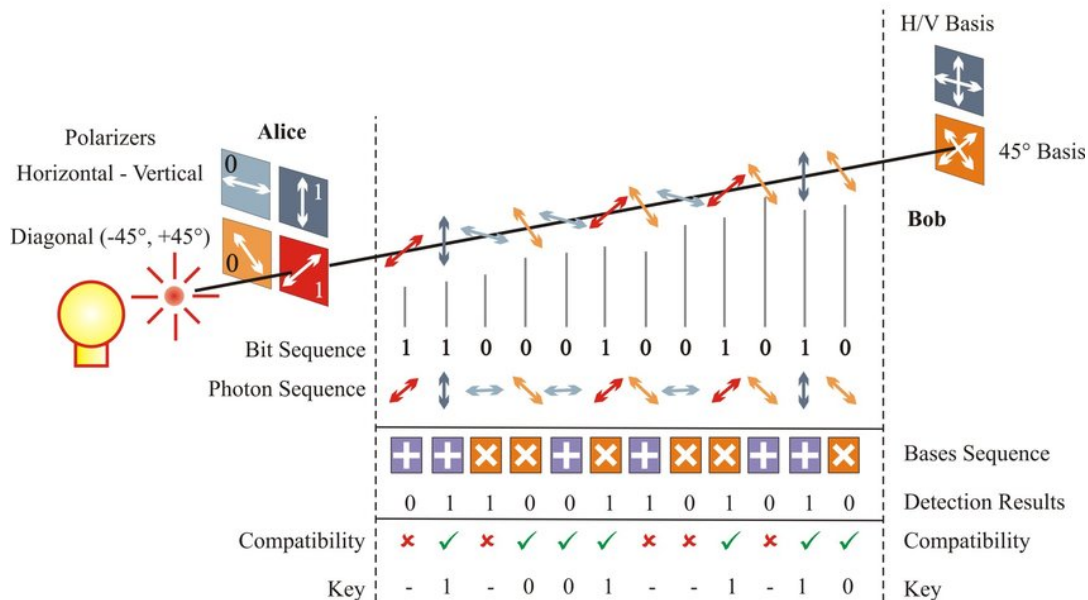


FIGURE 5.5: Visual Representation Of The BB84 Protocol

(Source: <https://www.researchgate.net/profile/Kamer-Vishi-2/publication/324115273/figure/>)

If an *eavesdropper* (in figure 5.3 depicted as *Eve*) is present and tries to intercept the Bits as they are being sent on the Quantum Channel. Effectively inserting herself **between Alice and Bob**. Let us focus on *Eve's* task. She **has to** measure each before sending the Bit to *Bob* [72]. That *necessity* is guaranteed by the fact that a particle of *unknown state cannot be replicated*, she has to guess a Basis and just get on with it. If her guess is **not** correct, then by the HUP it is demonstrated that the information stored in that particular photon, **would** effectively be **lost**. Think of it as a photon can effectively handle a *single* measurement.

### 5.2.2 The B92 Protocol

In 1992, Charles Bennett, that was the creator of the *BB84 Protocol* proposed a more simple version of the earlier protocol. In order not to repeat *all* the ways that the B92 protocol works the same way as the earlier BB84, I will just point out the **three** differences they have: [5]

1. Only **two** states are necessary in encoding, rather than **four** in the BB84 protocol (Depicted in figure 5.6).
2. In the first phase, *Alice* Basis selection is decided.
3. In the second phase, **Bob doesn't** measure nothing, if the Basis he **randomly chose** is *incorrect*.

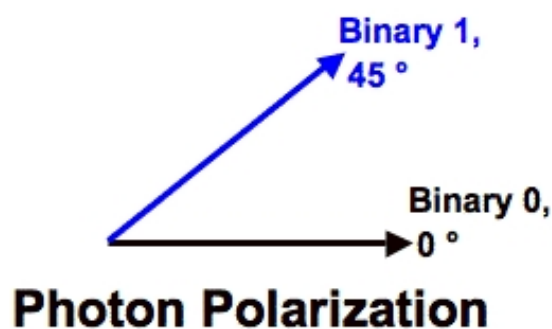


FIGURE 5.6: Visual Representation Of The BB84's 2-State Encoding

(Source: <https://www.cse.wustl.edu/~jain/cse571-07/ftp/quantum/>)

### 5.2.3 The Six-State Protocol

The Six-State Protocol (*aka:SSP*) was proposed by Pasquinucci and Gisin 1999. This **identical** to BB84. With its sole difference, that instead of using **four** states, it uses **six** states on **three orthogonal bases** [10]. In 2002, D.Bruss and C.Macchiavello, proposed an even **higher dimensional** QKD, but it **did not** took off [11].

### 5.2.4 The SARG04 Protocol

In 2004, Scarani, Acin, Ribordy, and Gisin proposed one of the last (*at the time of writing this Thesis*) protocol based on BB84. The *first phase* of this protocol, is **exactly** the same as BB84, but there are some differences in the **second** one.

In the second phase, where *Bob* is trying to match *Alice's* ones, she **doesn't announces** the basis she used, but *instead* sends over a pair of *Bit* encoded with the same state, as the one she **did** the first time. [31]

## 5.3 Protocols based On Quantum Entanglement

Besides HUP that gave birth to a number of QKD protocols, there is another type of protocols that are based on *Quantum Entanglement*. As a task, *Quantum Entanglement* is not an easy task. But nonetheless, it is a technique *widely used* in the field.

### 5.3.1 The E91 Protocol

The Eckert's protocol *aka: E91* was proposed in 1991 by Artur Ekert. [22]. It uses a source capable of generating **Entangled photons**. *Alice and Bob* both get a photon (as shown in figure 5.7). Then: Both *Bob and Alice* choose a **random** Base. They measure their *acquired photon*. Likewise in the protocol BB82, they discuss *on the classical "insecure" channel* their base on which they made the measurement. *If they both used the same base (for measurement) of the Bit*, then due to the principle of quantum entanglement, they **will** expect **opposites results**. In the end when all Bits have been sent, they'll end up with a *string of Bits* **Binary Complement** to each other. Finally, either *Alice or Bob* could just invert the values of the constituting Bits, and the **secret** Sifted key is shared among them.

In the presence of an **eavesdropper**, *Alice or Bob* can detect an intruder, by examining their *"discarded"* photons and measure one of them in a **third** measurement

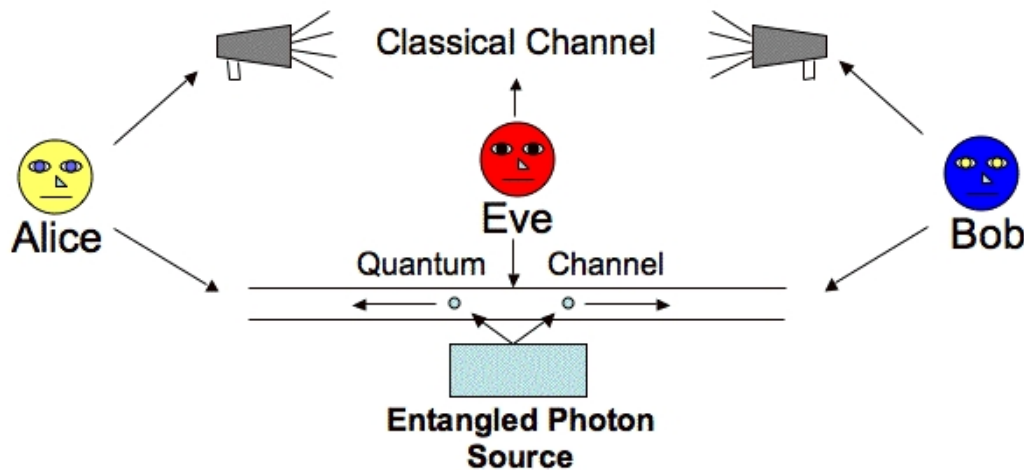


FIGURE 5.7: Visual Representation Of A QKD

(Source: <https://www.cse.wustl.edu/~jain/cse571-07/ftp/quantum/>)

base and **communicate their results**. With this newly acquired information, they can test **Bell's Inequalities**. Normally, they **do not hold true** for a *quantum particle*. If they **do**, then the sole explanation is for someone have intercepted the communication [32].

### 5.3.2 Entangled Versions of the BB84 Protocol

- In 2002, Enzer and his team of Scientists proposed an **Entangled** version of the SSP protocol [23].
- In 2005, Fung and his team of Scientists made an version of the SARG04 protocol that used **entangled** photons. Their aim for this new protocol was to increase the tolerance in errors. They made it possible for this protocol to work with a **two-photon source** rather than a **single-photon source** [29].

### 5.3.3 Cryptography without Bell's Theorem

There is a **idea** out, that scientists do not understand exactly the **Bell's theorem** and it's implications. The *Nobel prize* winners in the field of *Physics* for 2022 were three **Theoretical Physicists** that made made major contributions in the field of *Quantum Mechanics* and, in part to the **Bell's theorem and inequalities** (for a reminder on Bell's Theorem, please read Appendix E.4). So this is safe to assume that the cited before theorem and inequalities, are not easy to deal with. So, the logical conclusion is that in the eminent future, we may not see a rise of protocols that are based on **Entanglement**. On the contrary, HUP is much easier to deal with in an everyday life scenario [7]. **Quantum entanglement** is not the only way scientists have to make *Quantum-Safe communications protocols*.

## 5.4 Real World QKD Concerns

*In Theory*, QKD is **secure enough** for usage in the *Physical World*. The secure component of a QKD protocol, based on *HUP* or *Entanglement*, is not based whether or not an *eavesdropper* can compute hard **mathematical** problems, but rather in the inability



to violate **Physics**. The premise of a **man-in-the-middle** attack, is for the *eavesdropper* (in this example figure 5.3, Eve) to **pretend** to be *Bob* to *Alice*, and vice versa. Using a QKD protocol, an attack like this, is rendered **impossible** as *Alice and Bob* would **not** authenticate themselves to each other.

