SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Computational Science and Engineering

Modelling of Satellite-Based Quantum Key Distribution

Pedro Muñoz Jiménez

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Computational Science and Engineering

Modelling of Satellite-Based Quantum Key Distribution

Modellierung der satellitenbasierten Quanten-Schlüsselverteilung

Author:	Pedro Muñoz Jiménez
Examiner:	Tobias Vogl
Supervisor:	Davide Orsucci
Submission Date:	10.06.2025

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, 10.06.2025 Pedro Muñoz Jiménez

Acknowledgments

First of all, I thank my supervisor Davide Orsucci for his guidance and support. Also Tobias Vogl, without him this would not have been possible either. Special thanks to Philipp Kleinpaß, who also provided important help. And in general thanks to the DLR and TUM.

And, of course, I thank all of my family and friends and every one who were there in this journey along in any way.

Abstract

Quantum Key Distribution (QKD) offers a fundamentally secure method for key exchange, grounded in the principles of quantum mechanics rather than computational complexity. As advancements in quantum computing threaten classical cryptographic systems (such as RSA) QKD presents itself as a crucial technology for future secure communications. Among the primary challenges for global-scale QKD is the limited range of optical fiber channels, making satellite-based free-space optical links an attractive alternative. QKD presents unique constraints related to the short and unreliable communication windows, due to atmospheric conditions and the presence of sunlight, which makes the real time operation of QKD constellations almost impossible. Traditional configurations offering instantaneous global coverage, such as Walker constellations, are not necessary for QKD networks and other constellation designs may be explored. In this work, we present as an alternative a two-height, single orbital plane constellation, which facilitates regular and extended inter-satellite visibility, enabling the constellation to function as a virtual trusted node and increasing network capacity and flexibility. The main technical contribution is the development of a simulation tool designed to evaluate and optimize satellite QKD constellations. The simulator allows researchers and mission planners to assess key generation rates and overall system performance, helping to reduce the risks, costs, and uncertainties involved in satellite deployment. Our approach provides a powerful framework for advancing the development of scalable and efficient global QKD networks.

Contents

1. Introduction

QKD is a modern technology that allows for secure communications whose security does not rely on computational complexity assumptions, as the current encryption methods do. Instead, it leverages the physical laws of quantum mechanics to establish a communication channel between two users and, if the protocol succeeds, it can be guaranteed that any potential eavesdropper could have no information about the generated key. Presently, this is a relevant topic because the current advances on quantum computing promise to break the currently employed asymmetric cryptographic algorithms, therefore, to potentially compromise all the communications. For example, most of the current encryption methods use the RSA system, which relies on the enormous computational complexity that is required for factoring large semi-primes [RSA78]. However, in 1997, Peter Shor presented an efficient quantum algorithm for factorization [Sho97], and, although it is yet to be implemented in real quantum computers, it could compromise the security of communications in the future.

Moreover, any information transmitted via a classical channel nowadays is susceptible of being stored by any eavesdropper. This is a potential risk, because the hacker can simply wait for technological developments (such as quantum computers that are capable of performing the Shor algorithm) and decipher highly sensitive information in the future.

The main advantage of QKD is that it avoids computational assumptions altogether and therefore is not susceptible to attacks by quantum computers or other algorithmic advancements. Instead QKD is only based on the fact that quantum mechanics is a valid theory that correctly describes the fundamental nature laws.

QKD is a technology that is reaching a sufficient level of maturity and it is starting to become commercially available. However, one of the main limitations for global quantum communications is the lack of an efficient quantum repeater technology. Optical fibers (which is the most commonly used channel for this type of links) introduce an absorption which, despite being low ($\sim 0.2 db/km$), scales exponentially with the distance. This limits the communication distance to a few hundred kilometers. That is why present efforts are being made on the use of free-space optical communications via satellites to build long distance quantum links. This method eliminates the problem of a high-absorption channel and, therefore, allows for a global communication range.

There are multiple satellite based QKD projects that are working in this direction, some of them with successful results such as [Lia+17] or [Li+25], and some in preparation such as [Cal+24]. In this state of the field, it is important to have access to of useful simulators since, oftentimes, these systems have a lot of free parameters whose optimal value can not be computed analytically. Furthermore, satellite tests not only involve certain risks but also take a big amount of effort, time and money and, once a satellite is in orbit, we cannot change its hardware. So it is not feasible to simply run experiments without having carried out a careful study of the performance of the system beforehand.

1.1. Challenges of real time operation of satellite QKD networks

As the technology develops, satellite constellations will be soon required in order to provide enough key for all the network users for real-time communications. QKD communications need a direct line of sight between the satellite and the ground stations, as in classical communications. However, QKD also has important differences compared to classical communications so that solutions that are typically employed there are not necessarily the best ones for QKD. In a traditional setting (such as Starlink) Walker constellations are usually the preferred solution because they enable global coverage at any point in time. For QKD other constellation types may be more suitable.

The main observation is that in the foreseeable future it will be essentially impossible to operate QKD in real time. This means that days or even weeks may be needed before a key is actually delivered to a ground station. This is because of the following reasons:

- Use of Low Earth Orbit (LEO): Diffraction and pointing losses increase with the distance, that is why low orbits are typically used. However, this kind of orbits only allow for short passes of a few minutes per day over each ground station.

- Presence of cloud blockage: Optical communications use photons in the visible or near infrared range which are completely blocked in the presence of clouds or fog.

- Night time communications: Quantum communications use strongly attenuated laser pulses, which contain on average less than one photon per pulse. During day, the presence of sunlight introduces a lot of noise into the quantum communication because of background radiation. That makes quantum communications by day very challenging. Therefore our assumption is that satellite based QKD links will only be performed by night.

However, for successful operations of QKD real time key delivering is actually not required.

- This is because QKD is only employed to exchange symmetrical key pairs, which are used then to encrypt information afterwards. For instance, this could be done by one time pad, so that the output would be purely random for anyone who does not possess the key, making it safe against all quantum attacks, no matter the potential technological breakthroughs. This means that the key distribution between the stations and the satellite only needs to happen at some time previous to the communication between the users, while classical channels are employed for the encrypted information exchange.
- In a trusted node architecture, the satellites are trusted by the end user on the ground. Once QKD keys have been exchanged between satellite and ground stations, a pair of these can be combined to create a secret key between end users using one time pad. This key combination procedure can be done only using classical communication via radio link which are not blocked by clouds or by sunlight and thus can be performed in real time.
- Therefore, in trusted node satellite QKD we may assume that keys are generated between satellites and ground stations weeks or months before their use. If two end users issue a secret key from the network and both have already established a key with a satellite, then the final key combination step can be performed in real time via radio links.

1.2. Operation of a QKD constellation as a virtual trusted node

Until a few QKD users are present, a single satellite my be sufficient to serve all of them, but at a certain point multiple satellites will be required to provide a QKD link to a large number of users.

A first mode of operation is to use multiple satellites as completely independent trusted nodes. This offers the easiest implementation, since no quantum communication between satellites is required. the downside is that the satellites are not operated as a network but independently from each other. This means that if two users have connected to different satellites they can not establish a secure key between them.

The second mode of operation is to employ inter-satellite QKD links to securely synchronize the information among the satellites in the constellation. This allows to operate the satellite network as a single virtual trusted node. This second mode of operation is potentially much better performing but requires the use of inter-satellite QKD links. This will be the focus of the present work.

1.3. Two-heights, single-orbital plane constellation

Given the points above, we hypothesize that it is not necessary to build a Walker constellation to achieve global coverage. Furthermore, a Walker constellation could hinder inter-satellite connections due to the nature of the passes on the configuration: Satellites orbit the Earth in non-parallel orbits and the crossing times between them are short (only a few minutes). The consequence is that it is very complicated to establish a stable optical link between two satellites because a line of sight without excessive jitter or pointing errors must exist between them.

Hence, we suggest a different kind of configuration to overcome this issue: a two-height single-orbital plane constellation. The idea is to employ, instead of a Walker constellation with many orbits, a single orbit where all the satellites are placed in a ring-like configuration with roughly constant inter-satellite distances. We can do this because we do not need to cover the whole planet's surface at every single minute, we only need to update the key with sufficiently high frequency. The use of a single-orbital plane also has the advantage that all the satellites may be put into orbit with a single rocket launch. However, it is not efficient to perform inter-satellite communication with only one orbit, because satellites would need to be very close to each other. This would either require a lot of satellites (with the costs and risks derived from that) or very close trailing satellites (with a very limited coverage). The solution we propose to this is to include a second orbit in the same plane but at a slightly higher altitude. The satellites of the upper trajectory will move slightly more slowly around the Earth and, as a consequence, every two satellites of different altitudes will see each other regularly in a long and slow pass. This kind of passes make it possible to expediently exchange a large amount of keys between the satellites, which allows using them as as a single virtual trusted node.

In conclusion, the use of a two-height single-orbital plane QKD constellation could enable a QKD service with higher capacity and flexibility, compared to the (default) case where two QKD users need to connect to the same satellite in order to exchange a private key.

2. Quantum Key Distribution

Quantum Key Distribution is a key exchange method that leverages the laws of quantum mechanics to establish a safe communication channel between the legitimate users usually called Alice and Bob. All of the procedure relies on the no-cloning theorem of quantum mechanics which states that it is impossible to create an independent and identical copy of an arbitrary unknown quantum state.

The implication of this is that any bit of information sent by Alice in the form of a quantum state (qubit) cannot be cloned by a potential eavesdropper (Eve). Moreover, Eve cannot interact with the quantum state (e.g. using entanglement) without introducing disturbances in the quantum channel that can be detected by the users Alice and Bob. This also means that, since quantum information is volatile, Eve must break into the connection in real time, otherwise the information will be safe forever.

2.1. BB84 protocol

There are different protocols to realize QKD. The most popular one, because of its practical feasibility, is BB84, named after its inventors Bennett and Brassard [BB84].

The ideal version of the protocol, based on a single-photon source, works as follows:

1. The sender (Alice) randomly encodes a single photon (qubit). There are different ways in which a photon can be encoded: physical polarization, time-bin encoding... In the case of polarization, Alice could use one of the four polarization states: vertical/horizontal (horizontal basis) 45-degree/135-degree (diagonal basis), and sends the photon to Bob.

   The original protocol states a 50% probability of choosing each basis. However, one can increment the efficiency with a biased version of the protocol by choosing one basis with probability p_X and the other with probability $p_Z = 1 - p_X$

2. For each qubit that is received, Bob selects one of the two bases to perform a measurement on the polarization of the received photon. Then, Alice and Bob publicly announce their basis choices via a classical channel. Since they announce the basis and not the results, no information about the key is leaked.

3. Alice and Bob discard the bits whose measurement basis and encoding basis do not match. The remaining data in the Z basis is the sifted key. To verify the security of the channel, Alice and Bob reveal the bits in the X basis and compare them to compute the Quantum Bit Error Rate (QBER).
4. If the QBER is too high, the protocol is aborted because the channel privacy can not be guaranteed. Otherwise, they proceed with classical post-processing to generate a secret key.

The two bases utilized here are associated to non-commuting (incompatible) observables, therefore, information about both of them cannot be obtained, i.e., there is no way to obtain information from the two bases at the same time. For example, consider the case when Alice and Bob choose the two bases with equal probability and Eve also does the same. If Eve tries to measure the photons, she will choose a wrong measurement basis 50% of the time. When the wrong basis is selected, 50% of the bits between Bob and Alice will differ, so Eve has introduced a 25% QBER. If Eve does not measure all the photons, the QBER will be lower but also the information that she gains will be less. In general there is a tradeoff between the info she can gain and the disturbance (QBER) she introduces i.e. *Eve can only gain information at the cost of introducing disturbance* [Xu+20]

In practical implementations one would often use laser pulses, because it is much easier and allows for higher repetition rates. However, these can have multi-photon components, which are insecure. Hence, one also has to modulate the intensity of the laser, through the so called decoy state method (sec. 2.6) to estimate how many (potentially insecure) multi-photon events contributed to the key generation.

2.2. Security definition

A secure key must be correct and secret:

- Correctness: the key bits that Alice and Bob have need to be identical
- Secrecy: the key bit string should be uniformly distributed to anyone other than Alice and Bob

Of course, this is an idealization. Due to finite key size and imperfect error correction, every key is assumed to have a small failure probability $\epsilon = \epsilon_{sec} + \epsilon_{cor}$ and we say that the protocol is ϵ-secure.

In a mathematical way we can define it as [Ben+05]:

$$\min_{\rho_E} \frac{1}{2}(1 - p_{\text{abort}}) \left\| \rho_{ABE} - \rho_{ABE}^{\text{ideal}} \right\|_1 \leq \epsilon \tag{2.1}$$

1. p_{abort} is the probability that the protocol aborts
2. ρ_{ABE} is the quantum state that Alice, Bob and Eve share after the communication is done, which includes all the qubits sent by Alice and whatever modification that Eve could have performed on them. Essentially, it describes how much information that each of them has about the other qubits. Some protocols use entangled photons which are described by this notation, while as others, such as BB84 use a Prepare and Measure (PM) scheme, but both are mathematically equivalent.

   The mathematical description of the state is:

   $$\rho_{ABE} = 2^{-m} \sum_{k_A,k_B} \Pr(k_A, k_B) \left|k_A\right\rangle_A \left\langle k_A\right| \otimes \left|k_B\right\rangle_B \left\langle k_B\right| \otimes \rho_E^{(k_A,k_B)} \tag{2.2}$$

   Where $k_A, k_B \in \{0,1\}^m$ are the bit values and $\rho_E^{(k_A,k_B)}$ is the state of Eve.
3. $\rho_{ABE}^{\text{ideal}}$ is the ideal quantum state that Alice and Bob share after the communication is done. A and B share a perfect entanglement while as Eve just shares a product state with them (no information can be acquired from A and B by measuring E).

   The mathematical description of the ideal state is:

   $$\rho_{ABE}^{\text{ideal}} = 2^{-m} \sum_{k} \left|k\right\rangle_A \left\langle k\right| \otimes \left|k\right\rangle_B \left\langle k\right| \otimes \rho_E \tag{2.3}$$

   Where $k = k_A = k_B$, which means that they share exactly the same key. Also the state of Eve ρ_E is completely independent of k

In a nutshell, what this equation describes is how close is the real quantum state to the ideal secret and correct state.

2.3. General QKD protocol description

A QKD protocol, as a general concept, consists in the following steps. Strictly speaking, the only quantum part is the first step, the rest consists in classical post-processing, which relies on bi-directional classical communication. All the classical information must be authenticated, even if it is encrypted; both can be accomplished by using pre-shared randomness. This could originate from previous QKD rounds or as a first initialization at system commissioning stage.

The protocol steps are [Ors+25]:

1. *Quantum communication*: Alice and Bob exchange N signals over a quantum channel (maybe involving a non trusted third party). Quantum signals are measured upon being received. In the end we get $N_d \leq N$ successful detection events. The state preparation and measurement settings are chosen randomly (e.g., employing a quantum random number generator [Ma+16]) and this information, as well as the measurement outcomes, is stored locally by Alice and Bob.

2. *Raw key extraction or key sifting*: Alice and Bob announce part of the information (e.g., the employed basis for the quantum state preparation and measurement) via the authenticated channel, use this information to select N_s out of the N_d detection events and employ some of the corresponding data (e.g., the measurement outcomes when Alice and Bob employ matching bases) to extract the sifted keys, k_A and k_B, of length of length ℓ_{raw}. The key mapping function might involve further random choices (e.g., applying a random permutation to the order of the sifted key bits) or compress the data (e.g., mapping a continuous value to a discrete value in continuous-variable protocols).

3. *Parameter estimation*: Part of the key may be used to estimate the quality of the quantum signal (e.g., the QBER in one basis). These estimations will be employed to tune the protocol parameters in the next steps.

4. *Information reconciliation or error correction*: The sifted keys kA and kB extracted usually differ on a small percentage. Once again, the classical channel may be used to run an error correction protocol, resulting on the keys k'_A and k'_B which are equal with a very high confidence.

5. *Error verification or hashing*: Alice and Bob apply a hash function to the respective keys (k'_A and k'_B) and discard them if the result doesn't match. The hash function guarantees that the remaining keys are equal, except for a small ϵ_{corr} probability of hash collision.

6. *Privacy amplification*: There is a way to set an upper bound to the amount of information that Eve can gather: if the right 2-universal hash function is applied to the key it yields a bit string of length $\ell \leq \ell_{raw}$. According to the quantum left-over hash lemma [Tom+11] the resulting string will appear as a uniformly random bit string from any external perspective (except for ϵ_{sec} failure probability. For this case we assume the worst case, which is the one in which the complete key has been leaks be leaked)

7. *Authentication*: All the classical information that has been exchanged is authenticated (for example, using pre-shared secret randomness to sign the messages). By construction, Eve has at most a probability of ϵ_{auth} of correctly signing a forged message exchange.