#### 5.4.1 QKD with Noisy Channels

In reality with physical systems, they are not **perfect**. If *one* of the parties discover their measurements not **perfectly** correlated, it is a very difficult task, **if not impossible**, to have conclusive proof of what caused their **imperfect** measurements. It may be something as *mundane* as a faulty instrument, some external "noise" in the channel, or an *eavesdropper* such as *Eve*. In the previous sections, we stated the way an *eavesdropper* was handled under **ideal conditions**. But the world we live in, is **not** ideal. Thus, *Alice and Bob* will be forced to discard **nearly every transmission that us not error-free**. This is a **major no-go**.

#### 5.4.2 QKD with the problem with Privacy Amplification

Let us start by explaining the premise of *Privacy Amplification* in context. Basically, it is the process that allows two parties of a communication process, to distill a **secret key** about which an *eavesdropper* has **partial** information about. In our example with *Eve*, we can logically draw the conclusion, that she successfully posses **some** Bits that constitute the **secret key**. [32].

### 5.5 The Research

The lack of **processing power**, bottlenecks currently (*at the time of writing this thesis*) *Quantum Computers* of becoming a threat to public-key algorithms. Even for common hash functions. For that matter, they are **theoretically considered as safe**. It is cited in numerous sources that a **doubling** of the *key size*, can effective enough to such an extent that even *Quantum Computers* will not be able to *break* the encryption. This is the reason the saying that, **Post-Quantum Cryptography does not** need to differ, by much from its *analogous* binary version [36].

Nonetheless, a large number of *Cryptographers* are designing new algorithms, and preparing the ground for when *Quantum Computers* pose indeed a **threat**. There approaches can be *summarized* into six fronts:

- Lattice-Based Cryptography
- Multivariate Cryptography
- Hash-Based Cryptography
- Code-Based Cryptography
- Supersingular Elliptic Curve Isogeny Cryptography
- Symmetric Key Quantum Resistance

### 5.5.1 Lattice-Based Cryptography

This term is much of a generic terminology, to indicate **all** the algorithms that base their cryptographic identity in forms of primitives using *lattices* (see Appendix E.3). Lattice-Based Cryptography was first introduced in 1996, by Miklós Ajtai. This approach include cryptographic systems such as:

- Learning With Errors
- Ring Learning With Errors (*aka: ring-LWE*)
- Ring Learning With Errors Key Exchange (*aka: RLWE-KEX*)
- Ring Learning With Errors Signature
- The open-source NTRU (or GGH) Encryption Schemes
- The new NTRU Signatures
- The BLISS Signatures

The Post Quantum Cryptography Study Group, sponsored by the European Commission, made the suggestion that the *standardization* of the NTRU algorithm shall be Stehle - Steinfeld variant of NTRU, and **not** NTRU algorithm by itself.

### 5.5.2 Multivariate Cryptography

This is a generalized term for Asymmetric cryptographic primitives, that are based on multivariate polynomials. The solution is defined over a finite field. Most used case scenario is that the polynomials are of second degree. If that is the case, then we talk about multivariate *quadratics*. Solving such a system is *NP-complete*. This kind of *encryption scheme* was presented by Tsutomu Matsumoto and Hideki Imai in 1988. The applications of such cryptographic systems are:

- Unbalanced Oil and Vinegar
- Hidden Field Equation
- SFLASH
- Rainbow
- QUARTZ
- TTS
- QUAD (Cipher)

Besides these seven algorithms, there are four cryptography **signature** schemes, that stood out in the *2nd round of the NIST post-quantum competition* [50]. They are:

- GeMMS
- LUOV
- Rainbow
- MQDSS

### 5.5.3 Hash-Based Cryptography

Hash-Based Cryptography, make usage of the *Merkle Hash Tree* as its signature scheme. Proposed in 2005, by Luis Garcia. He demonstrated that the one-way hash function are still secure against Quantum computers. In 2022, NIST announced that SPHINCS+, a hash-based algorithm, as one of the three algorithms that will be standardized for **digital signatures**. Other known algorithms that are *quantum-safe* are:

- Extended Merkle Signature Scheme (*aka: XMSS*)
- Leighton - Micali Signatures (*aka: LMS*)

**XMSS** as well as the original **SPHINCS** algorithms were introduced in 2011 and 2015 respectively. Back in the beginning, Leslie Lamport invented the hash-based signatures, all together back in 1979. The *Post Quantum Cryptography Study Group* sponsored by the European Commission, has recommended the usage of **Merkle Signature Schemes** as secure in *Post-Quantum* [19].

### 5.5.4 Code-Based Cryptography

The Code-Based Cryptographic systems use *error-correcting codes* for securing the error-free system. It includes:

- McEliece Encryption Algorithm
- Niederreiter Encryption Algorithm

The *Post Quantum Cryptography Study Group* sponsored by the European Commission, has recommended the usage of **The McEliece Encryption Algorithm** as a candidate for secure *public key encryption scheme* for *Post-Quantum* [19]. The Niederreiter Encryption Algorithm, even if it was based on the McEliece Cryptosystem, was proven insecure in the *Post-Quantum era*.

#### McEliece Encryption Scheme

McEliece Encryption Scheme (*aka: McEliece Cryptosystem*) is an *Asymmetric Algorithm*, that used **randomization** in the encryption process. It was first developed in 1978, by Robert McEliece. It uses **Goppa codes** and it is *NP-hard* for its cryptanalysis.

#### Niederreiter Encryption Algorithm

Niederreiter Encryption Algorithm (*aka: Niederreiter Cryptosystem*) is closely based, and a variation of McEliece Cryptosystem. The idea is the same, a *parity check matrix*. But despite, the one it is based upon, it provides an increase in encryption speed ten times faster. It can be used to create *digital signatures*. It was developed by Harald Niederreiter in 1986.

### 5.5.5 Supersingular Elliptic Curve Isogeny Cryptography

This *Cryptographic System* was intended as a replacement of Diffie-Hellman. It added *forward secrecy* as a bonus. It used the properties of *supersingular elliptic curves*, combined with the *supersingular isogeny graphs*. It used to work like Diffie-Hellman, and

provided a complete compatibility. In 2012, researchers Sun, Tian and Wang from the Chinese State Key Lab for Integrated Service Networks and Xidian University, extended the work of De Feo, Jao, and Plut, and created a Quantum-safe digital signatures system based on supersingular elliptic curve isogenies. But according to the fall of ECC, there is no patents covering this cryptographic system.

### 5.5.6 Symmetric Key Quantum Resistance

It is *common knowledge* that if provided with a large enough key for *Symmetric Key Cryptography*, the algorithms like AES and SNOW 3G, are pretty much safe against *Quantum Computers*. By extension, systems like Kerberos [16] (or even 3GPP Mobile Network Authentication Structure) are considered secure and hard to break.

## 5.6 Comparison of Algorithms in Post Quantum

The table below shows some values, for different schemes at a 128-bit **Post Quantum security level**, see table 5.2.

TABLE 5.1: Post Quantum Algorithms Comparison

Algorithm	Type	Public Key	Private Key	Signature
3072-bit Discrete Log	NOT SAFE	384 B	32 B	96 B
256-bit Elliptic Curve	NOT SAFE	32 B	32 B	65 B

**The above table is only for comparison between algorithms!**

### 5.6.1 Security Reductions

In Cryptography, *security reductions* is the term given to the proof of the amount of **hardness** an algorithm has. We do hope that all algorithms used in cryptography, are hard mathematical problems, but we can't know for sure, unless tested before. In *Quantum Computing*, as in regular one, Researchers are always looking for loopholes in a cryptographic algorithm, that may fully or partially compromise the security.

### 5.6.2 Lattice-based Cryptography - Ring-LWE Signature

*Ring-LWE* algorithms tend to be on the lower side as a matter of security. The problem is that **Lattices** are not so secure, and careful is required. Some Lattices however are secure for even the most advanced computer out there. This is the reason behind the flagging of some algorithms that do use lattices, are not recommended for secure usage. Scientist agree that the *Shortest Vector Problem (aka: SVP)* is an *NP-hard*. Systems like the *Lyubashevsky's* variant of Ring-LWE have been demonstrated to have a *security reduction*.

TABLE 5.2: **Post Quantum** Algorithms Comparison (*continued*)

Algorithm	Type	Public Key	Private Key	Signature
NTRU Encrypt	Lattice	766.25 B	842.875 B	
Streamlined NTRU Prime	Lattice	154 B		
Rainbow	Multivariate	124 KB	95 KB	
SPHINCS	Hash Signature	1 KB	1 KB	41 KB
SPHINCS+	Hash Signature	32 B	64 B	8 KB
BLISS-II	Lattice	7 KB	2 KB	5 KB
GLP-Variant GLYPH Signature	Ring-LWE	2 KB	0.4 KB	1.8 KB
NewHope	Ring-LWE	2 KB	2 KB	
Goppa-Based McEliece	Code-based	1 MB	11.5 Kb	
Random Linear Code Encryption	RLCE	115 KB	3 KB	
Quasi-cyclic MDPC McEliece	Code-based	1232 B	2464 B	
SIDH	Isogeny	564 B	48 B	
SIDH (Compressed Keys)	Isogeny	330 B	48 B	

### 5.6.3 Lattice-based Cryptography - NTRU, BLISS

Between NTRU and BLISS algorithms, there is not so much difference, hence the title, of putting those two *together*. Both of them rely on the Closest Vector Problem (*aka: CVP*) for their *cryptographic needs*. The CVP is known to be an NP-Hard problem, so NTRU and BLISS are too. We said this in an earlier section, like an proverb use to say: "*...the student is better than the teacher...*", hence *Stehle-Steinfeld variant* of NTRU, shall be used **instead** of the original NTRU.

### 5.6.4 Multivariate Cryptography - Unbalanced Oil and Vinegar

Unbalanced Oil and Vinegar is an Asymmetric algorithm. Its security relies on *Multivariate cryptographic primitives*. If using second degree polynomials, then by extension, the **Multivariate Quadric Equation Solving Problem** is NP-Hard, as demonstrated by Bulygin, Petzoldt and Buchmann [12].

### 5.6.5 Hash-based Cryptography - Merkle Signature Scheme

In 2005, a Scientist named Luis Garcia demonstrated that there **was a security reduction** in the *Merkle Hash Tree*. It was then linked to the underlying *hash function*. Since then, they fixed the issue, and from that time onward, *Merkle Hash Tree* is proven secure [57].

### 5.6.6 Code-Based Cryptography - McEliece

The McEliece Encryption System (*aka: McEliece Cryptosystem*) is a security reduction of the **Syndrome Decoding Problem** (*aka: SDP*). It is known to be an *NP-hard*. The *Post Quantum Cryptography Study Group* sponsored by the European Commission, has recommended the usage of this *cryptography* as being a safe bet against *Quantum Computers* [19].

### 5.6.7 Code-Based Cryptography - Random Linear Code Encryption

Random Linear Code Encryption (*aka: RLCE*) was first proposed by Yongge Wang, in 2016. It is derived by the *McEliece* scheme. But despite the latter, it can be constructed using any linear code (example: *Reed-Solomon* code). It inserts *random* columns in the linear code **generator** matrix.

### 5.6.8 Supersingular Elliptic Curve Isogeny Cryptography

The security of Supersingular Elliptic Curve Isogeny Cryptography relies in constructing an isogeny **between** *two* supersingular elliptic curves, with maintaining the **same** number of points as the two curves. This is known to be a NP-Hard problem, that require many years worth of processing power.

## 5.7 Forward Secrecy

The term of *FS* (*aka: PFS*) is a fundamental feature in any modern *key agreement protocol*. It is based on the reassuring idea that, a *session key* will **not** be compromised, even after an attack or the information is compromised. This is done by generating an *unique session key*, for every time a transfer is taken place. It is noted, that all the **other** circulating data from other sessions in the same time, will not be compromised, because of a faulty session. It is done mostly in the *Transport Layer of OSI model*.

Both, the Ring-LWE Key Exchange and *SIDH* Key Exchange can support *FS*, But they can also be used without *FS*, creating a classic *ElGamal encryption* of a Diffie-Hellman variant. While this feature is a good thing to have, some current algorithms, such as NTRU, for example, do not support *FS*.

(For a remind of The OSI model, please see figure 5.8)

## 5.8 Open Quantum Safe Project

In 2016, a project started, in order to develop and prototype *Quantum-Resistant Cryptography*. Its ultima ain is to construct a *single open source C* library, that will incorporate **Post Quantum Algorithms**. Obviously, most weight is put onto the *key exchange algorithms*, but it's not **limited** by them [53].

At the time of writing this thesis, the key exchange algorithms that are supported are:

Besides the **software** side, *Microsoft Research* is trying to implement the **PICNIC** (a PKI algorithm) using *Hardware Security Modules*. Besides *Microsoft* arriving late, *Google's NewHope algorithm* have all ready been implemented by **HSM** vendors.

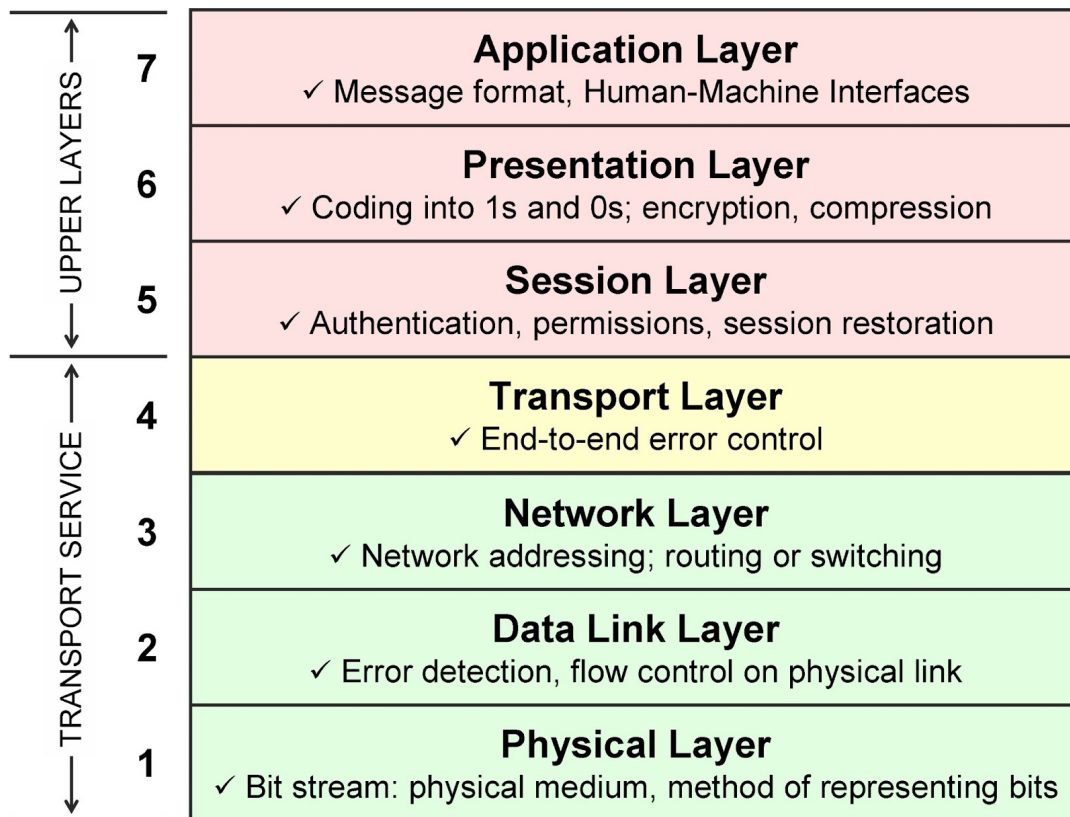


FIGURE 5.8: Visual Representation Of The OSI Model (Reference)

(Source: <https://huntbycode.blogspot.com/2017/01/osi-network-model.html>)

TABLE 5.3: Open Quantum Safe Algorithms

Algorithm	Type
BCNS15	Ring-LWE Key Exchange
NewHope	Ring-LWE Key Exchange
Frodo	LWE algorithm
NTRU	Lattice-based Cryptography
SIDH	Supersingular Isogeny Key Exchange
McBits	Error-Correcting Codes

## Chapter 6

# Conclusions

### 6.1 Conclusion

In this *thesis*, I tried my best to demystify some of the fuzz that goes around **Cryptography** and **Quantum Computers**. In my research, I came across articles, media as well as scientific (*pseudo-scientific, that is*), that made a case that *Cryptography* is dead, and **all secure transactions** will eventually cease, as soon as **Quantum Computers** will take over. As you can see, this is certainly **not** the case. Even thou, I am just a Computer Scientist, I tried to scratch the surface of the secure transactions. Giving out the bibliographic references, as a reminder of what I said, and the reason I did said what I did (*wrote, to be exact*). Someone more specialized than what I currently am, is free to further read the articles and books I cited and come to a conclusion on his/her own. But I don't **really** think, that this conclusive decision another Scientist *will* draw, would be much different than I did. Back at the topic at hand, the *Iceberg Effect* suggest that, there are more algorithms and systems, and I have chosen to address a subset of them. By no means, that thesis tells **everything** there is to *Cryptography* or *Quantum Computers* for that matter. It is simply **not a feasible task**. There is still ongoing research to our understanding of *Quantum Mechanics* as well as *Algorithms*, both **Quantum or not**.

Throughout the research I did, for writing this *Thesis*, I drawn a *good enough* idea of where the topic are going and *where we as Scientists* will more likely see a substantial upgrade. Quantum Computers, as for the time of writing this thesis, are gargantuan computing systems, the Quantum processor **alone** is *almost* palm-size. In comparison, a CPU (*wsf: Central Processing Unit*) is approximately a square of *4cm* in size, and *5mm* tall. The Inter Core i5-6600K is used in this example, but this provides a nice confidence order of magnitude for the physical dimensions of **any CPU**. With that said, we won't see random peoples replacing their household computers with Quantum ones, any time soon. Besides the obvious size issue, there is a **much bigger problem**, that is their **cooling**. Putting aside all these problems, when they **do** work, they speed the process of doing computational tasks, so *much*. An example is the simulation of a molecule (Beryllium Hydride,  $BeH_2$ ) using a seven Qbit processor, in a few minutes. Whereas for a traditional system this simplified task would be practically impossible, or in the best scenario, it would take years (or even *decades*) of computing power. Let us not root for those systems right away, because of their peculiar nature.

This is the *good part* about Quantum computing. But rarely, if not, **ever**, a topic is **one sided**. Quantum computing poses a series of issues on the table. Besides the "**Absolute Zero**" temperatures that are *tricky to achieve in a non-lab environment* (to say the least), and the fact that this machine **has to be shielded** from "Noise" (*External Electromagnetic Interferences*), the resulting system, is **only** compatible with a handful



of algorithms. This number indeed is getting bigger, but it is a (*small*) subset of the analogous number of algorithms that are constructed for *traditional binary* computers. Quantum computers are well suited for *Chemistry* or *Material Simulation*. But **are not** for accomplishing something more meaningful for the everyday consumer in a *typical* everyday life. As *quantum computers* are really **difficult** to manufacture, this is the reason behind their hefty price tag. I really think that *finances* and *science* are two different kind that **should never mix**. But this is the reason behind, the handful of organizations that **do** possess one. Let us put the economical aspect aside. Doing *Word Processing* or giving out a *PowerPoint* presentation using a quantum computer is a **very** long way off.