During steps 2, 3, 4, and 5, if the security conditions are not met (e.g., key is too short, QBER is too high, or the error correction and verification step fail), the protocol is aborted. Otherwise, a secure key of length ℓ is obtained. After information reconciliation, the probability δ that the keys differ, $Pr(k_A \neq k_B)$, is typically around 10^{-2}–10^{-6} and depends on the QBER and on the error correction scheme. Since hashing is employed to discard unequal keys, a high value of δ may decrease the protocol success probability but does not compromise the protocol security.

Overall, assuming a constant quantum channel quality, more signals result in longer keys, while more stringent security and correctness parameters result in shorter ones.

Error correction

For each communication round there must be an error correction. Since the efficiency of an error scheme can approach the Shannon limit only asymptotically, we introduce an efficiency factor $f > 1$. This means that on average $fh(\text{QBER})$ bits of information are revealed for error corrections.

There are several error correction methods that could be implemented, for instance, Cascade methods [BS94], which however require many interactions. Another common type of methods are the forward methods which only take one communication exchange. Among this methods the Low Density Parity Check (LDPC) [MN96] is the most common. The LDPC code works well for QKD due to its high error correction efficiency and very limited communication rounds requirements. The design and optimization of LDPC codes in QKD postprocessing is similar to the classical case.

Error verification

The error correction is not guaranteed to succeed or return equal results for Alice and bob, so they have to make sure that they share the same key. This is achieved through a hashing function $F(k)$ which takes as input a string k of arbitrary length l and outputs a string of fixed length k_{ev} (also called hash). Alice and Bob then compute the hashes of k_A and k_B respectively and compare the result.

The interesting thing of hash functions is that they are, by construction (almost) uniform over the space of the outputs, so the probability that they are (almost) uniform over the space of the outputs so the probability of $k_A \neq k_B$ but same hash is approximately $2^{-k_{\text{ev}}}$. By doing so, however, they reveal k_{ev} bit of information.

Privacy amplification

To mitigate finite-size effects, privacy amplification has proven to be very efficient. Once the keys have been exchanged, corrected and verified. Alice shares a randomly chosen hashing matrix $T \in \{0,1\}^{lxn}$ and sends it to Bob.

The final key will be $Tk_A = Tk_B$ [Xu+20]

2.4. Security assumptions

1. Source: The source is said to be basis independent. So whatever the qubit encoding is, the quantum state is indistinguishable before the measurement. Therefore, both states are completely indistinguishable to Eve. In the case of two polarization states:

   $$\rho_X = \rho_Z \tag{2.4}$$

   Where $\rho_\alpha = \frac{1}{2}(\rho_{\alpha,0} + \rho_{\alpha,1})$, which is the mathematical description of the quantum state of the superposition in one of the two polarizations.

   This also means that in QBER estimation, one can use one basis computation to estimate the other. [Xu+20]

   In practice, single-photon sources are expensive and not fully reliable. Therefore Weak Coherent State (WCS) sources are usually the chosen solution. In this case, the indistinguishability property given above only has to hold on the single-photon subspace

2. Measurement: The measurement is also required to be basis-independent. In our particular case:

   $$M_X = M_Z \tag{2.5}$$

   Where $M_\alpha = \frac{1}{2}(M_{\alpha,0} + M_{\alpha,1})$, which are the four POVM elements that represent the four possible outcomes of the measurement.

   However, while for source, one only needs to guarantee its basis-independent property, on measurement, it must be specific projection measurements. In a real setting, the source requirement is easier to meet than the measurement one, hence, there are more practical security issues on measurement than on source. [Xu+20]

3. Channel: For security proofs, the channel is always assumed to be under the absolute control of the potential eavesdropper. Therefore we do not impose any requirement on the channel. Moreover if any implementation deviation from the

ideal QKD protocol can be moved into the channel, it will not cause any security issues.

2.5. Quantum Hacking

2.5.1. Photon number splitting attack

The first quantum hacking attack that was thought of was the Photon-Number-Splitting (PNS) attack [Bra+00]. This attack is possible leveraging the weaknesses of using highly attenuated lasers instead of true single-photon sources. These lasers generate WCSs that, often, do not generate single photons, but rather multi-photon pulses. It is theoretically possible to perform a Quantum nondemolition measurement to count the number of photons in a pulse. Then, Eve can block all of the single-photons and split the rest in two, from which she can keep one part and send the rest to Bob. Later, can get all the information during the sifting phase, without inducing any errors.

This attack induced many doubts about the feasibility of quantum communications, but it was later shown that it can be easily avoided by using a decoy-state method (See 2.6) that guarantees the availability of enough single photon signals.

2.5.2. Other attacks

There are many other attacks that can be performed on a QKD system, being the detector the most vulnerable part of the system because it must receive signals that Eve can easily send.

However attacks can be performed at source, channel or receiver. Here is a list of some of the possible attacks, which are elaborated in [Mak+24]

- Superlinear detector control
- Detector efficiency mismatch
- Detector deadtime attack
- Trojan-horse attack
- Laser-seeding attack
- Light injection into Alice's powder meter
- Induced photo-refraction attack
- Laser damage

- APD blackflash
- Intersymbol interference
- Imperfect individual state preparation
- Calibration performed via channel Alice-Bob
- Quantum random number generator
- Compromised chain of supply
- etc.

2.6. Decoy state method

To defend against the PNS attack, requires Alice to adjust the intensity of her source so the amount of multi-photon events is minimal. For coherent sources it has been shown that the optimal intensity is close to the channel transmittance [Xu+20]:

$$\mu_{opt} \approx \eta \tag{2.6}$$

However, even in this scenario, multi-photon events can cause an information leakage because the photon number in a WCS follows a Poisson distribution.

2.6.1. Estimation of the amount of single photon events via decoy state

Many protocols have been proposed as a solution to PNS attacks, but one of the most successful ones is the Decoy-State protocol [Ma+05]. Experimental realization of a fully successful PNS attack is yet to be finished, but some recent experiments with promising results have been made on this direction [Ash+24]. That is why decoy state method is usually implemented as a preventive method. That is why we also incorporate this method in our simulator.

The idea is the following: instead of one single intensity, different intensities are used and they are chosen randomly for each signal. These decoy sates are then employed to monitor the transmittance of different photon components

In summary, the decoy state method allows to estimate the channel parameters of yield Y_n (conditional probability that a detector registers a click (i.e., a detection event) given that a signal containing exactly n photons was sent through the quantum channel) and QBER e_n for each photon number. The protocol security relies on the fact that the signal state and the decoy states are identical except for the average number of photons. The result of this is that Eve has no way to know whether the qubit she intercepted corresponds to a signal or a decoy state. Hence, Y_n and e_n can only depend on the number of photons n and not on the full distribution:

$$Y_n(signal) = Y_n(decoy) \tag{2.7}$$

$$e_n(signal) = Y_n(decoy) \tag{2.8}$$

The decoy state method can be applied in an active or a passive manner. To do it in an active manner Alice choses randomly the intensity μ of the source for each pulse, using for instance a WCS source. In this case, each μ is related to a different Poisson distribution $P_\mu(n) = \mu^n e^{-\mu}/n!$. Alice can use the measured gain Q_μ and the QBER E_μ to estimate the single photon gain Y_1 and the error e_1, through the following equations:

$$Q_\mu = \sum_{n=0}^{\infty} P_\mu(n) Y_n \tag{2.9}$$

$$E_\mu Q_\mu = \sum_{n=0}^{\infty} P_\mu(n) e_n Y_n \tag{2.10}$$

This equations just mean that the gain is the sum of the probabilities of Alice emitting an n-photon signal times the conditioned probability of Bob measuring an n-photon event (yield). Since the gain can be experimentally measured, we can estimate the yield and the error using different intensities.

It can be shown that we only require a few intensities to obtain a bound for Y_1 and e_1. Hence, Alice only needs to generate states with three intensities. Typically the vacuum state ($\mu = 0$), the signal intensity $\mu = \xi$ and a decoy state with an intensity $\mu = \nu$ in between both of them ($0 < \nu < \xi$).

Now we can extract a lower bound for the for yield and an upper bound for the error (Derivation of this can be found in [Ma+05]):

$$Y_1 \geq Y_1^L = \frac{\xi}{\xi\nu - \nu^2} \left(Q_\nu e^\nu \frac{\nu^2}{\xi^2} - \frac{\xi^2 - \nu^2}{\xi^2} Y_0 \right) \tag{2.11}$$

$$e_1 \leq e_1^L = \frac{E_\nu Q_\nu e^\nu - e_0 Y_0}{Y_1^L \nu} \tag{2.12}$$

If these inequalities are fulfilled, Alice and Bob can bound the amount of detections from single-photons as well as their error rate to generate secure key.

2.6.2. Expression for the SKL in decoy-state method

Finally, conditioned on passing the checks in the error estimation and error-verification steps, a $\epsilon_{sec} - secret$ key of length ℓ can be extracted [Lim+14]:

$$\ell = \left\lfloor s_{X,0} + s_{X,1} - s_{X,1}h(e_p) - \text{leak}_{\text{EC}} - 6\log_2\frac{21}{\varepsilon_{\text{sec}}} - \log_2\frac{2}{\varepsilon_{\text{cor}}}\right\rfloor \tag{2.13}$$

Where:

- leak_{EC} : number of bits revealed at the error correcting step

$$\text{leak}_{\text{EC}} = k_{ec} + k_{ev} \tag{2.14}$$

- $s_{x,0}$: Number of vacuum events
- $s_{x,1}$: Number of single photon events
- e_p: Phase error rate
- $h(e)$: Binary Shannon entropy function.

2.7. QKD protocols beyond BB84

There are other QKD protocols that have been developed along the years. Among these Entanglement-Based (EB) protocols and Measure Device Independent (MDI) protocols are particularly interesting as they offer increased security, as they may be used to avoid the use of trusted nodes in satellite communication. However, this is not yet considered in the context of this work.

2.7.1. Entanglement-Based protocols

As an example of an EB protocol we describe BBM92 [BBM92]. It relies on EPR pairs and do not require a Trusted Node (TN). The protocol is the following:

1. An untrusted third-party Charles prepares an EPR pair and sends it to Alice and Bob.
2. Alice and Bob choose each a measurement basis randomly and store the result of the measurement.

3. Alice and Bob announce the measurement bases and discard the mismatching cases (sifting). Since both qubits are part of a Bell pair, the measures that were performed using the same basis will be correlated. In some cases Alice flips the bits to guarantee the correct correlation with Bob.
4. Alice and Bob share a fraction of the qubits through a classical channel and estimate de QBER. They can use the CSHS inequality [BBM92] to check if the qubits are actually entangled (that is why Charles does not need to be trusted).
5. inally, they use the decoy-state method to estimate the gain and QBER of the single-photon contributions.

In this case, the third party, Charles (the satellite in our case). Has no information about the result of the measurements, only about the selected bases.

2.7.2. Measure Device Independent protocols

The MDI protocols [LCQ12] in general work as follows:

1. Alice and Bob prepare each one of four BB84 states using phase-randomized WCSs and decoy signals. One qubit of each state is then sent to an untrusted third party, Charles.
2. Charles performs a Bell State Measurement (BSM) that generates entanglement between Alces and Bob (entanglement swaping), therefore creating a Bell state between them.
3. Charles announces the outcome of his claimed BSM using a classical public channel, and whether it is a succesful measurement or not.
4. Alice and Bob discard the events corresponding to unsuccessful measurements and announce their basis choices for sifting the successful events. Finally, they only the events using the same bases. Depending on Charles's measurement result, Alice flips some of the corresponding bits to guarantee the correct correlation with Bob. Finally, they use the decoy-state method to estimate the gain and QBER of the single-photon contributions.

In this case, the third party, Charles (the satellite in our case), has no information about the result of the measurements, only about the Bell state measurement result.

Also, note that in this case the detection is untrusted and the transmitters (Alice and Bob) need to trust their equipment, therefore, MDI-QKD is immune to all attacks targeting the detection, which, usually, is the weakest element against quantum hacking

3. Link Budget

Link budget refers to the compound of all the gains and losses in a telecommunication system, whether it is free-space communications or optical fiber connections. Link budget calculations include antenna and receiver gains, devices losses, atmospheric absorption, etc., and they are a way to assess system parameters, such as the required transmitter power and receiver sensitivity.

The transmittance or loss η is defined as the ratio between the transmitted power and the received power.

$$\eta = \frac{P_{\text{received}}}{P_{\text{emitted}}} \tag{3.1}$$

Generally, the resulting transmittance received at the Optical Ground Station (OGS) will be the product of the transmitter (Tx), receiver (Rx) and channel (Ch) losses:

$$\eta = \eta_{Tx}\eta_{Ch}\eta_{Rx} \tag{3.2}$$

Where:

- $\eta_{Tx} = \eta_{\text{internal loss}}$
- $\eta_{Ch} = \eta_{\text{scintillation}}\eta_{\text{atmospheric absorption}}\eta_{\text{pointing loss}}$
- $\eta_{Rx} = \eta_{\text{internal loss}}\eta_{\text{Rx}}\eta_{AO}$

Alternatively, it can be also expressed in decibels:

$$\eta[dB] = \eta_{Tx}[dB] + \eta_{Ch}[dB] + \eta_{Rx}[dB] \tag{3.3}$$

3.1. FSO device description

A Satellite to OGS system is a FSO setting. The transmitting and receiving devices are two telescopes which have a transmitter (e.g. a laser) or a detector (e.g a single photon detector) attached at the end. In our case, the transmitter uses a telescope with a D_{Tx} diameter, and the receiver consists of a two-axes mounted Cassegrain telescope featuring a primary mirror with a $D_{Rx,ext}$ diameter and a secondary mirror with a $D_{Rx,int}$ diameter. The values used in our simulations are in Table 3.1

	D_{Tx}	$D_{Rx,ext}$	$D_{Rx,int}$
Telescope	85mm	85mm	20mm
OGS	...	0.80m	0.30m

Table 3.1.: Size of the telescopes used in the simulations.

The devices operate at a specific wavelength λ, typically 1550nm (C-Band) or 850nm (Si-Band) where technological solutions for laser sources and detectors are readily available and where, furthermore, the atmosphere is sufficiently transparent.

It is also important to note that our simulations we always consider downlink connections between ground stations and satellites, i.e., the transmitter is located on the telescope while the receiver is on the OGS.

Repetition rate

The pulse repetition rate is the rate at which the photons are emitted. Current implementations of single photon sources are not competitive with laser-based sources, that is one of the reasons the latter are the chosen solution. In our simulations we fix the repetition rate at 1GHz.

Dead time

Due to the detection mechanism, the receiver also imposes a limit to the rate in which signals can be received because of dead time. Dead time is the time that a device needs for recovering from a detection and also avoid false positives. It depends on the specific solution: SNSPD, InGaAs SPAD, silicon SPAD and so on [Ors+25]. In our simulations we use values between 22ns and 100ns

3.2. End-to-end efficiency

3.2.1. Diffraction limited efficiency

Due to beam divergence, the beam spot area grows quadratically with the link distance L and thus, for a fixed size of the receiver telescope, the collected power decreases quadratically in L, $\eta = O\left(L^{-2}\right)$. Moreover, in the far-field regime η is upper-bounded by diffraction as [AP05]:

$$\eta \leq \eta_{coll}^{max} = G_{Tx} G_{Rx} \eta_{\text{free space}} \tag{3.4}$$

Where

- G_{Tx} is the transmitter gain, which is a performance parameter of the antenna. It depend on the area of the transmitter (A_{Rx}) and the wavelength :

$$G_{Tx} \leq \frac{4\pi A_{Tx}}{\lambda^2} \tag{3.5}$$

- G_{Rx} is the receiver gain. It depends on the receiver area (A_{Rx}) and the wavelength:

$$G_{Rx} \leq \frac{4\pi A_{Rx}}{\lambda^2} \tag{3.6}$$

- $\eta_{\text{free space}}$ is the loss due to the link distance L

$$\eta_{freespace} = \left(\frac{\lambda}{4\pi L}\right)^2 \tag{3.7}$$

In the end the diffraction limit is:

$$\eta_{coll}^{max} = \frac{A_{Tx}A_{Rx}}{L^2\lambda^2} = O\left(L^{-2}\right) \tag{3.8}$$

3.2.2. Collection efficiency for Gaussian beam and Cassegrain receiver telescope

We model the beam as a Gaussian beam, which is a sufficiently good approximation for most applications and also closely matches the shape produced by real optical terminals.

A Gaussian beam with beam waist at transmitter plane w_0 and peak intensity I_0 has the intensity value at distance z to receiver and radius r from the central axis:

$$I(r,z) = I_0 \left(\frac{w_0}{w(z)}\right)^2 \exp\left(-\frac{2r^2}{w(z)^2}\right) \tag{3.9}$$

with the beam waist $w(z)$ at distance z from transmitter:

$$w(z) = w_0\sqrt{1 + \left(\frac{z}{z_R}\right)^2}, \quad z_R = \frac{\pi w_0^2}{\lambda} \tag{3.10}$$

The amount of power received at the detector will be the result of the integral of the intensity at $z = L$, over the annular aperture ring between $R_{Rx,int}$ and $R_{Rx,ext}$. Therefore:

$$P(R_{\text{Rx,int}}, R_{\text{Rx,ext}}, L) = \int_0^{2\pi} \int_{R_{\text{Rx,int}}}^{R_{\text{Rx,ext}}} I(r,L)\, r\, dr\, d\theta =$$

$$= \pi I_0 w_0^2 \left[\exp\left(-\frac{2R^2_{\text{Rx,int}}}{w(L)^2} \right) - \exp\left(-\frac{2R^2_{\text{Rx,ext}}}{w(L)^2} \right) \right] \tag{3.11}$$

The total power is $P_{tot} = P(R_{\text{Rx,int}} \to 0, R_{\text{Rx,ext}} \to \infty) = \pi I_0 w_0^2$. So the fraction of the total transmitted power that reaches the detector is [AP05]:

$$\eta_{\text{Rx}} = \exp\left(-\frac{2R^2_{\text{Rx,int}}}{w(L)^2} \right) - \exp\left(-\frac{2R^2_{\text{Rx,ext}}}{w(L)^2} \right) \tag{3.12}$$

Where $w(L)$ is the beam waist at the receiver.

3.2.3. Device transmission loss

Devices also present internal losses that are not conservative and affect the final power. Transmitter and receiver optical systems typically introduce internal losses due to absorption. In our simulations we use values between 0.7 and 0.8

3.2.4. Atmospheric loss

The equation for the attenuation due to atmospheric absorption depends on the absorption coefficient $A(\lambda)$ for a specific wavelength λ and can be modeled as [AP05]:

$$\eta_A = 10^{-\int A(\lambda,h) L(h) dh} \tag{3.13}$$

where L is the link distance and h the height.

The absorption coefficient also depends on the altitude and the relevant effects to it are light absorption and scattering.

3.2.5. Refractive index structure

The refractive index structure, C_n^2, is a statistical measure of the strength of optical turbulence in the atmosphere. It quantifies how the refractive index of air fluctuates due to temperature, pressure, and humidity variations, which are factors that affect the propagation of electromagnetic waves. It is defined with respect to the refractive index n [AP05]:

$$C_n^2 \cdot r^{2/3} = \langle [n(\mathbf{x} + \mathbf{r}) - n(\mathbf{x})]^2 \rangle \tag{3.14}$$

A	HA	B	HB	C	HC	D	HD	d
4.5e-15	100	9e-17	1500	2.00e-3	1000	0	1	1

Table 3.2.: Parameters of Hufnagel-Valley 10/10 [AP05].

Where x is a point in the atmosphere and r a distance between the two points. The units of C_n^2 are $m^{-2/3}$.

To model, this profile, Hufnagel-Valley models are employed, where the parameters are fitted according to the function 3.15 [AP05]

$$\begin{aligned} C_n^2(h) = A \cdot \exp\left(-\frac{h - h_{\text{ground}}}{H_A}\right) \cdot F + B \cdot \exp\left(-\frac{h}{H_B}\right) + \\ + C \cdot \exp\left(-\frac{h}{H_C}\right) \cdot \left(\frac{h}{10^5}\right)^{10} + D \cdot \exp\left(-\frac{(h - H_D)^2}{2d^2}\right) \end{aligned} \tag{3.15}$$

In particular, the model HV 10/10 has the parameters in Table 3.2.

3.2.6. Scintillation

It is also important to take account for the scintillation, which is the focusing and defocusing of optical intensity within the beam due to atmospheric turbulence. According to [AP05] the scintillation follows a log- normal distribution. With a variance which is equal to the power index and an average value of the free-space transmittance.

A log-normal distribution is a distribution whose logarithm is normally distributed. To generate a log-normal distribution, take a normally distributed function and apply an exponential operation to it.

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right) \tag{3.16}$$

Where μ and σ^2 are the mean and variance of the generating normal distribution. The relationship with the mean (μ_X) and variance (σ_X^2) of the resulting log-normal distribution are:

$$\mu = \ln \frac{\mu_X^2}{\sqrt{\mu_X^2 + \sigma_X^2}} \tag{3.17}$$

$$\ln\left(1 + \frac{\sigma_X^2}{\mu_X^2}\right) \tag{3.18}$$

Where the mean value is the collection efficiency $\mu_X = \eta_{Rx}$ and the variance is the power scintillation index $\sigma^2 = \sigma_I^2$

$$\sigma_{\mathrm{I}}^2 = \frac{\langle P_{\mathrm{Rx}}^2 \rangle - \langle P_{\mathrm{Rx}} \rangle^2}{\langle P_{\mathrm{Rx}} \rangle^2} \tag{3.19}$$

This quantity is also called Power Scintillation Index (PSI) and is a quantitative metric used to measure the intensity of signal fluctuations (scintillation) caused by atmospheric turbulence on laser light propagation.

Rytov index

The Rytov index or variance is a dimensionless quantity used to quantify the strength of optical or radio wave scintillation as the wave propagates through a turbulent medium. It is usually denoted as σ_R^2, describes the variance of the logarithm of wave amplitude under weak fluctuation (scintillation) conditions [AP05].

$$\sigma_R^2 = \sigma_I^2 = 2.25\, k^{7/6} \int_0^L C_n^2(z)\, (L-z)^{5/6}\, dz \tag{3.20}$$

k is the wave number, L is the link distance, and z is the distance to the transmitter.

Power scintillation with aperture averaging

The PSI is only defined in a single point but the power scintillations must be integrated over the aperture on the receiver telescope. This results in an averaging effect which reduces the optical power fluctuations.

There are different schemes to model the aperture averaged scintillation. In the weak turbulence regime the following equation is usually employed [AP05]:

$$\sigma_{\mathrm{P,weak}}^2 = 8.7\, k^{7/6}\, \Re \int_0^L C_n^2(z) \left[\left(\frac{kD_{\mathrm{rx}}^2}{16} + iz \right)^{5/6} - \left(\frac{kD_{\mathrm{rx}}^2}{16} \right)^{5/6} \right] dz \tag{3.21}$$

In our work we use another effective equation that works also in the strong turbulence. We use the Rytov index and effective path length to account for slant path geometry in strong fluctuation condition, resulting in the following equation [Yur18]:

$$\sigma_{\mathrm{P,strong}}^2 =$$

$$= \exp\left[\frac{0.49\sigma_R^2}{\left(1 + 0.65 d_{\mathrm{AA}}^2 + 1.11\sigma_R^{12/5}\right)^{7/6}} + \frac{0.51\sigma_R^2 \left(1 + 0.69\sigma_R^{12/5}\right)^{-5/6}}{1 + 0.9 d_{\mathrm{AA}}^2 + 0.62 d_{\mathrm{AA}}^2 \sigma_R^{12/5}} \right] - 1 \tag{3.22}$$

Where:

$$d_{\mathrm{AA}} = \sqrt{\frac{kD_{\mathrm{rx}}^2}{4L_{\mathrm{eff}}}} \tag{3.23}$$

$$L_{\mathrm{eff}} = \left(\frac{18 \int_0^{L_{\mathrm{atm}}} C_n^2(z)\, z^2\, dz}{11 \int_0^{L_{\mathrm{atm}}} C_n^2(z)\, z^{5/6}\, dz} \right)^{6/7} \tag{3.24}$$

3.2.7. Effect of wavefront perturbations and adaptive optics correction

The power received at the telescope has to be couple to either to a free-space detector or to a single mode fiber. In the first case, the coupling efficiency can be arbitrarily high in principle. In the second case, the coupling efficiency depends on how well the adaptive optics system can correct wavefront perturbations.

The instantaneous wavefront aberration $\Psi(r,t)$ introduced by a turbulent atmosphere at a point $\vec{r}$ on the receiver aperture can be decomposed in a superposition of Zernike polynomials [Scr+22]

The Zernike coefficient variances $\langle b_n^{m2} \rangle$ represent how relevant an aberration order is. It depends on the ratio of the receiver aperture D_{Rx} to the Fried parameter r_0 with a modal term scaling with the radial order n:

$$\left\langle b_n^{m^2} \right\rangle = \left(\frac{D_{\mathrm{Rx}}}{r_0} \right)^{\frac{5}{3}} \frac{n+1}{\pi} \frac{\Gamma\left(n - \frac{5}{6}\right) \Gamma\left(\frac{23}{6}\right) \Gamma\left(\frac{11}{6}\right) \sin\left(\frac{5}{6}\pi\right)}{\Gamma\left(n + \frac{23}{6}\right)}. \tag{3.25}$$

The instantaneous coupling efficiency is:

$$\eta_{\mathrm{AO}}(t) = \exp\left[- \sum_{n,m} b_n^m(t)^2 \right]. \tag{3.26}$$

The Zernike coefficients are independent and Gaussian-distributed random variables whose mean is 0 and its variance is given by . Hence, he average coupling efficiency can be derived:

$$\langle \eta_{\mathrm{AO}} \rangle = \prod_{n,m} \frac{1}{\sqrt{1 + 2\left\langle b_n^{m^2} \right\rangle}}. \tag{3.27}$$

To compute the average coupling efficiency in the presence of a partial Adaptive Optics (AO) compensation, we only need to suppress the lower orders which are corrected in eq. 3.28, i.e.:

$$\langle \eta_{\mathrm{AO}} \rangle = \prod_{n>n_{max},m} \frac{1}{\sqrt{1+2\left\langle b_n^{m2} \right\rangle}}. \tag{3.28}$$

3.2.8. Pointing errors

Pointing bias and pointing jitter also contribute to the losses

- Pointing bias: In case of a fixed off-set angle ν_{bias} (fixed bias with no jitter), the intensity is given by a Gaussian function where θ_0 is the beam divergence angle:

$$I(\nu_{bias}) = \exp\left(-2\frac{\nu_{bias}^2}{\theta_0^2}\right) \tag{3.29}$$

- Pointing jitter: In absence of bias pointing angle, the pointing error is distributed as a Rayleigh distribution (jitter) where σ_{jit} is the variance of the jitter:

$$\theta \sim \mathrm{Rayleigh}(\sigma_{\mathrm{jit}}) \Leftrightarrow p_{\mathrm{jit}} = \frac{\theta}{\sigma_{\mathrm{jit}}^2} e^{-\theta/(2\sigma_{\mathrm{jit}}^2)} \tag{3.30}$$

  from wich one can obtain the probability distribution of I, for the given beam divergence and jitter model:

$$p_I(I) = p_{\mathrm{jit}}[\theta(I)] \left| \frac{d\theta}{dI}(I) \right| = \frac{\theta_0^2}{4\sigma_{\mathrm{jit}}^2} I^{\theta_0^2/4\sigma_{\mathrm{jit}}^2 - 1} \tag{3.31}$$

For the case where we have both pointing jitter and pointing bias, the pointing angle results to be distributed according to a Rice distribution.

$$\theta \sim Rice(\nu_{\mathrm{bias}}, \sigma_{\mathrm{jit}}) \Leftrightarrow p_{\mathrm{jit+bias}}(\theta) = \frac{\theta}{\sigma_{\mathrm{jit}}^2} exp\left(\frac{-\sigma^2 - \nu_{\mathrm{bias}}^2}{2\sigma_{\mathrm{jit}}^2}\right) I_0\left(\frac{\theta\nu_{\mathrm{bias}}}{\sigma_{\mathrm{jit}}^2}\right) \tag{3.32}$$

Where I_0 is the modified Bessel function of the first kind with order zero.For a Gaussian beam, we have:

$$I(\theta) = e^{-2(\theta/\theta_0)^2} \tag{3.33}$$

$$\theta(I) = \theta_0\sqrt{0.5\ln I^{-1}} \tag{3.34}$$

$$\frac{d\theta}{dI}(I) = \frac{-\theta_0}{4I\sqrt{0.5\ln I^{-1}}} \tag{3.35}$$

We can now compute the probability distribution of I, for the given beam divergence and jitter model:

$$p(I) = p_{jit}[\theta(I)] \left|\frac{d\theta}{dI}(I)\right| = \frac{\theta_0^2}{4\sigma_{\text{jit}}^2} I^{\theta_0^2/4\sigma_{\text{jit}}^2-1} e^{-\gamma} I_0 \left(\frac{\theta_0\sqrt{0.5\ln I^{-1}}\nu_{\text{bias}}}{\sigma_{\text{jit}}^2}\right) \tag{3.36}$$

3.3. Noise model

3.3.1. Detector dark count

We model dark counts as independent events, therefore we consider them as a Poisson distribution (eq. 3.37). The Poisson distribution describes the probability of a given number of events occurring in a fixed interval of time or space if these events occur with a known constant mean rate and independently of the time since the last event.

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{3.37}$$

Where:

- λ is the mean number of occurrences
- k is the number of occurrences for which the probability is computed

The probability of no no dark counts happens for $k = 0$, and the mean number of dark counts μ_{dc} is a parameter that depends on the detector:

$$P_{dc} = 1 - P(k = 0) = 1 - e^{-\mu_{dc}} \tag{3.38}$$

The fraction of dark counts detected on average is the rate between the dark count rate, which is a parameter of the detector, and the pulse rate $\mu_{dc} = f_{dc}/f_{pulse}$

3.3.2. Background light

We use the same Poisson model for the background light. We take a fixed value for the background light during night time μ_{bgl}.

$$P_{bgl} = 1 - P(k = 0) = 1 - e^{-\mu_{bgl}/2} \tag{3.39}$$

The factor 1/2 in the exponent comes from the fact that we model a QKD receiver with 2 detectors, employed to detect orthogonal states of light, and we assume that the background light is unpolarized.

The fraction of background light counts detected on average is the rate between the dark count rate, which is a parameter of the detector, and the pulse rate $\mu_{bgl} = f_{bgl}/f_{pulse}$

4. Orbits

4.1. Keplerian orbits

Usually satellite orbits are defined as an ideal Keplerian orbit in a two-body system. They can be uniquely characterized by a set of orbital elements [Bat+20], which can be described in different mathematical ways.

Nevertheless, in reality, the orbit and its elements change over time due to the multipolar effects originated from the non-sphericity of the Earth, the gravitational perturbations by other solar system objects, the effects of general relativity, the atmospheric drag and so on [Bat+20].

4.1.1. Orbital elements

The orbital elements describe the position of a satellite with respect to the center of mass of the Earth and it consists of six parameters [Bat+20]. Five of them are required to define the orbit, and the sixth one is used to define the exact position of the body along it (see Fig. 4.1).

Two planes are important in the definition of these parameters:

1. Orbital plane: Plane that contains the orbital curve
2. Reference plane: Typically the equator of the central body

Also two reference points are needed:

1. Ascending node: point in which the orbit crosses the reference plane in the side where the object is moving north (the other side is the descending node). There are no nodes in equatorial orbits.
2. Vernal point: together with the central body it establishes a reference direction

Finally, the orbital elements are:

1. Semi-major axis (a): distance from the center of the elliptical orbits and the periapsis point. This is equal to the length of the major axis of the ellipse divided by two.

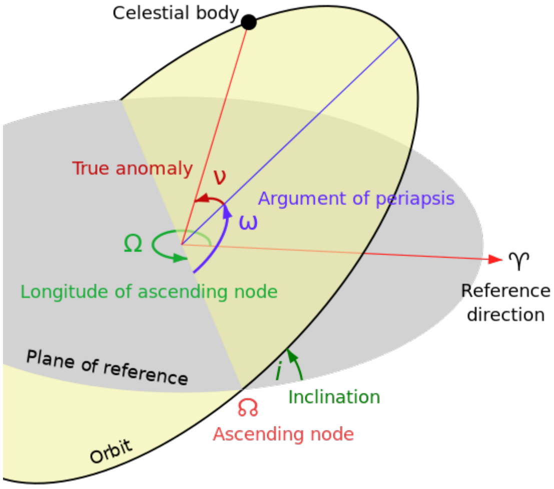


Figure 4.1.: Orbital elements [Las08]

2. Eccentricity (e): it defines how much the ellipse deviates from a perfect a circle. An eccentricity of zero defines a circle, values less than 1 describe an ellipse, values greater than 1 describe a hyperbolic trajectory, and a value of exactly 1 describes a parabola.

3. Inclination (i): it refers to the inclination of the orbital plane with respect to the reference plane measured at the ascending node. Inclinations near 0° indicate equatorial orbits, and inclinations near 90° indicate polar orbits. Values from 90 to 180° are used to denote retrograde orbits.