The bottom-line is that we do **not** know as much as **we need to know**. Solely that statement should be enough motivation, and it most certainly is, for Scientists (Theoretical or not) to keep going and push the boundaries of our understanding. Algorithms, as well as theories, are not carved in stone. (Remember the *Bayesian Reasoning* I said in the Thesis's Acknowledgments?) We are pretty much **spot-on** when it comes to the *Macrocosmos*, but we are still researching and correcting our knowledge of the *microcosmos*. Don't get me wrong, *Quantum Computers* are a pretty massive technological achievement. The idea is a little twisted, because unlike *binary logic*, we are constructing computer systems, from something that we **do not** fully understand. There is ongoing research going on, in the fields of *Quantum Mechanics* that will carry Quantum computers along. Our understanding, firstly in *Physics*, and then in **Informatics Engineering** is growing. But on specific sections (like **Bell's Theorem**) *Scientists* are stuck without conclusive proof towards proving or disproving sections of the theorem, and for a long time. (Too long if you ask me).

## 6.2 Further Research

*Quantum Computers* and *Cryptography* is an ongoing field, with **much** research put into it. And where some algorithms are **rendered** obsolete, like the example of *ECC* that was considered as *not safe*, others are created. Thees computer systems work with such a speed that is possible to process all the variables necessary for developing **true** Artificial Intelligence (*aka: AI*). Based on the **brains** of an award winning *Touring test bot*, AI is a series of nested *if* and *switch* statements. Is this **really** AI? (*Rhetorical question.*) The field of *AI* as well as *Machine Learning* (*aka: ML*), is a fascinating topic. Surely the speed of a Quantum PC, will come in handy to AI and ML usages. **If** we manage to correctly program those computers to getting the thing we want. Because programming for a *Quantum Computer* is not an easy task, and many errors are done in the process. To my current knowledge, there are **only two** programming languages used for programming on a *quantum computer*. Obviously **C** and **Python**. Both of those use a *library add-on* that is imported in the header portion of the program.

All this is *computer-related* subjects. From an **Engineering** perspective, **not all** topics revolve around *computers*. Understanding **any** piece of *knowledge* can **potentially result** on "better" Computing systems. For example: *Quantum Mechanics* to quantum computers, *Electronic Engineering* to regular computers, and so on... To generalize this idea, **Physics** and **Mathematics** give us **new** boundaries in order to make any technological achievement. But ultimately, it is up to us to decide and make the best use of this breakthrough information. We (*Humans*) may use that

piece of knowledge into a newly designed machine, a hardware of some kind, or just to make our life easier.

Since so many bright minds have demonstrated (Theoretically at least), that there can be speeds **faster than the light speed**. So, there is a possibility of something resembling to a computer system, that is faster and better than a Quantum computer. Who knows what the Future holds for us?

Like many Scientists before me, I agree with them *that the speed of light is a cosmic speed limit*, for anything that travels **through** space, making the possibility for faster than light travels. This does raise some eyebrows, because if we approach the *speed of light*, our mass tends to be **infinite**. For now it is solely **Theoretical**, but who knows what the future will unravel. Maybe, Quantum computers is just a stepping stone for something greater (*and faster*)... With our current grasp of Quantum Mechanics, we know that all the things in *Macro*, obey the **Newtonian Mechanics**, but when we start to talk about the *Micro*, they do not behave obeying the law of **Newtonian Mechanics**, but instead they behave differently. Those two "*rules*" cited above, explain the reason *Quantum Mechanics* is **not** understood. Maybe someday a Scientist will come up with a more plausible (and complete) explanation. Answering more "*questions*" we currently have unanswered. Spinning our current theories on their heads. But until then, we have some theories and explanation of *why things happen when they do*.

(...) In some cases, objects or waves may appear to travel faster than light (e.g., phase velocities of waves, the appearance of certain high-speed astronomical objects, and particular quantum effects). The expansion of the universe is understood to exceed the speed of light beyond a certain boundary. (...)

Wikipedia

I agree with the reader: "Wikipedia is not **really** a reputable source." and that statement is absolutely right. But since I have not researched the topic of *FTL* in depth, I really think that a Wikipedia article, as *unofficial as it may be*, it is a great starting point to start investigating deeper. Finally founding information (*and papers*) to support an **official** hypothesis. After all, *there is no smoke without a fire nearby*.

## Appendix A

# Clarifications For Chapter 1: Securing Information in the Pre-Quantum Era

### A.1 Pigeonhole Principle

The *mathematical Pigeonhole Principle* states that if  $n$  items, have to be put in  $m$  containers, with  $n > m$ , then at **least one** container **must** contain more than one item. Since humans live in a three-dimensional world, this premise is more than logical.

### A.2 The X.509 Standards

The X.509 came shortly after X.500 in 1988. X.509 standardized the format of *Public Key Certificates*. In other words digital documents that securely associate Cryptographic Key Pairs with the identities of peoples, organizations and websites. The X.500 was the standard for electronic directory services, while the X.509 build upon the X.500 by expanding it for **internet** use. The RFC 5280 profile incorporates the X.509 version 3 certificates, the X.509 v2 Certificate Revocation List (*aka: CPL*), and the description of an algorithm for Certificate Path Validation.

(Note: The SSL/TLS and HTTPS use the X.509 Certificates.)

## Appendix B

# Clarifications For Chapter 2: Honorable Mention: Kerberos

### B.1 Needham-Schroeder Protocol

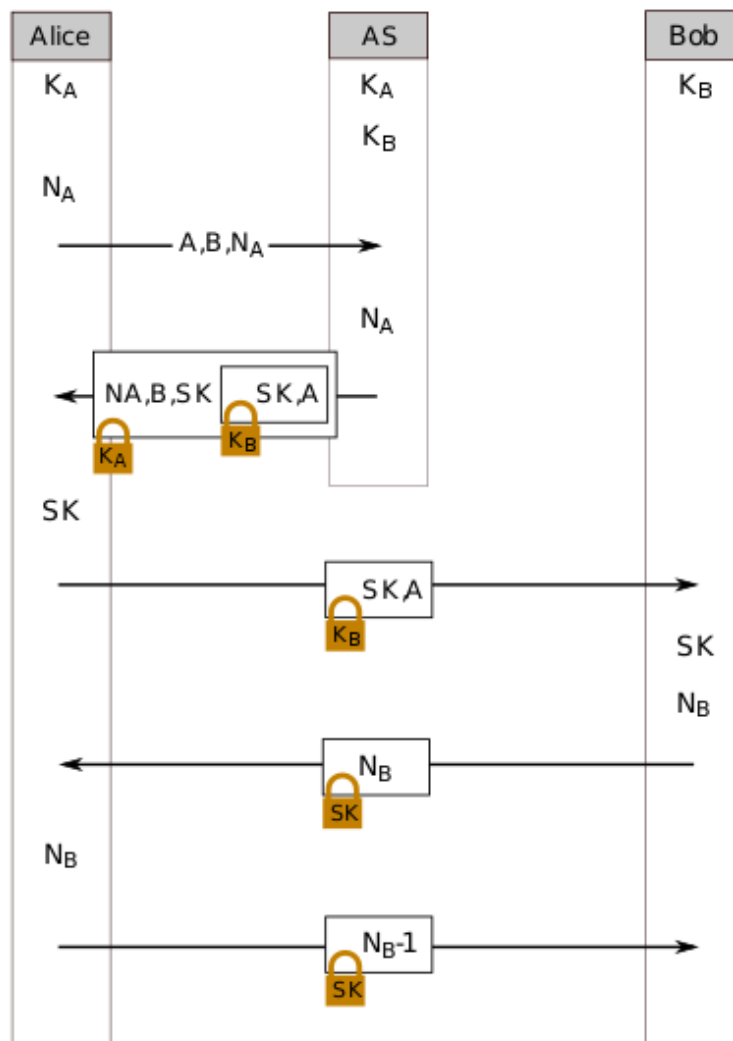


FIGURE B.1: The Symmetric Needham-Schroeder Protocol  
(Source: <https://handwiki.org/wiki>)

This is one of the oldest *Key Transport Protocol* that exists, for transporting a secure key over an insecure network. It dates back from 1978 and was conceived by Roger Needham and Michael Schroeder [51]. It really incorporates two *Protocols* (or ideas if you wish):

- The *Needham-Schroeder Symmetric Key Protocol*: As it's name implies, this is a symmetric key protocol, meaning that it uses the same key for encryption as for decryption. This was the building blocks for the early versions of *Kerberos*. The goal of this *protocol* is to establish a **session key** between two communicating parties, over an insecure network.
- The *Needham-Schroeder Public Key Protocol*: This is a version of the latter protocol, using *Asymmetric Key Cryptography*, to communicate securely. It aimed to provide a *mutual* authentication between the two parties. Other than for *Academic purposes*, this version is forgotten, because it was not very secure in the end.

As seen in figure B.1, we have many notations but it is not very complicated when one's understand the gist of the all the endeavor... so, let us slowly chunk away:

- *A and B* are just the identities representing *Alice and Bob* respectively.
- $K_A$  and  $K_B$  are the **Keys** for entity A and B.
- Now, it is time to introduce to the communication the *Authentication Server*, denoted here as AS.
- $K_{AS}$  and  $K_{BS}$  are the symmetric keys for Alice to the Authentication server, and by analogous, Bob to the Authentication server as well.
- $K_{AB}$  is a symmetrical **Session** key, generated for **Alice and Bob**.
- $N_A$  and  $N_B$ , are what we call **nonces** in cryptography. Think of it as a *Random Number Generator's Padding*.

## B.2 Project Athena

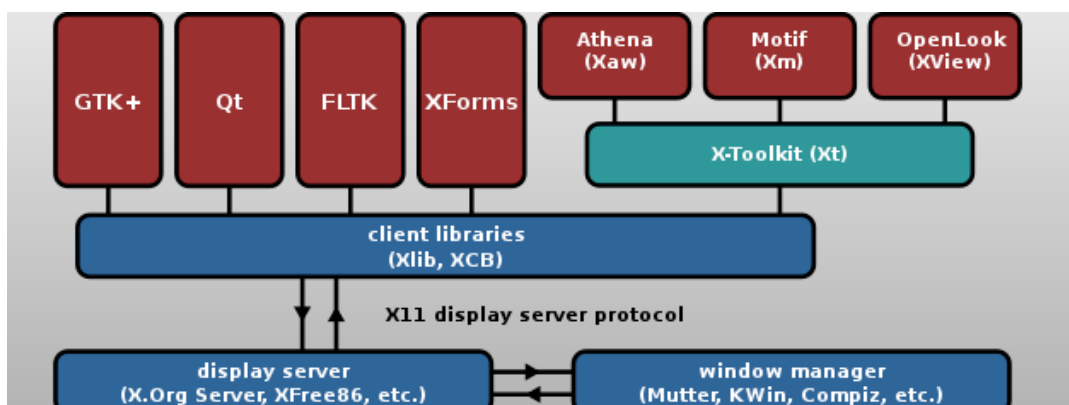


FIGURE B.2: The Project Athena

(Source: <https://engineering-high-tech.com/article/what-is-athena-project-at-mit>)

As a writer for this thesis, I feel that discussing *Project Athena* is on the borderline for a security-oriented thesis. But because *Project Athena* was the birthplace for a number of technologies, such as the LDAP, Active Directory, Zephyr Notification Services, the X Window System, Instant Messaging and the best of all: The Kerberos Protocol, I came to the decision to include a small section in my thesis. This project was a collaboration between MIT, Digital Equipment Corporation and IBM. Its goal was to produce a campus-wide **Distributed Computing Environment** that would be used for educational needs. This project dates back from 1983, and as of today, it is still in use. Although *R&D* came to a halt in *June 1991*. In 2020, it was incorporated with the *Debian package manager* for many Linux (*and UNIX*) packages.

## Appendix C

# Clarifications: Chapter 3: The Quantum Era

### C.1 What quantify as Quantum Particle or not?

Light, hence Electromagnetic Radiation, and at it's core, **the Photon** is not the only Quantum particle there is. According to the work of Max Bourn, the cyted above particles have in common that they can be described by an "*Wave Function*". With that said, except photons, other particles as electrons, neutrons and the newly discovered Higgs Boson, are **Quantum** as well.

### C.2 What is a "Wave Function"?

A "Wave Function" is a Mathematical system describing the states of a Quanta situated in a Quantum system. Using that function we can calculate the probabilities of possible results on the system. The most important wave function is the Schrödinger equation (see figure C.1). Commonly, the Greek letter of  $\psi$  (or  $\Psi$ ) is used to describe a that function.

The picture is a little intimidating, but if ones know the symbols, it become much simpler. So, please let me demystify the equation quite a bit.

$$i\hbar \frac{d}{dt} |\Psi(t)\rangle = \hat{H} |\Psi(t)\rangle$$

FIGURE C.1: Schrödinger's Equation (Time-Dependent)  
(Source: [https://en.wikipedia.org/wiki/Schr%C3%B6dinger\\_equation](https://en.wikipedia.org/wiki/Schr%C3%B6dinger_equation))

$i$ : Just an imaginary unit.

$\hbar$ : The reduced **Plank Constant**.

$\frac{d}{dt}$ : Derivative over Time.

$|\Psi(t)\rangle$ : The Wave function for that specific Time.

$\hat{H}$ : The *Hamiltonian* operator.

### C.3 What is *really* Superposition in Quantum Mechanics?

The simple act of **adding** quantum states (two or more) to a particle and the result will be a valid *new* quantum state. This is possible feasible because of the fact that

superposition is a complex (imaginary) number. As seen on picture C.2, it has two parts in it. A *real* part, and an *imaginary* one (here denoted as *Re* and *Im* respectively). With the use of these two, one's can express **any** possible number there is. Or, here in the example of quantum mechanics, **any** state.

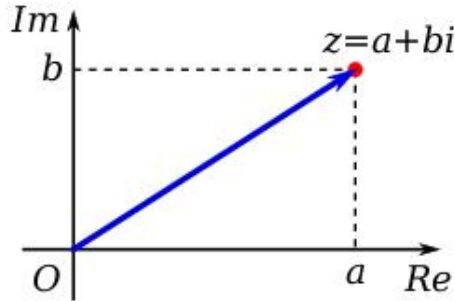


FIGURE C.2: Graphical representation of a Complex Number  
 (Source: [https://en.wikipedia.org/wiki/Complex\\_number](https://en.wikipedia.org/wiki/Complex_number))

### C.4 What happens to the original object after it has undergone teleportation?

This is a common question. The **simple** answer is that it exists at the **most** mix up state possible, that is **maximally entangled**. So, let's take an a example to illustrate the concept (Doing the Maths, is beyond the scope of a Computer Scientist):

Let's say that we want to teleport a *word encoded as a quantum state*. Let's say we want to teleport *Bob*.

TABLE C.1: Teleportation of Bob

Before Teleportation	After Teleportation
$ BOB\rangle$	$ AAA\rangle +  AAB\rangle +  AAC\rangle + \dots +  ZZZ\rangle$

Let us simply prove that quantum teleportation can be done:

$$\text{Teleport}(|0\rangle + |1\rangle) = |x\rangle \cdot (|0\rangle + |1\rangle) \equiv \text{Teleport}(|0\rangle) + \text{Teleport}(|1\rangle)$$

### C.5 The Proof for Quantum Cloning

In order to clone an object, you **need** three things:

1. The thing to be copied.
2. The **raw** materials that will turn into the copy.
3. A procedure to transform the raw materials into the copy.

In physics cloning should be an exact copy down to the subatomic particles down to **every position, momentum and spin** (interactions and energy levels). It is **not** the same as *cloning in biology*, that two organisms share the same *DNA* but grow differently. It's not because we have do not know how, or it really hard to achieve, it



is boldly **impossible**. It is proved mathematically that perfect cloning **cannot** be achieved even in principle. Let us start with a generic claim:

Everything in the universe is made of elementary quantum particles and the forces by which they interact. So for the cloning proof, what does it mean to clone a quantum particle. We need to know **three** properties quantum particles share.

1. Superposition: Like the famous thought experiment of Schrodinger's Cat!  $|0\rangle + |1\rangle$  or a quantum particle wave function occupying many points at once:  $|\psi\rangle = |x_1\rangle + |x_2\rangle + \dots$  More generally  $|A\rangle = |A_1\rangle + |A_2\rangle$
2. Composite Systems: Multiple particles when viewed together as **one single** object, like an atom or entangled photons, are the product of their components. Or since this is quantum mechanics, a superposition of their components. **Generally** speaking:  $|AB\rangle = |A\rangle \cdot |B\rangle$
3. Transformation Distribution(*aka: Linear Distribution*): Any change to a particle that is in a superposition of states, it affects those states independently. **Generalizing**  $T(|A_1\rangle + |A_2\rangle) = T(|A_1\rangle) + T(|A_2\rangle)$

To recap the Cloning preliminaries and the **Mathematical System** that a **Quantum Cloning Machine** should satisfy:

1.  $|A\rangle = |A_1\rangle + |A_2\rangle$
2.  $|AB\rangle = |A\rangle \cdot |B\rangle$
3.  $T(|A_1\rangle + |A_2\rangle) = T(|A_1\rangle) + T(|A_2\rangle)$

*PS: Our machine should not know in advance what we will clone, or else it's a machine for building **unknown** things*

The problem relies if we try to clone something in **superposition**. So  $Clone(|0\rangle + |1\rangle) = (|0\rangle + |1\rangle) \cdot (|0\rangle + |1\rangle)$

So by the *Transformation Distribution* property:

$$Clone(|1\rangle) + Clone(|0\rangle) = (|1\rangle \cdot |1\rangle) + (|0\rangle \cdot |0\rangle) \neq Clone(|0\rangle + |1\rangle) = (|0\rangle + |1\rangle) \cdot (|0\rangle + |1\rangle)$$

**IF we distribute all the way:**  $Clone(|0\rangle + |1\rangle) = |0\rangle \cdot |0\rangle + |1\rangle \cdot |1\rangle + |0\rangle \cdot |1\rangle + |1\rangle \cdot |0\rangle$

Basically, if cloning and quantum mechanics are true:  $(A + B)^2$  must be the same as  $A^2 + B^2$  but  $(A + B)^2 = A^2 + 2AB + B^2$  so either quantum mechanics is **wrong** or that **cloning is wrong!** This is what we call: *proof by contradiction!*

Non cloning does **NOT** means you **cannot** have more than **one copy of something** in the universe. **It just means that you cannot take something existing inside the universe and make a clone of without knowing everything about it.** A machine to build copies of something, can be built, but it has to **know** in advance what it will make copies of!