4. Longitude of ascending node (Ω) or Right Ascension of the Ascending Node (RAAN): angle from the ascending to the reference direction. It is measured in the reference plane. However, it is undefined for equatorial (coplanar) orbits, but is often set to zero instead by convention.

5. Argument of peri-axis (ω): orientation of the ellipse in the orbital plane. It is the angle between the ascending node and the periapsis of the ellipse (the closest point the satellite body comes to the primary body around which it orbits). It is set to zero in circular orbits by convention.

6. True anomaly (ν): Angle between the argument of periapsis and the position of the orbiting body. It is measured anti-clock wise (from a north perspective) and it is has a value between 0° and 360°.

4.1.2. Precession

However, the orbits described by these elements are not static but they change over time due to different factors. The most important change of the orbit of an Earth satellite over time is Nodal Precession, which is defined as "the precession of the orbital plane of a satellite around the rotational axis of an astronomical body such as Earth". The origin of this rotation is the non-spherical nature of a rotating body (since the Earth is not a sphere but an oblate spheroid due to the equatorial bulge), which creates a non-uniform gravitational field.

Due to this deformation on the shape of the central body, the gravitational force acting on a satellite is not directed toward the center of it, but rather is offset toward its equator. Whichever hemisphere of the central body the satellite lies over, it is pulled slightly toward the equator of the central body. This results on a torque over the satellite. However, this torque does not reduce the inclination but it causes a torque-induced gyroscopic precession, which causes the orbital nodes to drift with time.

The formula for the precession rate is [Bro02]:

$$\omega_p = -\frac{3}{2}\left(\frac{R_E^2}{a^2(1-e^2)^2}\right) J_2 \omega \cos i \tag{4.1}$$

where i, a and e are the orbital parameters and furthermore:

- $R_E = 6.378137 \times 10^6 m$ is the Earth's equatorial radius.
- $J_2 = 1.08262668 \times 10^{-3}$ is the Earth's second zonal harmonic coefficient related to the oblatteness of the Earth.
- ω is the mean motion, given by:

  $$\omega = \sqrt{\frac{GM}{a^3}} \tag{4.2}$$

  where $GM = 3.986004418 \cdot 10^{14} m^3 s^{-2}$ is the standard gravitational parameter of the Earth.

4.1.3. Sun Synchronous Orbits

The precession phenomenon can be leveraged to create a Sun Synchronous Orbit (SSO). This is a type of orbit in which the precession is tailored so that it completes a full revolution for each Earth revolution around the Sun. This means that the relative position of the orbit with respect to the Sun is constant and therefore, it will pass over each point on Earth at the same local time ever time [Boa04].

The inclination needed to achieve a SSO for an Earth orbiting satellite with a circular orbit is:

$$\cos i \approx -\left(\frac{a}{12352km}\right)^{7/2} \tag{4.3}$$

For a LEO such as $a = 500km + R_E$, the SSO angle is $i = 97.4°$, so it is close to a polar orbit and retrograde.

4.2. Satellite constellation

To reach a global coverage that can provide connection access in real-time to many grounds stations around the world, a single satellite is not enough. In fact, many satellites working together as a system are required to provide this kind of service, that is, a satellite constellation.

4.2.1. Walker constellation

In classical communications, Walker constellations [Wal84] are usually the preferred choice (for instance, Starlink uses this type of constellation). All of the orbits in this kind of setting have similar orbital parameters, namely, they share the same inclination so the precession effect is the same for all the objects.

Walker constellations are determined by a set of parameters:

- i: inclination of the orbital planes
- t: total number of satellites
- p: number of equally spaced orbital planes
- f: relative spacing between satellites in adjacent planes. The change in true anomaly (°) for equivalent satellites in neighboring planes is $f \times \frac{360}{t}$

The notation is
$i : t/p/f$

4.2.2. Two-heights, single-orbital plane constellation

As we discussed in 1.3, our new proposal for a QKD satellite constellation consists in using a single orbital plane with two different orbital heights as it is shown in Fig. 4.2. This configuration is expected to optimize the inter-satellite communications in QKD.

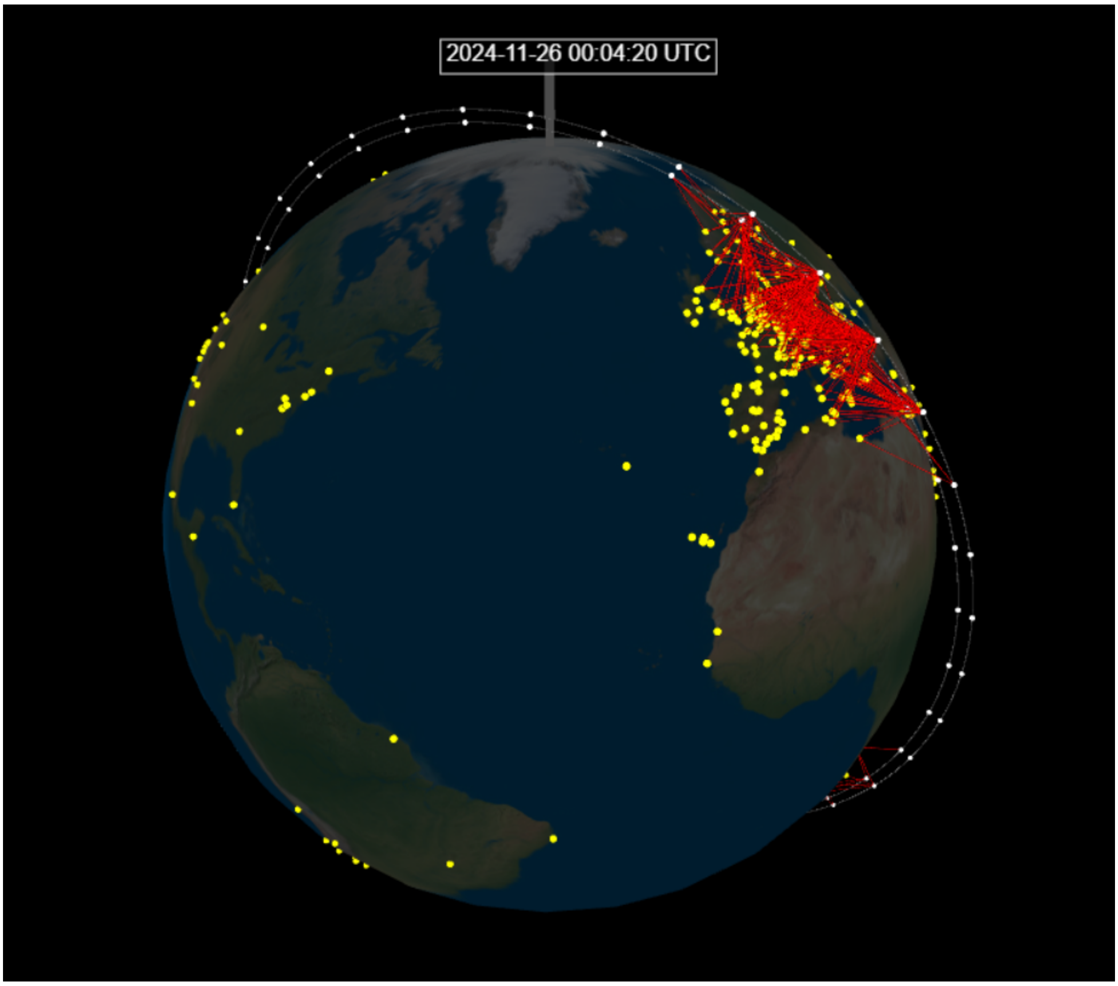


Figure 4.2.: Visualization of a system with two satellites at different heights and 367 ground stations.

Synodic period

To compute how often two satellites will meet we can use the synodic period, which is the time that takes for an object to appear with respect to a second object in the same position in relation to a third object. In our case this translates two how long it takes for a satellite in the first orbit to be aligned again with a satellite of the second orbit with respect to the Earth (time between conjunctions).

According to Kepler's Third Law, the orbital period of an object with a circular orbit of radius r orbiting a massive object of mass M is:

$$T = 2\pi\sqrt{\frac{r^3}{GM}} \tag{4.4}$$

If we have two bodies with orbital periods T_1 and T_2 with respect to the same massive object, so that $T_1 < T_2$ the synodic period is [Kar+16]:

$$\frac{1}{T_{syn}} = \frac{1}{T_1} - \frac{1}{T_2} \tag{4.5}$$

Therefore, the number of passes between two satellites during a time t is:

$$N_{\text{passes}}(t) = \left\lfloor \frac{t}{T_{syn}} \right\rfloor \tag{4.6}$$

4.3. Orbit propagation

To compute the orbital propagation of the satellites around the Earth, simplified perturbations models are the preferred choice [HR80]. These models are used to calculate orbital state vectors of satellites with respect ton an Earth-centered coordinate system. There are five different models: SGP, SGP4, SDP4, SGP8 and SDP8.

The models include several relevant perturbation effects that can affect the satellite trajectory and differ from the Keplerian orbit such as: precession, drag (caused by the Earth's atmosphere), radiation and gravitational influence of other celestial bodies such as the Sun or the Moon.

There are two main kinds of models:

- Simplified General Pertutbations (SGP): near Earth objects, i.e. period under 225 minutes
- Simplified Deep Space Perturbations (SDP): objects with an orbital period greater than 225 minutes

The SGP4 model (the most commonly used) error is around 1 km at epoch and grows between 1 and 3 km per day. This data is updated frequently in NASA and NORAD sources due to this error.

4.3.1. Two-element line set

Simplified perturbations models are commonly used in combination with the Two-Line Element (TLE) set. This is a standard way provided by NASA and NORAD to provide the orbital elements of a Satellite orbiting the Earth at a given time (epoch).

One TLE [NAS00] uniquely determines a Earth-centered orbit. The format was originally designed for punch cards and that is why it possess a very strict structure, with all the data contained in 69 columns and does not contain a trailing character.

Line 1

- Column 1: Line number (always 1)
- Columns 3–7: Satellite catalog number
- Column 8: Classification
  - U: Unclassified
  - C: Classified
  - S: Secret
- Columns 10–17: International Designator
  - Columns 10–11: Last two digits of launch year
  - Columns 12–14: Launch number of the year
  - Columns 15–17: Piece of the launch
- Columns 19–20: Epoch year (last two digits)
- Columns 21–32: Epoch (day of the year and fractional portion of the day)
- Columns 34–43: First derivative of mean motion (ballistic coefficient)
- Columns 45–52: Second derivative of mean motion (decimal point assumed)
- Columns 54–61: B*, the drag term (decimal point assumed)
- Column 63: Ephemeris type (always zero; used only in undistributed TLEs)
- Columns 65–68: Element set number (incremented when a new TLE is generated for the same object)
- Column 69: Checksum (modulo 10)

Line 2

- Column 1: Line number (always 2)
- Columns 3–7: Satellite catalogue number
- Columns 9–16: Inclination (degrees)
- Columns 18–25: Right ascension of the ascending node (degrees)
- Columns 27–33: Eccentricity (decimal point assumed; e.g., 0006703 means 0.0006703)
- Columns 35–42: Argument of perigee (degrees)
- Columns 44–51: Mean anomaly (degrees)
- Columns 53–63: Mean motion (revolutions per day)
- Columns 64–68: Revolution number at epoch (total orbits completed)
- Column 69: Checksum (modulo 10)

An example of TLE (International Space Station (ISS)) is:

```
1 25544U 98067A   08264.51782528 -.00002182  00000-0 -11606-4 0  2927
2 25544  51.6416 247.4627 0006703 130.5360 325.0288 15.72125391563537
```

5. Satellite QKD networks

QKD networks are different from classical communication networks. The main difference is that they are not employed for real time communication but rather for key exchange. This happens before the encrypted communication.

5.1. QKD with satellites

The main obstacle with QKD networks currently is the inexistence of quantum repeaters. That implies the need for TN architectures. A TN is essentially a node in which the information is converted to classical information, and therefore, it needs to be trustworthy of not being controlled or hacked by any eavesdropper.

However, for security reasons, it is favorable to reduce the number of TN to a minimum and also make them as hardened against hacking as possible. In satellite QKD networks the trusted nodes are the satellites, which posses the advantage that, being in space, they are difficult to be physically accessed and this makes hacking them more challenging. Moreover, a single satellite may be sufficient for the communication between two OGS that can be located at any point in Earth, as long as the satellite can establish a link with each of the two OGS at a certain point in time.

The process is very simple. Given two ground stations A and B and a satellite:

1. The satellite and A exchange a random secret key_A at a certain time.

2. The satellite and B exchange a random secret key_B at a later time.

3. When, in a third moment, A and B request to have a secret key, the satellite performs a one time padding operation and broadcasts the result. That is, the satellite computes:

$$\text{key}_{\text{sat}} = \text{key}_A \oplus \text{key}_B \tag{5.1}$$

   where $\oplus$ means a xor operation (binary sum). Note that key_{sat} is completely random to anyone who does not know key_A or key_B, hence, it is completely secure to share this key through a public classical channel.

4. B performs
$$\text{key}_A = \text{key}_B \oplus \text{key}_{\text{sat}} \tag{5.2}$$
and now A and B share a common key that is secure against any eavesdropper (unless a potential hacking of the TN).

Now B and A share the same key. Here we used the property of xor that states $x \oplus x = 0$. One time padding system is simple and completely secret. The only disadvantage is that it requires a key that is the same length as the message to be encrypted.

Another advantage is that it can be expanded to a system of more satellites. Imagine we have a second satellite. A communicates key_A with satellite 1 and B communicates key_B with satellite 2. Both satellites communicate $\text{key}_{\text{inter-sat}}$ to each other.

1. Satellite 1 performs $\text{key}_{\text{sat1}} = \text{key}_A \oplus \text{key}_{\text{inter-sat}}$ and sends it to satellite 2
2. Satellite 2 performs $\text{key}_{\text{sat2}} = \text{key}_{\text{inter-sat}} \oplus \text{key}_{\text{sat1}} \oplus \text{key}_B = \text{key}_A \oplus \text{key}_B$ and sends it to B
3. B performs $\text{key}_{\text{sat2}} \oplus \text{key}_B = \text{key}_A$

5.1.1. Untrusted node networks

Although BB84 is still the only practical protocol to realize QKD for satellites, there are other protocols in a more experimental and theoretical state that could theoretically allow satellite networks without the need of a trusted node by using entanglement. The main problem with this is the lack of quantum memories, therefore, to use entanglement the satellite should be able to communicate with two ground stations at the same time, which increases the technical complexity and limits the communication distance range. One example of EB protocols is BBM92. Also, alternatively, MDI may be used, in which a third node is used to perform entangling measures on incoming state.

5.2. End-user requirement modeling

Once a satellite constellation has been defined, we need to check whether it can fulfill the requirements of a realistic network. We use a Barabasi-Albert (BA) network for this purpose i.e., we generate the user pairs that want to communicate according to a BA model.

5.2.1. Barabasi-Albert networks

Many real-world networks have been observed to have a property called preferential attachment. This means that the more connected is a node, the more likely it is to get new connections [AB02]. This phenomena can be observed for instance in the world wide web, where big hubs, such as Google, are more likely to be linked to new sites than small, unknown websites. We assume that this will be the same in the case of the network of QKD users. Not all of the users will have the need to connect to every other user: in a realistic setting, there are some important nodes, like hubs, which will require connections to many other users, but there will also exist less popular nodes that may only communicate with very few others.

The BA model [AB02] integrates preferential attachment together with growth (number of nodes in the network increases over time) to generate scale-free networks. To say that a network is scale-free means that the fraction $P(k)$ of nodes in the network having k connections to other nodes goes for large values of k as:

$$P(k) \sim k^{-\gamma} \tag{5.3}$$

The BA model only has one parameter $m \in \mathbb{N}$ which is the number of connections that each new node generates at the moment when is created.

The initial number of vertices in the network is $m_0 \geq m$. Then add a new node which has m random networks selected among the already existing vertices. The probability of a vertex to be selected is proportional to its degree:

$$p_i = \frac{k_i}{\sum_j^{i-1} k_j} \tag{5.4}$$

Where k_i is the degree of the node. Therefore, heavily connected nodes will tend to accumulate new links more quickly.

The original papers did not specify how to handle cases where the same existing node is chosen multiple times, in our case we just select another random node.

The resulting degree distribution of a BA network is

$$P(k) \sim k^{-3} \tag{5.5}$$

5.2.2. Link request timing

Once all the connections between the users, meaning the two end points who issue a request for a QKD key are initialized (these connections are an abstraction of the network and are represented by the edges of the BA graph, not to be confused with the physical QKD links), it is necessary to decide when to activate them. The selected method is a constant probability of getting activated at every time. Therefore, in a unit time interval the probability of a random link between two ground stations to request a key exchange is p. All of the links have the same probability of being active. Therefore, the nodes with more connections will require more key than the nodes with fewer connections.

5.2.3. Prioritization

When there is a big number of ground stations and satellites, there is another problem that needs to be addressed: prioritization. That is, if two connections between two different ground stations and a satellite (or vice versa) interfere with one another, there needs to be a selection criteria. In this section we describe the criteria that is used in the code.

First a weight v needs to be given to each of the potential connections:

$$v = \begin{cases} k_i g/s & \text{if } s > 0 \\ \infty & \text{if } s = 0 \end{cases} \tag{5.6}$$

Where

- k_i is the node degree (number of vertices in the network)
- g is the potential key gain (how much key is expected to be generated during the exchange
- s current cumulated exchanged key between the station and the satellite

Then check which are the conflicting links. Two links are conflicting if:

- they share at least one satellite or ground station
- they overlap in time, i.e. $t_{start,1} < t_{end,2}$ and $t_{end,1} > t_{start,2}$

The algorithm is:

1. Find all the conflicting links with the current one
2. Compute v for each link
3. Select the link with the biggest v value

6. Simulator

6.1. Code structure

The code is structured in five parts:

1. Orbit propagation: it computes all the passes between the satellites and the ground stations as well as the passes between the satellites.
2. Link Budget: computes the total transmittivity for each point in time for each pass.
3. QKD protocol: It computes the statistics of the protocol given the trasmittivity probability distribution and returns the total key length.
4. Visualization: Tool that plots an animation of the satellite orbiting around the earth that allows for the visualization of the passes
5. Network Simulation: We simulate the users requirements in a real time simulation

6.1.1. QKD protocol

The last part to simulate is the QKD protocol.

Physical Setup

The QKD specific parameters are contained in a different class for each element:

- Transmitter class `Tx_BB84()`. It contains the following attributes:
  - wavelength
  - repetition rate
  - maximum attenuation
  - polarization misalignment
  - temporal misalignment

- Channel class Ch_BB84() . It contains the following attributes:
  - transmission
  - background rate
  - misalignment angle
- Receiver class Rx_BB84() . It contains the following attributes:
  - gating time
  - dead time
  - dark rate
  - efficiency

Modulation parameters: Free parameters to optimize

The modulation parameters are the probabilities of using each encoding basis used for the encoding and the probabilities and intensities of the decoy states. These are stored in the BB84_DS_Modulation_Params class.

Protocol

All of the main methods for the protocol are contained in the class BB84_DS_Protocol

6.1.2. Link Budget

To determine how much key length can be exchanged, we need to compute how much losses/transmitivity exist in our link during the pass.

There are three main elements that contribute to the losses:

- Transmitter: Internal losses of the transmitter due to the optical system
- Receiver: Internal losses of the transmitter due to the optical system and to the fibre-coupling (if exists)
- Channel: Losses due to many factors.
  - Laser diffraction: there exists a fundamental minimal diffraction limit when a beam of light spreads from the source. In our case, we are dealing with big distances (and limited telescope sizes) so the diameter that reaches the ground from the satellite (in Downlink setting) is way bigger than the diameter of our telescope. This is the main source of our main loss.

  - Atmospheric absorption. Even if it is small, part of the light is absorbed by the atmosphere.
  - Pointing bias and jitter. The process of tracking and pointing is very delicate. There exist very precise systems but this is still a source of signal losses
  - Scintillation. Due to the heterogeneousness of the atmosphere, the light suffers refraction when going from lower transmittance regions to higher ones or vice versa. After the wavefront hits the atmosphere it is not flat (or spherical any more) but it shows some deformities due to different refraction paths of the different points of the wavefront.

However we do not compute an exact η for every instant in time but rather we calculate the full Probability Density Function (PDF)

To compute the link budget we have three different types of classes: Transmitter, Channel, Receiver

Transmitter

The transmitter class contains all the parameters of the transmitter. There is a parent class that defines the two basic methods for every transmitter (Internal_Loss and Gain_Total) It also defines the basic attributes :

- Wavelength λ
- Internal Efficiency (Source of transmitter losses)

Different kinds of devices can be defined as child classes: fiber link, Gaussian Beam... For our specific study, we simulate a Gaussian beam, which posses the following attributes, besides the basic ones:

- beam waist:
- half divergence:
- curvature radius:
- pointing jitter:
- pointing bias:
- link margin:

And the methods:

- Methods compute beam parameters:
  - MinimumHalfDivergence : computes the minimum possible half divergence given by the diffraction limit.
  - HalfDivergence_from_Curvature : computes the minimum possible half divergence using the curvature value.
  - InitialCurvature_from_HalfDivergence : Computes the curvature value given the half divergence
- PointingLoss : It computes the pointing loss given by the pointing jitter and the pointing bias
- PointingLoss_PDF : : It computes the probability of a certain pointing loss given by the pointing jitter and the pointing bias
- Array_PointingLoss_PDF : : It computes the full PDF of the pointing loss given by the pointing jitter, the pointing bias, η_{min}, η_{max} and the number of points to compute
- GainFromHalfDivergence : computes the gain of the beam
- General methods:
  - Internal_Loss : computes the internal loss of the transmitter
  - Gain_Total : the total gain is the gain from the beam minus the internal and pointing losses (if considered)

Receiver

Following the same structure, the receiver class holds all the physical parameters of the receiver. There is a parent class that defines the two basic methods for every kind of receiver (Internal_Loss and Gain_Total) It also defines the basic attributes :

- Wavelength λ
- Internal Efficiency

Depending on the link (fibre link or telescope) different child classes can be defined. The Rx_Telescope class possesses the following attributes besides the basic ones:

- external radius:
- internal radius:

- focal ratio:
- fibre coupled (yes/no):
- detector diameter (if fibre coupled):
- mode field diameter (if fibre coupled):
- Zernike correction order (if fibre coupled):

And the methods:

- OccultationRatio : Ratio of primary to secondary mirror diameter
- Gain_from_Telescope : Gain from the telescope (without the fibre coupling)
- FibreCouplingEfficiency : Efficiency of the fibre coupling
- General methods:
  - Internal_Loss : computes the internal loss of the receiver
  - Gain_Total : the total gain is the gain from the telescope minus the internal losses (if considered)

Channel

The channel class holds the rest of the physical parameters. There is a parent class that defines the two basic methods for every channel (Loss_Total). It also defines the basic attributes :

- Wavelength λ
- Transmission efficency

Depending on the link (fiber link or free space) different child classes can be defined. The Ch_FreeSpace class has the following attributes:

- Visibility: it can be high, low or medium
- LinkGeometry: Holds all the necessary geometrical values and methods
- Base Atmospheric Model: The model that defines the atmospheric attenuation along the different atmospheric altitudes

And the methods:

- FreeSpaceLoss : Loss due to the distance that the beam must travel
- AtmosphericLoss : Gain from the atmospheric absorption
- FibreCouplingEfficiency : Efficiency of the fibre coupling
- General methods:
  - Internal_Loss : computes the internal loss of the receiver
  - Loss_Total : the total loss is the free space loss plus the atmospheric loss (if considered)

Scintillation

We also have a defined class for the Scintillation values. There are three kinds of Scintillation:

- Uplink
- Downlink
- Path

In our system we have downlink links (Satellite-OGS) and path links (Inter-Satellite). Although the former one is not really relevant because there is no meaningful atmospheric density.

The child class that defines the methods for the Downlink case is Scintillation_Downlink . The more relevant methods are:

1. PowerScintillationIndex : computes the power scintillation index, which includes the effect of aperture averaging when the receiver has an extended aperture. It can be computed according to different references: "AP", "Churnside" or "Yura". We use the result from Yura as the standard formula.
2. RytovVariance : computes the Rytov Variance σ_R^2. For weak turbulence conditions, it is equal the scintillation index.
3. EffectiveLinkLength : computes an effective link length to account for slant path in strong fluctuation atmospheric conditions.
4. convert_to_StrongScintillationPower : converts from weak scintillation case to strong scintillation.

5. FriedParameter : computes the fried parameter r_0.

6. IntensityScintillationIndex : computes the scintillation index σ_I^2

7. ApertureAveraging : computes the aperture averaging factor for the optical power scintillations when signal is received by an extended aperture.

Link budget

Finally, there is a class called LinkBudget that manages the other classes (transmitter, receiver, channel and scintillation) and takes care of all the losses that involve elements from two or more elements (transmitter, receiver and channel).

The main method here is Transmission_PDF because it computes the full PDF of the transmittance for each time instant. This is the algorithm:

1. Create an array of η values equally spaced in the logarithm scale (dB)

2. Compute the collection efficiency PDF(CollectionEfficiency_PDF) for each η value which is shaped as a log-normal function. The distribution mean is the average collection efficiency ($\mu_\eta = \eta_{R_x}$) and the variance is equal to the power scintillation index ($\sigma_\eta = \sigma_I$) (include reference!!!!!)

$$p_{R_X}(\eta;\mu,\sigma) = \frac{1}{\eta\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln\eta-\mu)^2}{2\sigma^2}\right), \quad x>0 \tag{6.1}$$

$$\sigma^2 = \ln\left(\frac{\sigma_\eta^2}{\mu_\eta^2}+1\right) \tag{6.2}$$

$$\mu = \ln(\mu_\eta) - \frac{\sigma^2}{2} \tag{6.3}$$

If there is no scintillation, such as in the inter-satellite case, the distribution is a Dirac delta function:

$$p_{R_X}(\eta) = \delta(\eta - \eta_{R_x}) \tag{6.4}$$

And the function returns a single float value η_{R_X}.

3. Renormalize the PDF function to correct numerical errors

4. If the pointing losses are also taken in to account, they are computed now with the method PointingLoss_PDF from the transmitter class. There are two sources of pointing loss:

- Pointing Bias
- Pointing Jitter

The combination of the two results in a Rice distribution (see section 3.2.8)

$$\beta = \frac{\theta_0^2}{4\sigma_{jit}^2}, \tag{6.5}$$

$$\gamma = \frac{\nu_{bias}^2}{2\sigma_{jit}^2} \tag{6.6}$$

$$p_{PL}(\eta) = \beta \cdot \eta^{\beta-1} \cdot e^{-\gamma} \cdot \eta_0 \left(2\sqrt{\beta\gamma \ln\left(\frac{1}{\eta}\right)}\right) \tag{6.7}$$

5. If Scintillation and pointing loss are both considered, we have to combine both probability distributions. We consider both random variables η_{Scint} and η_{PL} to be statistically independent. Therefore, the product of both variables can be seen as a single variable whose PDF is the convolution of the logarithm of the product (See Math Methods section)

6. Finally, we must combine the constant probabilities (delta Dirac functions). These are:

   - Internal transmitter and receiver losses: η_{int,T_x} and η_{int,R_x}. They are attributes of the transmitter and receiver class respectively
   - Atmospheric loss η_{atm}. It is computed with the AtmosphericLoss function of the Channel class
   - If the receiver is fiber coupled, the Coupling loss η_{CL} is also considered. It is computed thorough the Coupling_Loss method of the Scintillation class.

   Therefore the constant value of η is:

$$\eta_{const} = \eta_{int,T_x}\eta_{int,R_x}\eta_{atm}\eta_{CL} \tag{6.8}$$

   The total value η_{tot} is:

   - In case there is no pointing loss or scintillation, the resulting η is another Dirac delta whose value is the product of all of them: $\eta_{tot} = \eta_{R_x}\eta_{const}$
   - In case there is at least Scintillation or Pointing loss the total distribution is:

$$p(\eta_{tot}) = \frac{p(\eta \cdot \eta_{const})}{\eta_{const}} \tag{6.9}$$

7. Assume $p_{t_i}(\eta)$ constant during Δt and the total distribution of η of the pass is the sum of all the instant distribution. Therefore:

$$p_{pass}(\eta) = \sum_{i=0}^{N} p_{t_i}(\eta)\Delta t \tag{6.10}$$

Where N is the total number of time steps in the pass

6.1.3. Orbits

To find all the passes we use the Skyfield library.

"Skyfield computes positions for the stars, planets, and satellites in orbit around the Earth. Its results should agree with the positions generated by the United States Naval Observatory and their Astronomical Almanac to within 0.0005 arcseconds (half a "mas" or milliarcsecond)."

Skyfield uses the SGP4 model to compute the satellite objects and the World Geodesic System 1984 (WGS84) to compute the positions of the ground stations.

The procedure is the following:

1. A minimal altitude θ_{min} over which a pass is considered is set. Also a maximum minimal altitude θ_{min}^{max}, that the satellite must reach during the pass (two low/short passes are ignored). For our simulations we selected $\theta_{min} = 20$ and $\theta_{min}^{max} = 30$
2. Select a Satellite-OGS pair from the chosen configuration
3. Use the "find_events" from Skyfield to find all of the moments between t_{start} and t_{end} in which the satellite rises over the OGS θ_{min} altitude and the moments in which is sets under that latitude. The time frame between these two moments is considered a pass.

   Then, θ^{max} is computed for each of the passes. All the passes for which $\theta^{max} < \theta_{min}^{max}$ are discarded.
4. Now that all the passes have been found, an array of time for the pass with Δt separation is created. For each time in the array, the following values are computed
   - Distance of the satellite to the center of the earth in meters (Sat_Abs_Altitude)
   - Elevation angle of the satellite as seen from the OGS perspective in meters (OGS_Elevation_Angle)
   - Azimuth angle of the satellite as seen from the OGS perspective in radians(OGS_Azimuth_Angle)

  - Distance between the satellite and the OGS in meters
  - Whether the OGS is sunlit at the given time step (OGS_is_Sunlit)

5. Repeat the same process for all the possible Satellite-OGS combinations

6. Store the data in a dictionary and export it as a JSON file

For inter-satellite passes the procedure is similar:

1. Instead of an elevation angle, in inter-satellite passes we establish a minimum visibility distance d_{min}. Above this distance the satellites are considered to be too far apart to establish a link. For our simulations we selected $d_{min} = 100km$

2. Take a $Sat_A - Sat_B$ pair from the chosen configuration. In order to avoid repetitions, since all the satellites are numbered $num(Sat)$, the pairs are selected such $num(Sat_A) < num(Sat_B)$

3. We use the find_discrete method to find the points between t_{start} and t_{end} in which the two satellites distance d becomes $d < d_{min}$ and when it becomes $d > d_{min}$. The time frame between these two moments is considered a pass.

4. Same as in the Satellite-OGS case an array of time for the pass with Δt separation is created. For each time in the array, the following values are computed

   - Distance of the two satellites to the center of the earth in meters (Sat_Abs_Altitude)
   - Distance between the two satellites and the OGS in meters

5. Repeat the same process for all the possible combinations

6. Store the data in a dictionary and export it as a JSON file

6.1.4. Satellite constellation

To initialize a walker constellation we use the function walker_constellation, which takes the constellation parameters (i, t, p, f) the start time, the height of the orbits h and the constellation type (star or delta).

The algorithm is the following:

- Compute the orbital parameters relevant for building a TLE

- For each orbital plane, a reference satellite sat_ref is computed.

  Depending on the type of orbit the $RAAN$ is computed differently. For star constellations is $RAAN = 2\pi n/p + \pi/2$, and for delta constellations is $RAAN = \pi n/p + \pi/2$. Where $n = 0, ..., p-1$ is the orbital plane number

- sat_ref is propagated during a full orbit
- The orbital data of sat_ref is computed at times $t_n = t_{start} + n/N_{orbit}, n = 0, .., N_{orbit} - 1$, where $N_{orbit} = t/p$ is the number of satellites per orbit in the constellation.
- A new TLE is created for each of the times and stored in a dictionary. The TLE is used to create a satellite object.
- Repeat for each orbital plane

6.1.5. Visualization

We use VPython to create the graphics and Skyfield to compute the astronomical positions (see Fig. 4.2). The Earth is modelled as a sphere with a texture map overlayed. The objects that can be displayed are the locations of the OGSs, the satellites, the orbits and the active QKD links.

6.1.6. Network

The network simulation has three parts:

1. Create the network connections (BA method)
2. Create the users key request
3. Simulate an exchange of keys applying a prioritization scheme

Barabasi-Albert method

The BA method consists on creating a graph adding a new node at a time with n connections to the already existing nodes which are randomly chosen. The more connections a node already posses (degree), the higher chances to be selected for a new connection.

This simulates real social network such as the internet.

Prioritization scheme

The idea is to prioritize the links that can generate more key, but also the users with less accumulated key and bigger connections. We define the priority v as:

$$v = \frac{\text{potential\_SKL} \cdot \text{node\_degree}}{\text{accumulated\_SKL}} \tag{6.11}$$

Then find all the conflicting links and select the bigger v value

6.2. Mathematical methods

6.2.1. Gaussian Quadrature

We use the Gaussian quadrature to accelerate the optimizer. This method, uses orthogonal polynomials to compute accurate integrals numerically with few computations. The idea is the following.