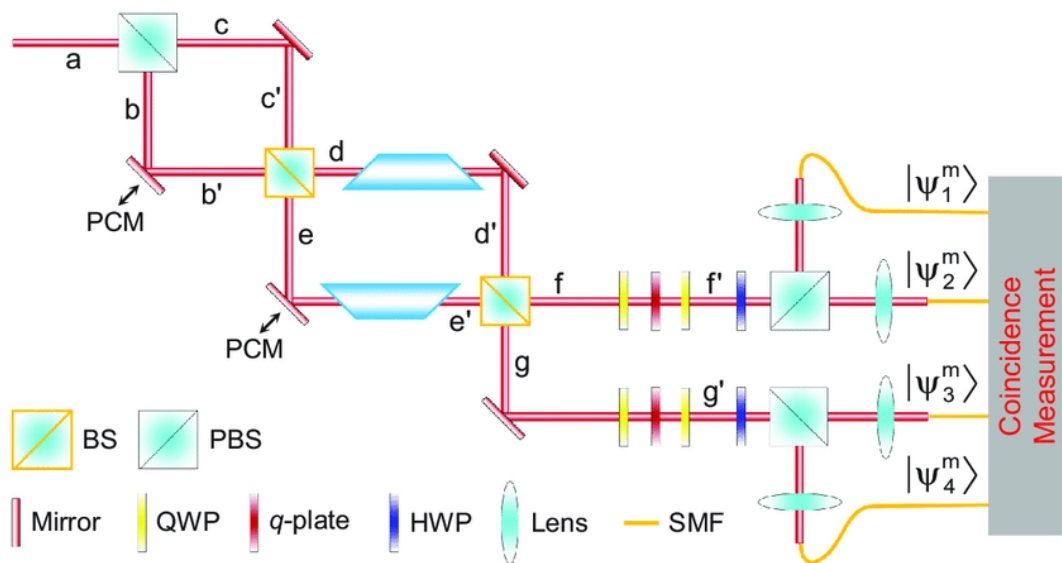


FIGURE C.3: A Schematic of a Bell State Analyzer

(Source: <https://www.researchgate.net/figure/>)

## C.6 What is a Bell State Analyzer?

In short, it is a device that can copy a Quantum state, onto another Quantum particle. Diving deeper, it's far beyond the scope of this thesis, and for a *Computer Scientist* no less, to explain further, in any order of details. This machine is based on the *Bell's Inequalities* and its phenomenon.

## C.7 Looking into Heisenberg Uncertainty Principle

In order to explain to take my best shot, at explaining a peculiar phenomenon, let us start by a common example in the *macrocosmos*. That of a speeding car on a highway with a police car chasing it. This is a very famous way of explaining HUP. So famous in fact, that became an *anecdote*. So, let's get to the example:

*The two cars pull over, and the policeman walks up to the other's car driver. He tells the car's driver that he is driving too faster than the allowed speed. The other man replied, "How can you be sure, policeman?". The policeman then raised his "speed gun" showing the other's man speed he was traveling at. He told the driver than this "speed gun" was a device that, when pointed to an object, it displays the approximate speed of the moving object.*

Now, let us circle back to *Science* and carry on. In the *macrocosmos*, it is easy to predict the **position** of a *large enough* object, and more importantly **estimate the object's position** (Moving object or not!) in a short duration of *time*. This is because the *Uncertainties* in speed **and** position of the object, are so minor that we (*Humans*) **cannot** detect them. Thus, making the *assumption* that *speed and position will not change* in the next moments.

In the *microcosmos* it is not that *simple* of a deal. The same *methodology* for estimating an objects position or speed, **does not apply** to *small objects* (like atoms, electrons and particles smaller than that!). In order to *being* able to *observe* a particle, is not a straightforward task. A photon need to bounce off that particle and be detected. In doing

so, **some** amount of momentum to the collided particle, effectively altering it's path. So according to the *principle*:

*If we know everything about the location, then we know **nothing** about its momentum, and conversely, if we know everything about its momentum, then we know **nothing about its location**.*

Generally, We **cannot** measure the **position and momentum** of a particle with absolute certainty, and in the same time. Mathematically, it is expressed like this:

$$\Delta_x \Delta_p \geq \frac{\hbar}{2}, \text{ where } x \text{ represent the particle's } \mathbf{position} \text{ and } p \text{ its } \mathbf{momentum}.$$

## Appendix D

# Clarifications: Chapter 4: Building A Quantum Computer

### D.1 Quantum Parallelism

At the base, this is the simple process of having quantum states that cancel out other quantum states. But, let's illustrate that complex process by a more concrete example.

Let us imagine a simple mathematical *function* that takes two *binary* bits as an input, and outputs a *single* bit. Let us denote that function as  $f$ .

$$f(x) : \{0, 1\}^2 \rightarrow \{0, 1\}$$

In order to evaluate all four possibilities with two bits, we need to call out the function  $f$  **four times**:  $f(0,0), f(0,1), f(1,0), f(1,1)$ . Now, with the usage of *quantum parallelism*, we are able to evaluate **all four** possibilities with a **single** call of the  $f$  function. Since all *four* possibilities of the input two bits, are mashed up into a single output, the  $f$  function is not reversible. Since, the *input* of the  $f$  function is no longer some *binary bit* and is a **Qbit** in *superposition*, all operations on Qbits, must be reversible. We modify the function  $f$  to satisfy this premise. Our **new**  $f$  function becomes:

$$Q|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$$

The  $Q$  denoted from the previous equation, and all the rest, is a Quantum function. The  $\oplus$  symbol denoted here, and so on, is for the XOR logical operation. An important thing to keep in mind is that, if we set  $y = 0$ , then the second output is just  $f(x)$ .

Now, let us write down the equation, taking into account our Q-bit in superposition.

$$|\phi\rangle = (H \oplus H)|00\rangle * \frac{|0\rangle+|1\rangle}{\sqrt{2}} \oplus \frac{|0\rangle+|1\rangle}{\sqrt{2}} = \frac{1}{2} * (|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

Finally, let us apply the our Quantum function  $Q$  and  $y$  remains equal to 0.

$$Q|\phi\rangle|0\rangle = \frac{1}{2} * Q(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \oplus |0\rangle$$

$$Q|\phi\rangle|0\rangle = \frac{1}{2} * Q(|00,0\rangle + |01,0\rangle + |10,0\rangle + |11,0\rangle)$$

$$Q|\phi\rangle|0\rangle = \frac{1}{2} * (|00, f(00)\rangle + |01, f(01)\rangle + |10, f(10)\rangle + |11, f(11)\rangle)$$

This is the simple explanation and proof that two bit with four possible values is **equal** to a **single** Qbit, in superposition of the *states* of these two bits.

## D.2 Adiabatic and Diabatic theorems

This is a Mathematical theorem, used mainly in the field of *Thermodynamics*. The easiest procedure that explains simply these two theorems, is to compare them side by side, and I assure you, the difference will become very obvious.

TABLE D.1: Diabatic and Adiabatic Theorems

Diabatic	Adiabatic
Rapidly <b>changing</b> conditions preventing the system to <b>adapt</b> it's configuration, hence, the spacial probability remains the same and the system ends up, as a linear <b>combination</b> of states. Whom the sum produces the initial probability density.	<b>Gradually</b> changing the conditions so that the system <b>can adapt</b> to it's configuration, hence, the spacial probability is modified by the process and the system ends up in a <i>corresponding</i> state, in relation to the one it started from.

## D.3 Penrose's Quantum Notation Scheme

All Quantum schematics, follow a specific scheme.

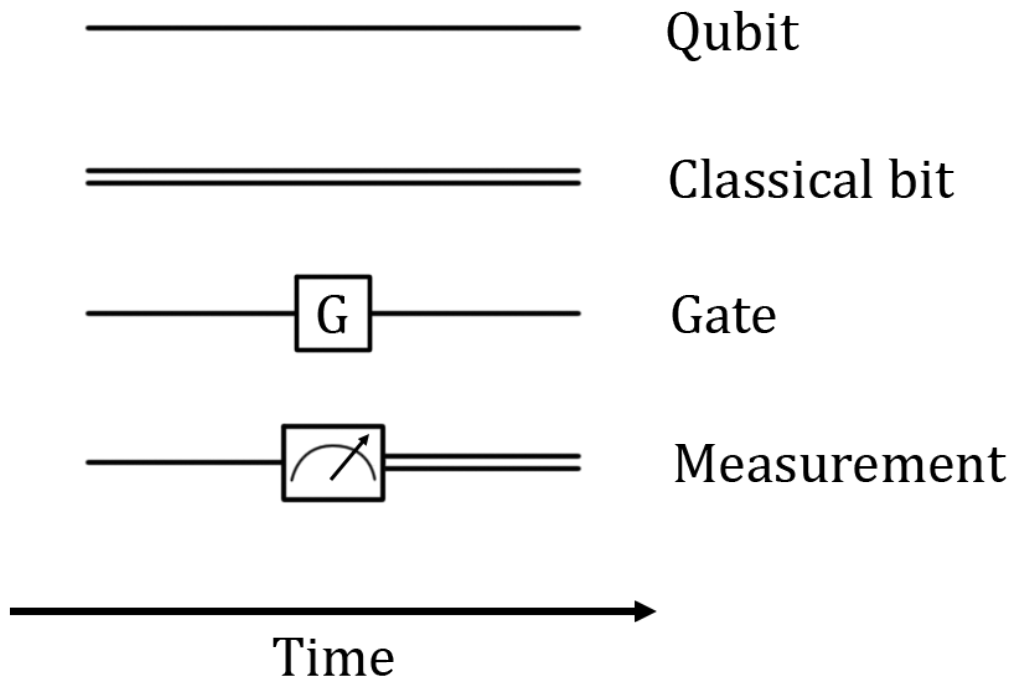


FIGURE D.1: Penrose's Quantum Notation Scheme  
 (Source: <https://techcommunity.microsoft.com/t5/image/serverpage/image-id/95812i89160AB77F8A3218?v=v2>)

Note: Lines do not represent physical cables, just sequence of events.