If we have a weighted integral such as:

$$\int_a^b w(x)f(x)dx \tag{6.12}$$

we can find a polynomial of order n, p_n, that is orthogonal according to the inner product defined by $w(x)$, i.e. :

$$\left\langle x^k, p_n \right\rangle_w = \int_a^b w(x)x^k p_n(x)dx = 0, \text{for } k = 0, ..., n-1 \tag{6.13}$$

Then, the following equation is exact for polynomials up to order $2n$ (i.e., twice as accurate for half of the number of points)

$$\int_a^b w(x)f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i) \tag{6.14}$$

where x_i are the zeros of the polynomial and w_i are:

$$w_i = \frac{a_n}{a_{n-1}} \cdot \frac{\int_a^b \omega(x)p_{n-1}(x)^2\,dx}{p'_n(x_i)p_{n-1}(x_i)} \tag{6.15}$$

Where p_{n-1} is the orthongonal polynomial of order $n-1$. For a derivation of this see [Ste96]

Algorithm to compute orthogonal polynomials

There is a fast way to compute the roots and the weights for any weight function $w(x)$. In our case, the weight function is the distribution probability of the transmittance.

The method is the following. To construct an orthogonal polynomial we use the three-term recurrent relation with $p_{-1} = 0$ and p_0 = 1:

$$p_{n+1}(x) = xp_n(x) - \alpha_n p_n(x) - \beta_n p_{n-1}(x) \tag{6.16}$$

Where:

$$\alpha_n = \frac{\langle p_n, xp_n \rangle_w}{\langle p_n, p_n \rangle_w} \tag{6.17}$$

$$\beta_n = \frac{\langle p_n, p_n \rangle_w}{\langle p_{n-1}, p_{n-1} \rangle_w} \tag{6.18}$$

This algorithm is implemented in the function orthogonal_polynomial. It takes the wished number of points, and the arrays of values of the weight function. It returns the coefficients of the polynomial and the list of the α_i and β_i values.

The Golub-Welsch algorithm

The three-term recurrent relation [GW69] can be written like $\mathbf{J}\tilde{P} = x\tilde{P} - p_n(x)\mathbf{e}_n$, where $\tilde{P} = [p_0, p_1, ..., p_{n-1}]$, $\mathbf{e}_n$ is the nth standard basis vector, and $\mathbf{J}$ is:

$$\mathbf{J} = \begin{bmatrix} \alpha_0 & 1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_1 & 1 & \ddots & \vdots \\ 0 & \beta_2 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \alpha_{n-2} & 1 \\ 0 & \cdots & 0 & \beta_{n-1} & \alpha_{n-1} \end{bmatrix}. \tag{6.19}$$

The eigenvalues of $\mathbf{J}$ are the roots x_i of the polynomial. But it is easier to compute the similar (same eigenvalues and eigenvectors) and symmetric matrix $\mathcal{J}$:

$$\mathcal{J} = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & 0 & \cdots & 0 \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & \ddots & \vdots \\ 0 & \sqrt{\beta_2} & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \alpha_{n-2} & \sqrt{\beta_{n-1}} \\ 0 & \cdots & 0 & \sqrt{\beta_{n-1}} & \alpha_{n-1} \end{bmatrix}. \tag{6.20}$$

To compute the weight w_i the first value of the eigenvector $\phi_1^{(i)}$ is used:

$$w_i = \mu_0 {\phi_1^{(i)}}^2 \tag{6.21}$$

$$\mu_0 = \int_a^b w(x) \tag{6.22}$$

This algorithm is implemented in the function golub_welsch. It takes the arrays of values of the weight function and the list with the values of α_i and β_i. It returns a list with the roots and weights.

6.2.2. Product of random variables

Given two random independent continuous variables $X \in (0, 1]$ and $Y \in (0, 1]$, whose probability density functions are $p_X(x)$ and $p_Y(y)$, the total probability distribution $p_Z(z)$ of $Z = XY$ is [insert reference]:

$$p_Z(z) = \int_0^1 p_X(x)\, p_Y(z/x)\, \frac{1}{|x|}\, dx \tag{6.23}$$

However, this formula can be transformed into a convolution if we use decibels:

$$x_{dB} = 10log_{10}x \tag{6.24}$$

$$x = 10^{x_{dB}/10} \tag{6.25}$$

Then apply a change of variable, using that:

$$\frac{dx}{dx_{dB}} = 10^{x_{dB}} \frac{log10}{10} = x\frac{log10}{10} \tag{6.26}$$

We get the following integral

$$p_Z(z) = \frac{log10}{10} \int_{-\infty}^{0} p_X\left(10^{x_{dB}/10}\right) p_Y\left(10^{(z_{dB}-x_{dB})/10}\right)\, dx_{dB} \tag{6.27}$$

Which is basically a convolution, we can see it more clearly if we use the function $f_X(x_{dB}) = p_X(10^{x_{dB}/10})$

$$f_Z(z_{dB}) = \frac{log10}{10} \int_{-\infty}^{0} f_X(x_{dB})\, f_Y(z_{dB} - x_{dB})\, dx_{dB} \tag{6.28}$$

A convolution can be quickly computed using a Fast Fourier Transform (FFT) (method which is already exixting and well optimized in the scipy library)

Constant probability

If one of the two probabilities is just a constant value, we can model it as a Dirac delta function:

$$\delta_{x_0}(x) = \begin{cases} \infty, & x = x_0 \\ 0, & x \neq x_0 \end{cases} \tag{6.29}$$

Using the properties of the Dirac delta we get that:

$$p_Z(z) = \int_0^1 \delta_{x_0}(x)\, p_Y(z/x)\, \frac{1}{|x|}\, dx = \frac{1}{|x_0|} p_X(z/x_0) \tag{6.30}$$

Fast Fourier Transform for Convolution

Convolution is a mathematical operation on two functions f and g that produces a third function $f * g$, as the integral of the product of the two functions after one is reflected about the y-axis and shifted. This is exactly the operation used to find the product probability of two random variables.

There is a fast way to compute this using the FFT. The discrete formula for a convolution is

$$(f * g)[n] = \sum_{m=-\infty}^{m=\infty} f[m]g[n-m] = \sum_{m=-\infty}^{m=\infty} f[n-m]g[m] \tag{6.31}$$

In our case, we deal with probability functions that have a support in the set $\{-M, -M+1, ..., M-1, M\}$, so:

$$(f * g)[n] = \sum_{m=-M}^{m=M} f[n-m]g[m] \tag{6.32}$$

This convolution can be computed in a $O(Nlog(N))$ complexity instead of $O(N^2)$ using FFT

Given an array f of length N and array g of length M and a linear convolution length of $L = M + N + 1$, the algorithm is the following:

1. Zero pad both arrays to get lenth L array (this avoids circular convolution)
2. Compute the FFT of each array
3. Multiply both arrays (point-wise)
4. Take the inverse of the FFT (maybe it will be necessary to discard the imaginary part due to numerical errors)

This method for convolution is included and optimized in the scipy library.

Demonstration

This method works for any Discrete Fourier Transform (DFT), however, if we use a slow version of DFT, we do not obtain any improvement in the algorithm complexity. That is why FFT algorithm is used.

We call the convolution $y = (\tilde{f} * \tilde{g})$. Where $\tilde{f}$ and $\tilde{g}$ mean the zero-padded f´versions of the arrays. Then, apply a DFT to y

$$Y[k] = \sum_{n=0}^{L-1} y[n] e^{-j\frac{2\pi}{L}kn}$$

Since $y[n] = \sum_{m=0}^{L-1} \tilde{f}[m] \cdot \tilde{g}[n-m]$, substitute into the DFT:

$$Y[k] = \sum_{n=0}^{L-1} \left(\sum_{m=0}^{L-1} \tilde{f}[m] \cdot \tilde{g}[n-m] \right) e^{-j\frac{2\pi}{L}kn}$$

Swap the order of summation:

$$Y[k] = \sum_{m=0}^{L-1} \tilde{f}[m] \left(\sum_{n=0}^{L-1} \tilde{g}[n-m] e^{-j\frac{2\pi}{L}kn} \right)$$

Change variable $l = n - m \Rightarrow n = l + m$:

$$Y[k] = \sum_{m=0}^{L-1} \tilde{f}[m] \left(\sum_{l=0}^{L-1} \tilde{g}[l] e^{-j\frac{2\pi}{L}k(l+m)} \right)$$

Split the exponential:

$$Y[k] = \sum_{m=0}^{L-1} \tilde{f}[m] e^{-j\frac{2\pi}{L}km} \left(\sum_{l=0}^{L-1} \tilde{f}[l] e^{-j\frac{2\pi}{L}kl} \right)$$

The inner and outer sums are the DFT of the convolving functions:

$$\mathrm{DFT}(f * g)[k] = \mathrm{DFT}(f)[k] \cdot \mathrm{DFT}(g)[k]$$

So, to get the final convolution you only need to undo the DFT

7. Results

7.1. One satellite, two ground stations

7.1.1. Transmittance

We first test the code with the simplest scenario: a single LEO satellite in a SSO orbiting around the Earth and two OGS. To keep the analysis simple, we consider the case where there are no conflicts between the links, that is, the ground stations that are far enough to avoid possible simultaneous sight to the satellite from both of them. The selected stations are Lisbon (38.71°N,9.14°W) and Athens (38.0°N, 23.733°E) because they are both European capitals that fulfill these conditions and are also located in similar latitudes.

The transmittance distribution is the sum of all the single distributions for each time step times the pass duration. In Fig. 7.1 two different passes for the two different ground stations are shown. Note that the units in the X axis are in dB, i.e. logarithmic scale. In this scale the peak appears to have a much higher relevance than it actually has. If we look at the mean value of the distributions, we see that is shifted to the right of the peak.

7.1.2. Key exchange during time

It is also interesting to see the Secure Key Length (SKL) generated along time. Note that, for this latitude, the increase rate is roughly constant for every season. That is because the SSO is selected so the satellite pass always happens such that it is always night for every pass no matter the position of the Earth with respect to the Sun.

7.2. One satellite, 367 ground stations

The next step for our study is to increase the number of ground stations. We selected a list of 367 OGS around the globe and run the test with the same satellite during a full year to obtain a better statistic of the different passes. A curated set of OGS locations was compiled in a orbits study [Fuc+18]

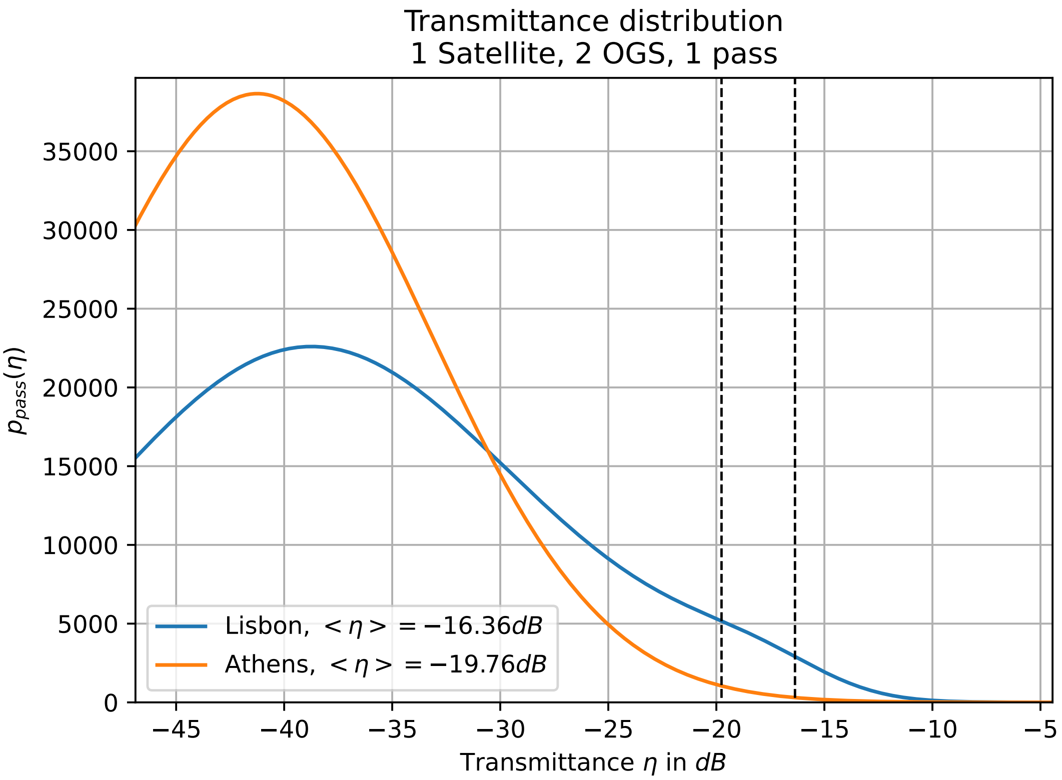
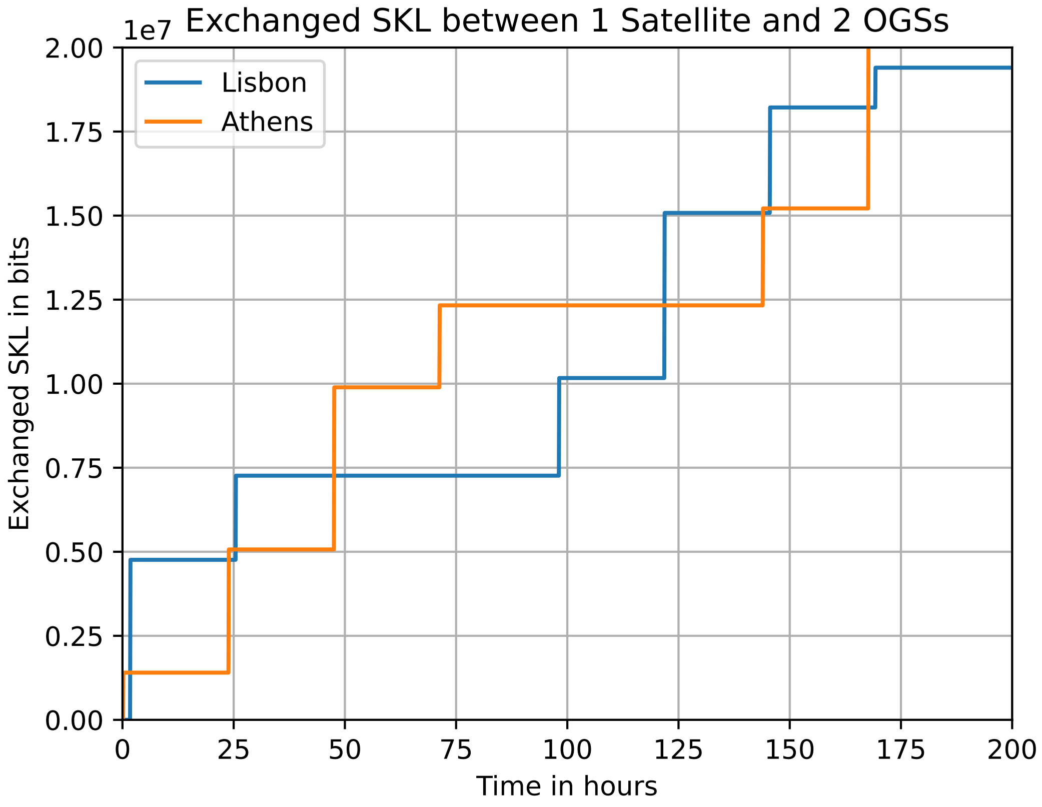


Figure 7.1.: Transmittance distribution during one pass of a single satellite over two different OGS with a similar latitude (Athens and Lisbon). It is not a PDF because the total area of the distribution equals the total pass duration in seconds. The dashed lines represent the expected values of each distribution.

7.2.1. SKL vs. maximum elevation angle

In the data analysis we find a direct relationship between the maximum angle of each satellite pass and the SKL generated during that pass.

In Fig. 7.5 a correlation can be clearly observed, however there is a broad range of SKL values for each angle. If we identify the passes for each ground station (Fig. 7.6) it is clear that each ground station follows a very specific pattern, which corresponds to an order 3 polynomial (Fig. 7.7). This fit can be used to speed up the computations and obtain an accurate approximation of the SKL just based on the maximum elevation angle.



h!

Figure 7.2.: SKL generated for one satellite and two OGS during 200 hours.