## D.4 Quantum Logic Gates







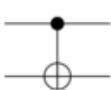


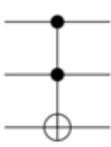
Operator	Gate(s)	Matrix
Pauli-X (X)	 $\oplus$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

FIGURE D.2: Quantum Logic Gates  
 (Source: [https://en.wikipedia.org/wiki/Quantum\\_logic\\_gate](https://en.wikipedia.org/wiki/Quantum_logic_gate))

## Appendix E

# Clarifications For Chapter 5: Securing Information in the Post-Quantum Era

### E.1 Shor's Algorithm

In 1994, Peter Shor, an American Mathematician, proposed an algorithm for finding the **prime factors** of an Integer. It was extensively used in Quantum computing. The premise of this algorithm is not that difficult to take hold. The algorithm has two parts to it:

1. A reduction of the factoring problem, in order to find the mathematical *order* of the integer. This can be done on a traditional computer.
2. A Quantum algorithm that tries to solve the problem of finding the order of the Integer. (*aka: Quantum part, Period-Finding subroutine*).

*PS: In Mathematics, The order of an Integer is to find the number of its elements.*

The aim of this algorithm is to find the square root of an integer (let's denote here as  $b$ ), of 1 modulo  $N$ .  $N \neq 1$  or  $-1$ .

$$b^2 - 1 = (b + 1)(b - 1) = mN$$

*Note:  $m$  is just a non-zero Integer.*

The algorithm goes as follows:

1. Choosing a *random number* such as  $1 < a < N$ .
2. Computing  $K = \gcd(a, N)$ .
3. If  $K \neq 1$ , the algorithm is done.
4. If  $K \neq 1$ , we use the *Period-finding subroutine* on a Quantum Computer with the period of the function:  $f(x) = a^x \pmod{N}$ .  
( $r$  is the smallest positive number, that satisfy  $a^r \equiv 1 \pmod{N}$ )
5. If  $r$  is odd, then we go back to the first step of the algorithm.
6. If  $a^{r/2} = -1 \pmod{N}$ , then we also go back to the first step of the algorithm.
7. If we arrived this far, it means that both  $\gcd(a^{r/2} - 1, N)$  or  $\gcd(a^{r/2} + 1, N)$  are **factors** of  $N$ , and the algorithm is done.

## E.2 What is a Cusp in a function?

It is a purely *Mathematical* term, and this is the reason that I should better define it here. A *Cusp* (aka: *Spinode*) is a point on a curve where a *moving point must reverse direction*. It is easier to explain with a picture.

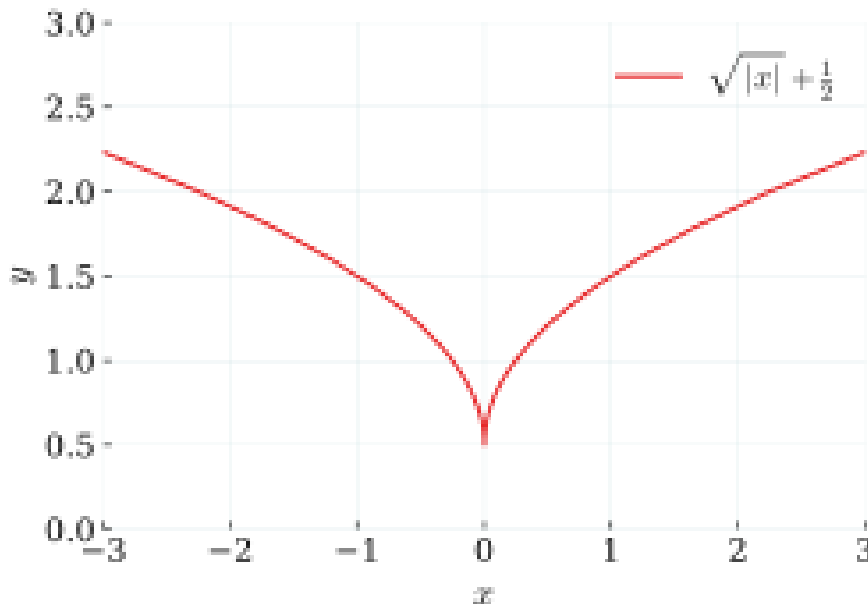


FIGURE E.1: Visual Representation Of A Cusp at (0,0)  
 (Source: [https://en.wikipedia.org/wiki/Cusp\\_\(singularity\)](https://en.wikipedia.org/wiki/Cusp_(singularity)))

Let us take two distinct functions:  $x = f(t)$  and  $y = g(t)$ . Both these functions are zero, and their *directional* derivatives, as well. So, we can say that  $\sqrt{|x|} + \frac{1}{2}$  has a cusp at point (0,0).

## E.3 The *Lattice* terminology

This is a *Mathematical* term, or to be more specific, it is commonly used in **linear algebra**. In the real coordinate system (aka: *Euclidean Plane*) is an infinite group of points with distinct values. While adding and subtracting a subset of points with each other will also form a lattice.

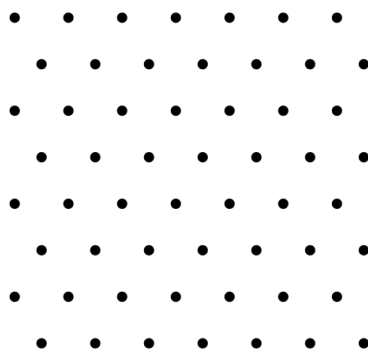


FIGURE E.2: Visual Representation Of A Lattice in Euclidean Plane  
 (Source: [https://en.wikipedia.org/wiki/Lattice\\_\(group\)](https://en.wikipedia.org/wiki/Lattice_(group)))



$L \subset \mathbb{R}^n$ . So in simple terms,  $L = \{\sum a_i b_i : a_i \in \mathbb{Z}\}$ . While,  $\{b_1, \dots, b_n\}$  of  $\mathbb{R}^n$ . In this example,  $\mathbb{Z}^n$  is also a lattice that is generated by the *standard basis* of  $\mathbb{R}^n$ . Lattices are used extensively in the SVP (*aka: Shortest Vector Problem*) and provides us the solution of the minimal Euclidean length of a non-zero lattice vector.

## E.4 Bell's Theorem

Bell's theorem is a very important statement in Quantum Mechanics. It demonstrated that a category of Physical theories called "*Local Hidden Variables Theory*" **could not** account for some degree of **correlation** between the spins of *entangled particles*. Thus by essence, **Quantum Theory** is *non-local* in some way. The theorem states:

*No theory of local realism such as local variable theory can account for correlation between entangled electrons predicted by Quantum Mechanics.*

The results of **Quantum Mechanics** experiments, has showed a large number of "loopholes" that Scientists are trying to close, for over fifty years in the field's research. In the most popular **Copenhagen interpretation of Quantum mechanics**, spins over quantum particles are shared. Even before any **measurement** is made. It is a **bold claim**, to say that something is wrong in the theorem. But like stated earlier, Scientists pound over this theorem and the math behind it, for **fifty years** and not making *significant progress* in proving or discarding part of the hypothesis that is around that theorem. This theorem is *unofficially* called as the "Most Weird Theorem".

# Bibliography

- [1] Ross Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, 2020.
- [2] L Astrand and T Yu. “Deprecate DES, RC4-HMAC-EXP, and Other Weak Cryptographic Algorithms in Kerberos”. In: *RFC 6649, RFC Editor* (2012).
- [3] Jean-Philippe Aumasson. *Serious Cryptography: A Practical Introduction to Modern Encryption*. No Starch Press, 2017.
- [4] Gustavo Banegas et al. “Concrete quantum cryptanalysis of binary elliptic curves”. In: *Cryptology ePrint Archive* (2020).
- [5] Charles H Bennett. “Quantum cryptography using any two nonorthogonal states”. In: *Physical review letters* 68.21 (1992), p. 3121.
- [6] Charles H Bennett and Gilles Brassard. “Quantum cryptography: Public key distribution and coin tossing”. In: *arXiv preprint arXiv:2003.06557* (2020).
- [7] Charles H Bennett, Gilles Brassard, and N David Mermin. “Quantum cryptography without Bell’s theorem”. In: *Physical review letters* 68.5 (1992), p. 557.
- [8] Charles H Bennett et al. “Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels”. In: *Physical review letters* 70.13 (1993), p. 1895.
- [9] Daniel J Bernstein. *Introduction to post-quantum cryptography*. Springer, 2009.
- [10] Dagmar Bruß. “Optimal eavesdropping in quantum cryptography with six states”. In: *Physical Review Letters* 81.14 (1998), p. 3018.
- [11] Dagmar Bruss and Chiara Macchiavello. “Optimal eavesdropping in cryptography with three-dimensional quantum states”. In: *Physical review letters* 88.12 (2002), p. 127901.
- [12] Stanislav Bulygin, Albrecht Petzoldt, and Johannes Buchmann. “Towards provable security of the unbalanced oil and vinegar signature scheme under direct attacks”. In: *Progress in Cryptology-INDOCRYPT 2010: 11th International Conference on Cryptology in India, Hyderabad, India, December 12-15, 2010. Proceedings 11*. Springer. 2010, pp. 17–32.
- [13] Vladimir Bužek and Mark Hillery. “Universal optimal cloning of arbitrary quantum states: from qubits to quantum registers”. In: *Physical review letters* 81.22 (1998), p. 5003.
- [14] Hughes C. *Quantum Computing for the Quantum Curious*. Springer, 2022.
- [15] John Calsamiglia and Norbert Lütkenhaus. “Maximum efficiency of a linear-optical Bell-state analyzer”. In: *Applied Physics B* 72.1 (2001), pp. 67–71.
- [16] Matt Campagna et al. “Kerberos revisited quantum-safe authentication”. In: *ETSI Quantum-Safe-Crypto Workshop*. 2013, pp. 26–27.
- [17] Nicolas J Cerf, Gerd Leuchs, and Eugene S Polzik. *Quantum information with continuous variables of atoms and light*. World Scientific, 2007.