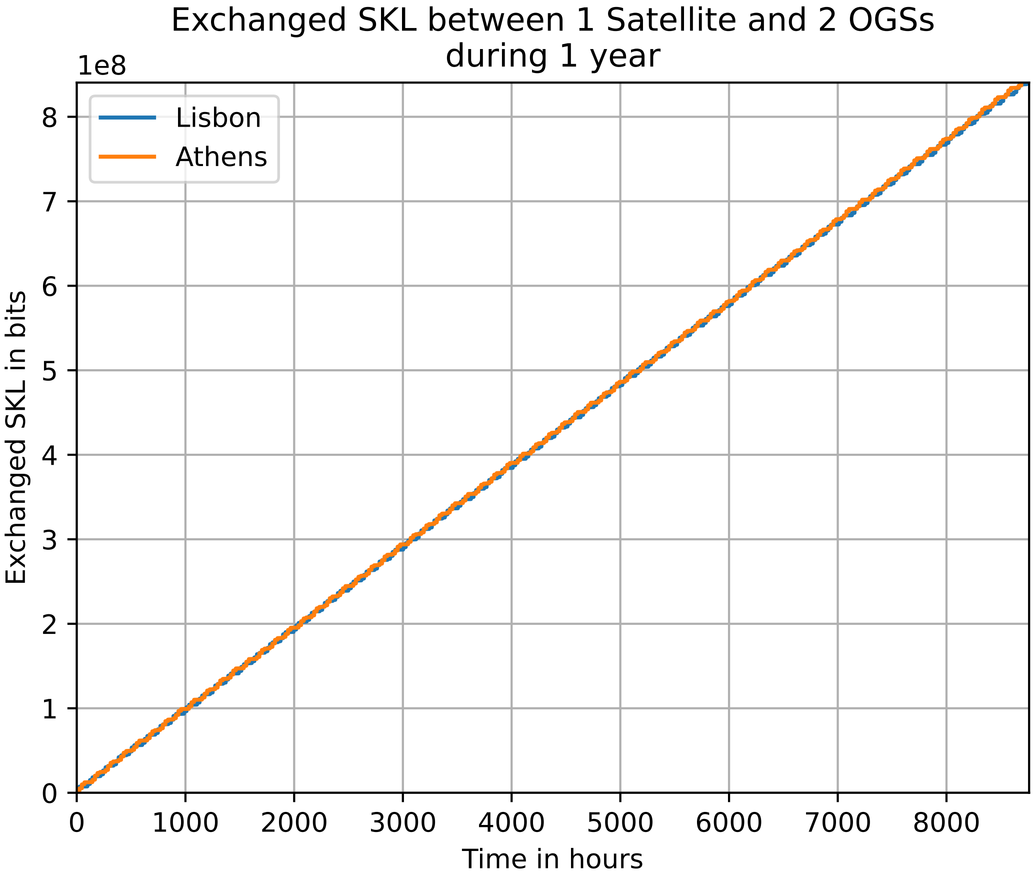


Figure 7.3.: SKL generated for one satellite and two OGS during one year.

Also note that there are some points that fall out of the general pattern. This is due to the optimization algorithm, since it does not always converge and, therefore, the SKL does not correspond to the optimal parameters.

7.2.2. Influence of the latitude

Another relationship that we found with our simulator is the total amount of key generated in one year vs. the latitude of the ground station. The stations that are closer to the poles receive more amount of key (Fig. 7.8).

This is due to the SSO which provides better coverage for higher latitudes. As it is shown in Fig. 7.9 higher altitudes get more passes. This is easy to see if we imagine a satellite orbit with inclination $i = 90^\circ$. In this scenario, the ground station with latitudes

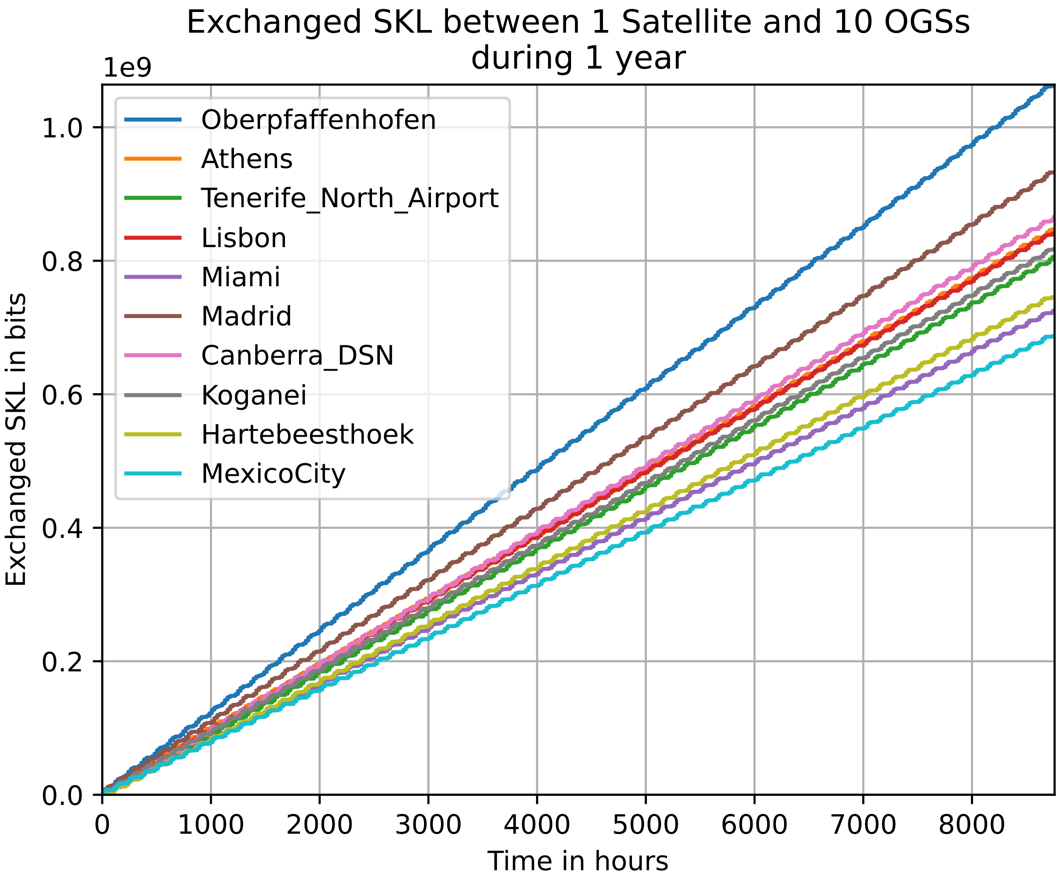


Figure 7.4.: SKL generated for one satellite and 10 OGS during one year.

equal to 90° and −90° (the poles) get a pass for every satellite revolution, while a point in the equator needs to wait a few satellite revolutions to get a new pass. The greater the latitude, the lesser the angular velocity around the Earth axis, and therefore the more frequent passes of the satellite. That is very clear if you look at 7.4, where Oberpfaffenhoffen (lat=48°) increases the key exchange very rapidly with respect to Mexico city (lat=19.4°), which has a much smaller latitude.

7.3. Inter-satellite SKL

To estimate how much key can be exchanged between the two orbits, we only need to test two satellites and try to find the optimal height difference that allows the biggest

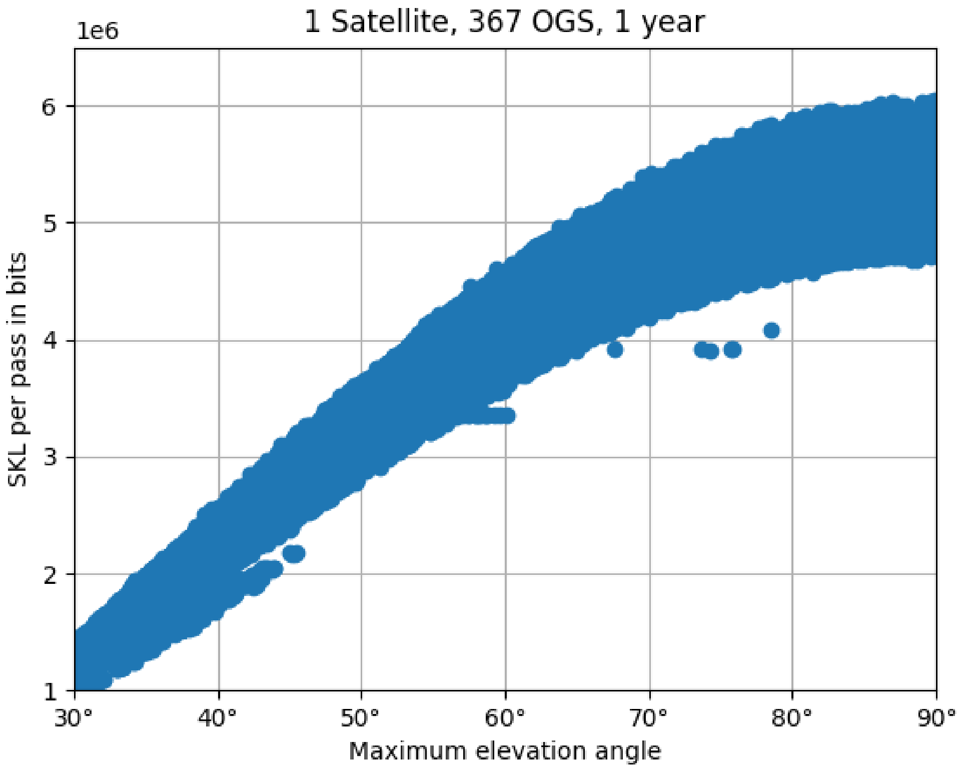


Figure 7.5.: SKL generated during each satellite pass with respect to the maximum elevation angle of the pass for all the ground stations.

key exchange.

The result of this simulation is that there is a sweet spot around 65 km of orbital height difference that produces the maximum amount of key exchange (Fig. 7.10)

The number of passes follows a linear correlation with the height difference (Fig. 7.11), according to eq. 4.6.

However, the average key exchanged per pass does not fulfill a linear pattern (Fig. 7.10). It is obvious that it is descending because the bigger *dh* the bigger the orbital difference speed and therefore the shorter the passes are (Fig. 7.12). However we can differentiate two regimes: a non-linear one and a linear one. The non-linear part is most likely due to the near-field corrections.

The combination of this two things (number of passes and SKL per pass) result in the peak that shows in Fig. 7.10

Lastly, it is important to note that there is a drop in the total key exchanged per year

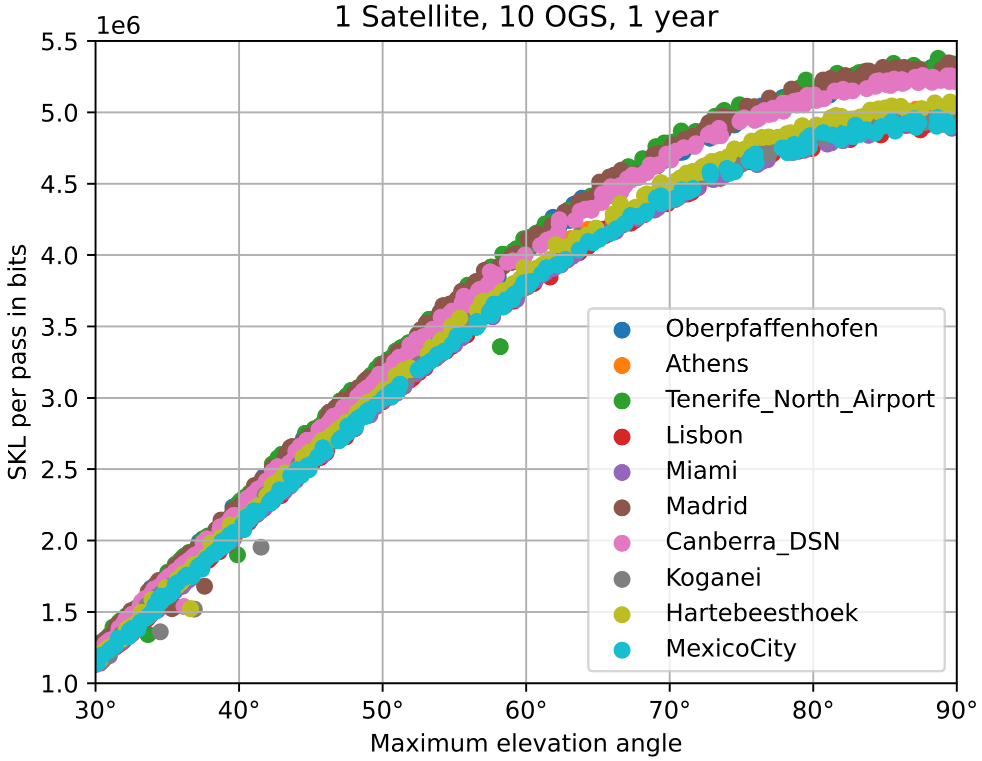


Figure 7.6.: SKL generated during each pass with respect to to the maximum elevation angle of the pass for for 10 ground stations separated by colors.

over 65 km (Fig. 7.10), this is likely due to the noise, whose relative importance with respect to the signal becomes bigger for greater distances (recall that the telescope of the satellite is a small one compared to the ground stations).

7.4. Network results

Finally, we simulate a whole network that requires key in a realistic way and test how many satellites are needed to fulfill these requests. Here is when we apply the prioritization algorithm; until now, we only performed studies of general satellite passes over the ground stations or other satellites. This last section is a test of how this would work in a real network where different ground stations may request key at the same time and satellites need to prioritize which key to exchange.

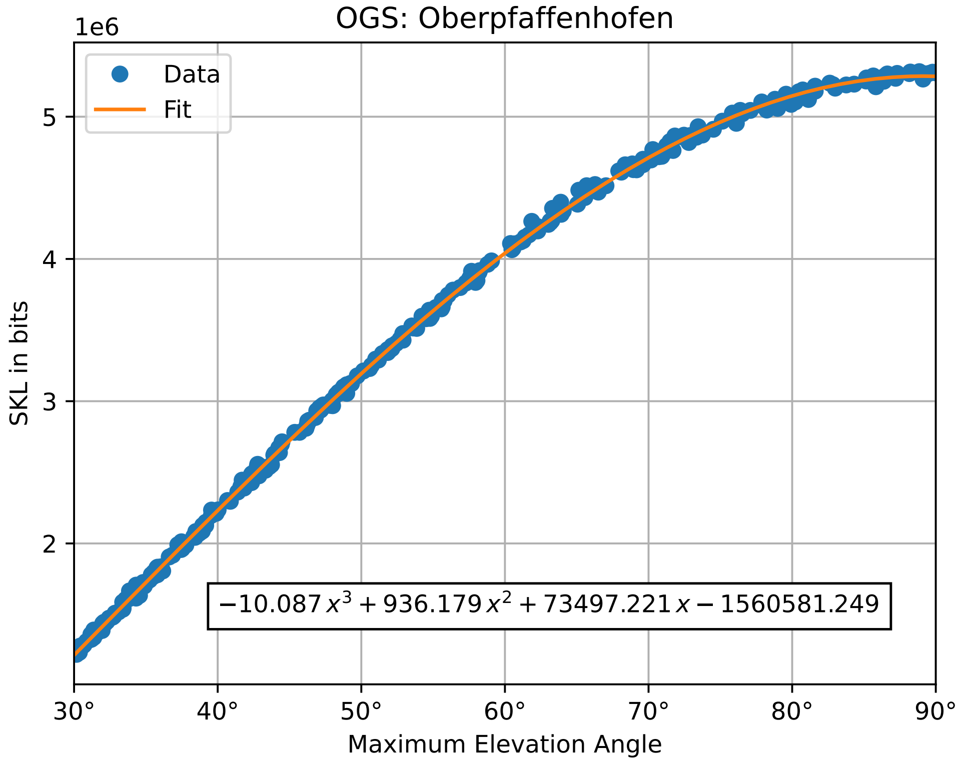


Figure 7.7.: Polynomial fit for the SKL generated during each satellite pass with respect to the maximum elevation angle for the Oberpfaffenhoffen station.

If we apply the prioritization algorithm (sec. 5.2.3) and run an actual simulation of one satellite and different numbers of stations, we can see the amount of key that is actually exchanged between ground stations (Fig. 7.14 and 7.16). We see that all the OGSs that have a connection (Fig. 7.13 and Fig. 7.15) in the network managed to exchange some amount of key. In the case of 3 OGS both end-to-end connections get similar amount of key, because of the lack of overlapping. However, for the 10 OGS case, this exchange is not equal: since all links have the same probability of being active, this means that there have been unsuccessful key request that the satellite was not able to fulfill due to conflicting requests.