- [18] *Commercial National Security Algorithm Suite*. Tech. rep. NSA, 2019. URL: <https://apps.nsa.gov/iaarchive/programs/iad-initiatives/cnsa-suite.cfm>.
- [19] Augot Daniel, B Lejla, et al. "Initial recommendations of long-term secure post-quantum systems". In: *PQCRYPTO. EU. Horizon 2020* (2015).
- [20] David Deutsch. "Quantum theory, the Church–Turing principle and the universal quantum computer". In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117.
- [21] Orr Dunkelman and Eli Biham. "A framework for iterative hash functions: Haifa". In: *2nd NIST cryptographich hash workshop*. Vol. 22. 2006.
- [22] Artur K Ekert. "Quantum cryptography based on Bell's theorem". In: *Physical review letters* 67.6 (1991), p. 661.
- [23] Daphna G Enzer et al. "Entangled-photon six-state quantum cryptography". In: *New Journal of Physics* 4.1 (2002), p. 45.
- [24] Edward Farhi et al. "Quantum computation by adiabatic evolution". In: *arXiv preprint quant-ph/0001106* (2000).
- [25] Jan Faye. "Copenhagen interpretation of quantum mechanics". In: (2002).
- [26] Richard P Feynman. "Quantum mechanical computers". In: *Optics news* 11.2 (1985), pp. 11–20.
- [27] Aleta Berk Finnila et al. "Quantum annealing: A new method for minimizing multidimensional functions". In: *Chemical physics letters* 219.5-6 (1994), pp. 343–348.
- [28] Lance Fortnow. "One complexity theorist's view of quantum computing". In: *Theoretical Computer Science* 292.3 (2003), pp. 597–610.
- [29] Chi-Hang Fred Fung, Kiyoshi Tamaki, and Hoi-Kwong Lo. "On the performance of two protocols: SARG04 and BB84". In: *arXiv preprint quant-ph/0510025* (2005).
- [30] Jason Garman. *Kerberos: The Definitive Guide*. " O'Reilly Media, Inc.", 2003.
- [31] Lizal Iswady Ahmad Ghazali et al. "Security proof of Improved-SARG04 protocol using the same four qubit states". In: *International Conference On Photonics 2010*. IEEE. 2010, pp. 1–4.
- [32] Nicolas Gisin et al. "Quantum cryptography". In: *Reviews of modern physics* 74.1 (2002), p. 145.
- [33] John Gribbin. *Quantum Phisics*. DK Essential Science, 2002.
- [34] Michaël Peeters Guido Bertoni Joan Daemen and Gilles Van Assche. "Sponge Functions". In: *Ecrypt Hash Workshop 2007* (2017).
- [35] Laszlo Gyongyosi and Sandor Imre. "A survey on quantum computing technology". In: *Computer Science Review* 31 (2019), pp. 51–71.
- [36] Monica Heger. "Cryptographers take on quantum computers". In: *IEEE Spectrum* 46.1 (2008), pp. 14–14.
- [37] Martin E. Hellman. "An overview of public key cryptography". In: *IEEE Communication Magazine* (2002).
- [38] Tony Hey. "Quantum computing: an introduction". In: *Computing & Control Engineering Journal* 10.3 (1999), pp. 105–112.

- [39] David Holmes. *RSA in a "Pre-Post-Quantum" Computing World*. Tech. rep. f5, 2020. URL: <https://www.f5.com/labs/articles/threat-intelligence/rsa-in-a-pre-post-quantum-computing-world>.
- [40] David Jao and Luca De Feo. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies". In: *Post-Quantum Cryptography: 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29–December 2, 2011. Proceedings 4*. Springer. 2011, pp. 19–34.
- [41] Mark W Johnson et al. "Quantum annealing with manufactured spins". In: *Nature* 473.7346 (2011), pp. 194–198.
- [42] Amir H Karamlou et al. "Analyzing the performance of variational quantum factoring on a superconducting quantum processor". In: *npj Quantum Information* 7.1 (2021), pp. 1–6.
- [43] Taehyun Kim. *Applications of single-photon two-qubit quantum logic to the quantum information science*. Tech. rep. 2008.
- [44] Neal Koblitz. "Elliptic curve cryptosystems". In: *Mathematics of computation* 48.177 (1987), pp. 203–209.
- [45] John T Kohl, B Clifford Neuman, and Y Theodore. "The evolution of the Kerberos authentication service". In: (1994).
- [46] Göran Lindblad. "A general no-cloning theorem". In: *Letters in Mathematical Physics* 47.2 (1999), pp. 189–196.
- [47] Navin B Lingaraju et al. "Bell state analyzer for spectrally distinct photons". In: *Optica* 9.3 (2022), pp. 280–283.
- [48] PV McMahon. "SESAME V2 public key and authorisation extensions to Kerberos". In: *proceedings of the symposium on Network and Distributed System Security*. IEEE. 1995, pp. 114–131.
- [49] Christopher Monroe et al. "A "Schrödinger cat" superposition state of an atom". In: *Science* 272.5265 (1996), pp. 1131–1136.
- [50] Dustin Moody and LILY Chen. "The 2nd round of the nist pqc standardization process". In: *Online: <https://csrc.nist.gov/CSRC/media/Presentations/the-2nd-round-of-the-nist-pqc-standardization-proc/images-media/moody-opening-remarks.pdf> [accessed: April 2021]* (2019).
- [51] Roger M Needham and Michael D Schroeder. "Using encryption for authentication in large networks of computers". In: *Communications of the ACM* 21.12 (1978), pp. 993–999.
- [52] Tadayoshi Kohno Niels Ferguson Bruce Schneier. *Cryptography Engineering: Design Principles and Practical Applications*. Wiley, 2010.
- [53] *Open Quantum Safe*. URL: <https://openquantumsafe.org/>.
- [54] T. Shrimpton (2004) P. Rogaway. "Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance". In: *Springer-Verlag* (2012).
- [55] Marmik Pandya. "Securing Cloud-The Quantum Way". In: *arXiv preprint arXiv:1512.02196* (2015).
- [56] James L Park. "The concept of transition in quantum mechanics". In: *Foundations of physics* 1.1 (1970), pp. 23–33.

- [57] Geovandro CCF Pereira, Cassius Puodzius, and Paulo SLM Barreto. “Shorter hash-based signatures”. In: *Journal of Systems and Software* 116 (2016), pp. 95–100.
- [58] Rick van Rein and Tom Vrancken. *Quantum Relief with TLS and Kerberos*. Internet-Draft draft-vanrein-tls-kdh-08. Work in Progress. Internet Engineering Task Force, Oct. 2022. 22 pp. URL: <https://datatracker.ietf.org/doc/draft-vanrein-tls-kdh/08/>.
- [59] James H. Reinholm. “Classification of Cryptographic Keys (Functions & Properties)”. In: *Cryptomathic* (2017).
- [60] Eleanor Rieffel and Wolfgang Polak. “An introduction to quantum computing for non-physicists”. In: *ACM Computing Surveys (CSUR)* 32.3 (2000), pp. 300–335.
- [61] RIKEN. “It takes three to tangle: Long-range quantum entanglement needs three-way interaction.” In: *ScienceDaily* (2022). URL: [www.sciencedaily.com/releases/2022/05/220506113326.htm](http://www.sciencedaily.com/releases/2022/05/220506113326.htm).
- [62] Ronald L Rivest and Jacob CN Schuldt. “Spritz—a spongy RC4-like stream cipher and hash function.” In: *Cryptology ePrint Archive* (2016).
- [63] Damien Robert. “Breaking SIDH in polynomial time”. In: *Cryptology ePrint Archive* (2022).
- [64] Martin Roetteler et al. “Quantum resource estimates for computing elliptic curve discrete logarithms”. In: *Advances in Cryptology—ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II* 23. Springer. 2017, pp. 241–270.
- [65] David J. Griffiths & Darrell F. Schroeter. *Introduction to Quantum Mechanics (2nd ed.)* Prentice Hall, 2004.
- [66] Jennifer G Steiner, B Clifford Neuman, and Jeffrey I Schiller. “Kerberos: An Authentication Service for Open Network Systems.” In: *Usenix Winter*. 1988, pp. 191–202.
- [67] Hiroki Takesue et al. “Quantum teleportation over 100 km of fiber using highly efficient superconducting nanowire single-photon detectors”. In: *Optica* 2.10 (2015), pp. 832–835.
- [68] Takeji Takui, Lawrence Berliner, and Graeme Hanson. *Electron spin resonance (ESR) based quantum computing*. Vol. 31. Springer, 2016.
- [69] Jim Utsiler. *Quantum Computing Might Be Closer Than Previously Thought*. Tech. rep. IBM, 2013.
- [70] JAW Van Houwelingen et al. “Experimental quantum teleportation with a three-Bell-state analyzer”. In: *Physical Review A* 74.2 (2006), p. 022303.
- [71] JR Weber et al. “Quantum computing with defects”. In: *Proceedings of the National Academy of Sciences* 107.19 (2010), pp. 8513–8518.
- [72] William K Wootters and Wojciech H Zurek. “A single quantum cannot be cloned”. In: *Nature* 299 (1982), pp. 802–803.
- [73] Pei Zeng, Jinzhao Sun, and Xiao Yuan. “Universal quantum algorithmic cooling on a quantum computer”. In: *arXiv preprint arXiv:2109.15304* (2021).