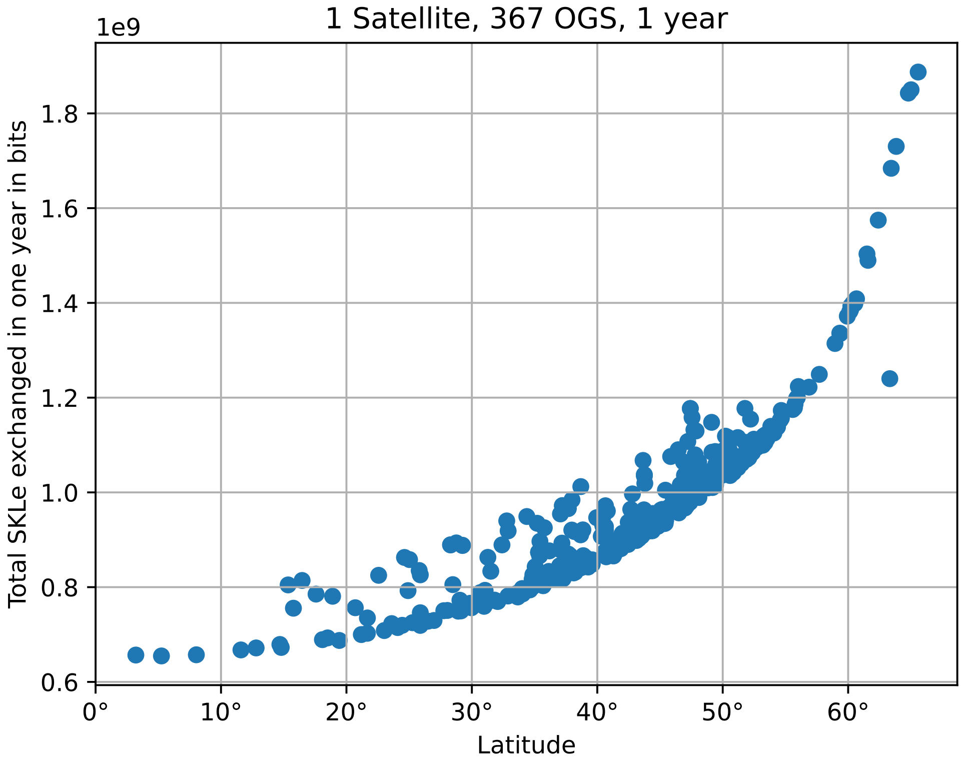


Figure 7.8.: SKL exchanged between the satellite and each ground station during a year depending on the latitude.

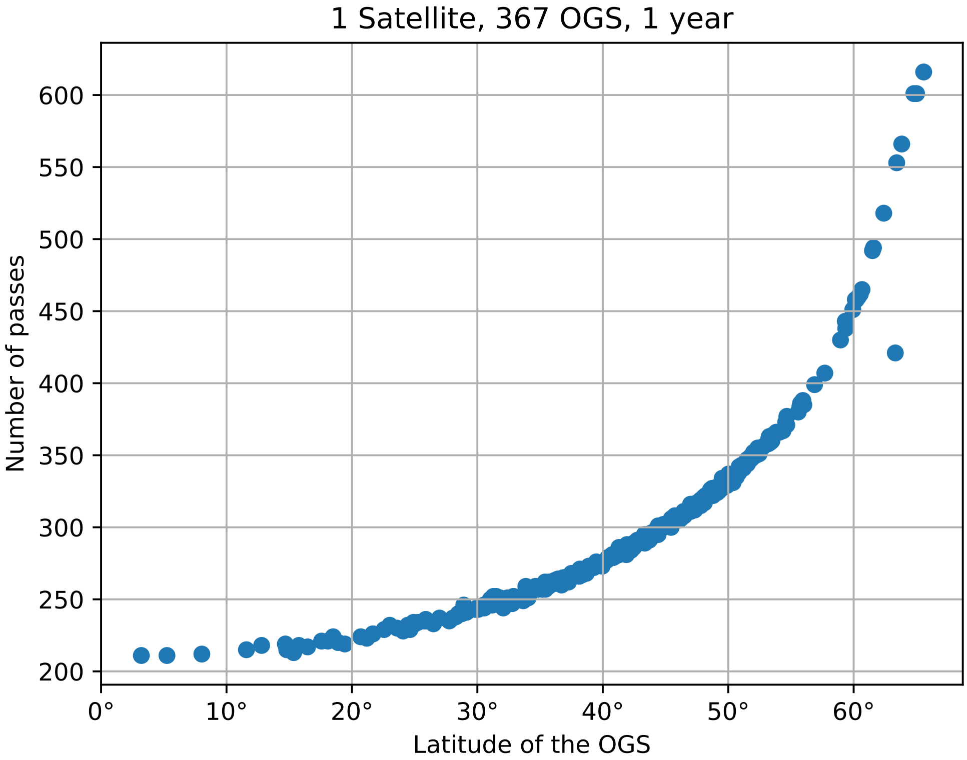


Figure 7.9.: Total number of passes between the satellite and each ground station during a year depending on the latitude.

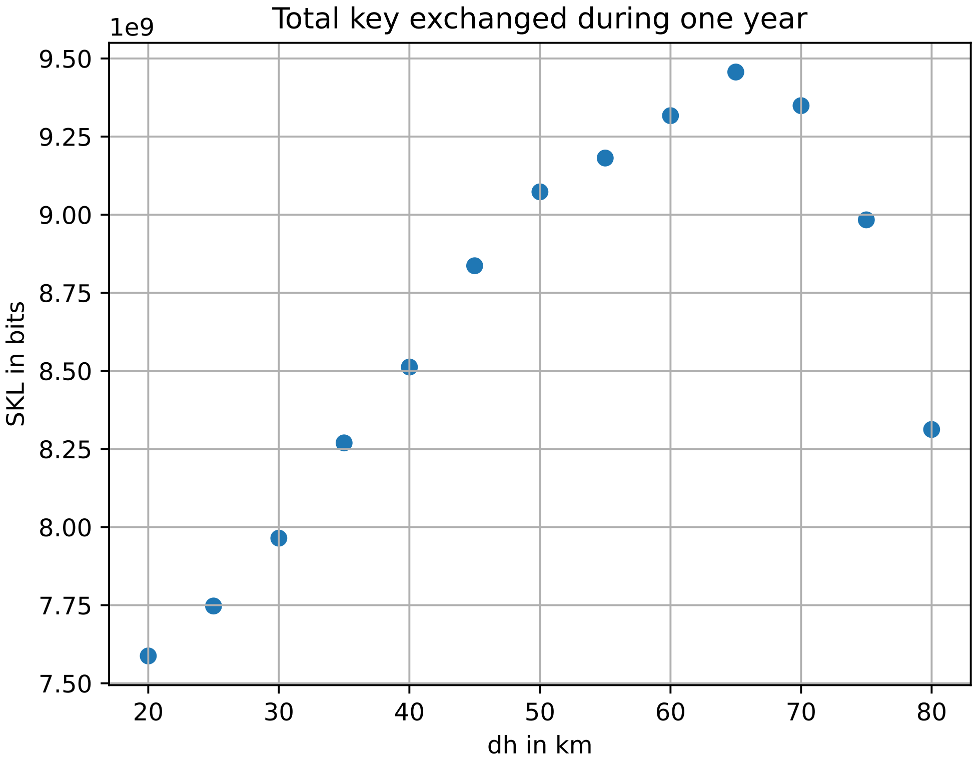


Figure 7.10.: SKL exchanged between two satellites for different orbital height differences.

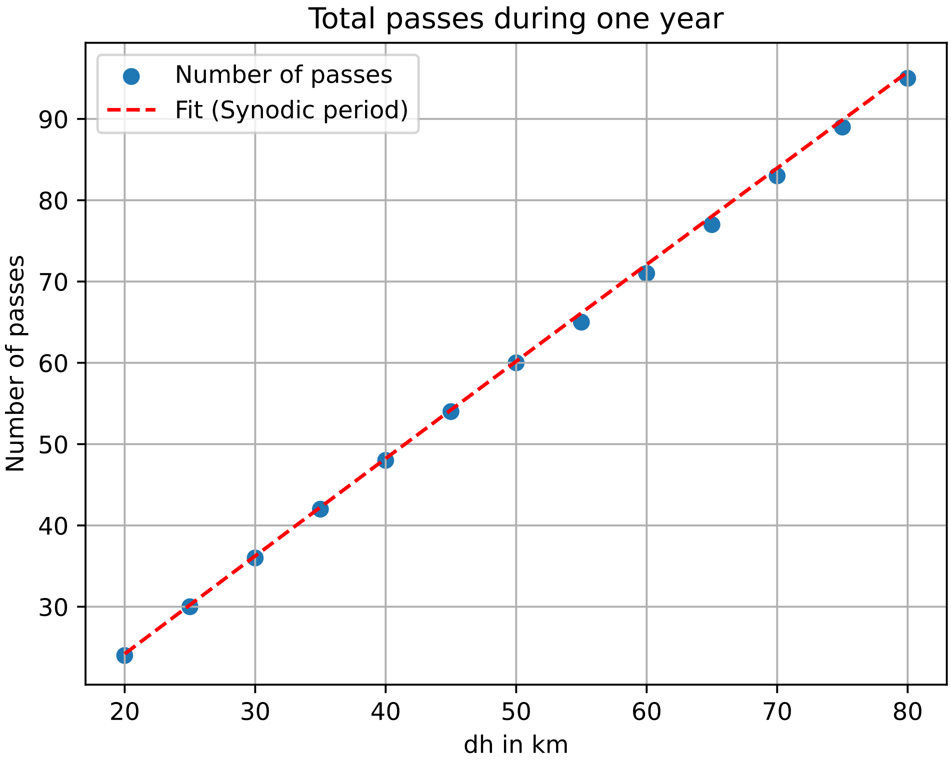


Figure 7.11.: Number of total passes between two satellites for different orbital height differences. The fit is realized using the synodic period formula (eq. 4.6)

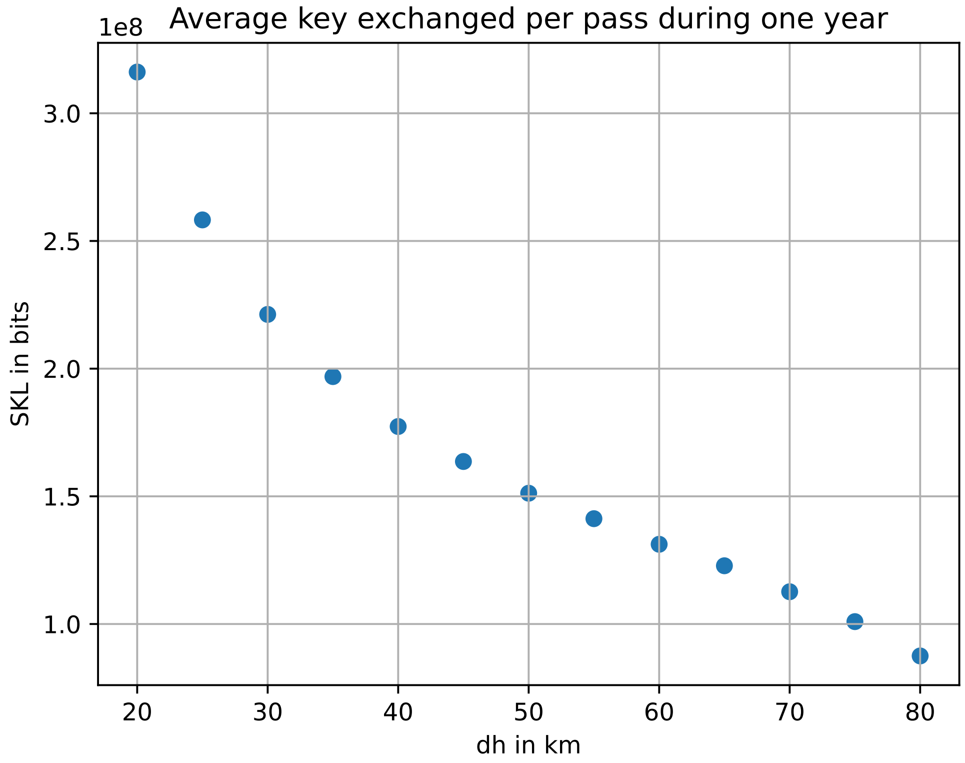


Figure 7.12.: Average SKL exchanged per pass between two satellites for different orbital height differences.

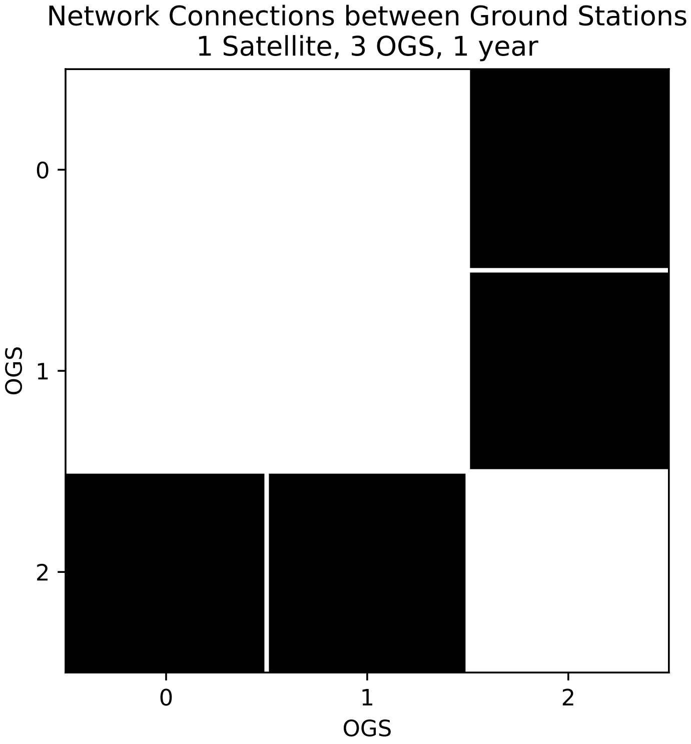


Figure 7.13.: Network connections created using the BA algorithm. Black boxes means that there is an edge in the network between two ground stations.

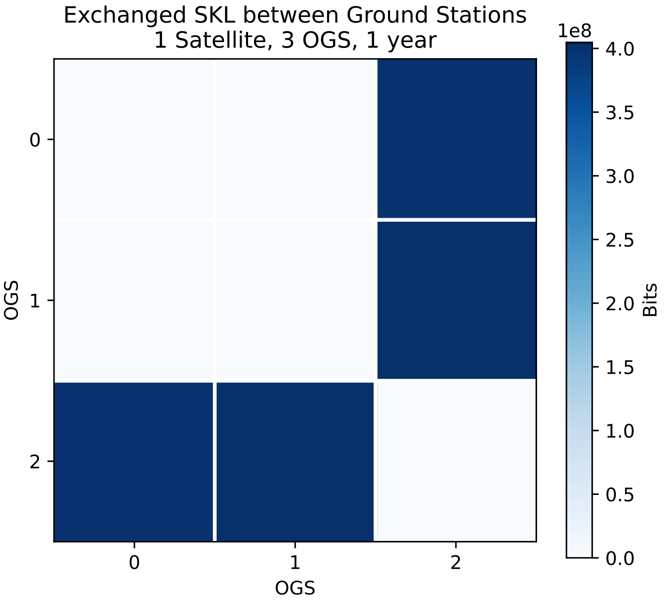


Figure 7.14.: Average SKL exchanged key during a year between different ground stations. White boxes means there is no established connection between the corresponding two OGS

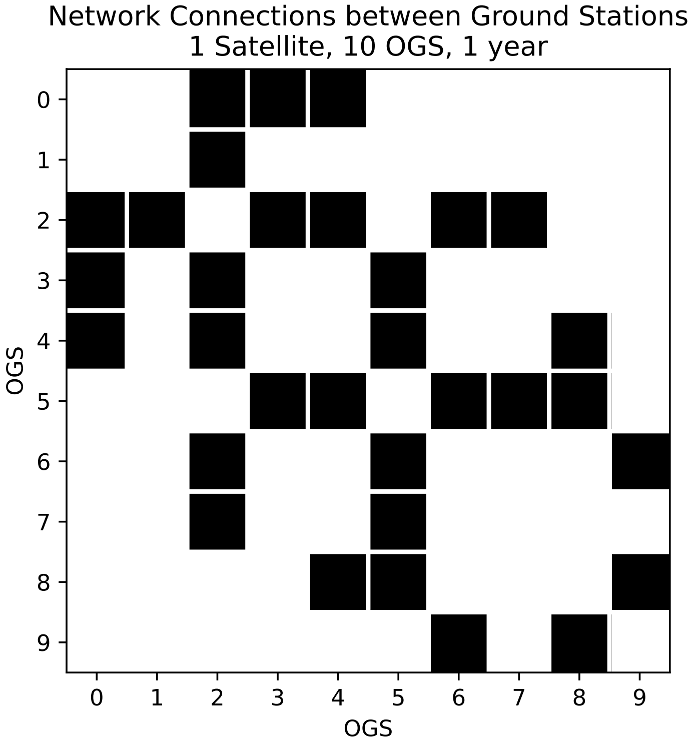


Figure 7.15.: Network connections created using the BA algorithm. Black boxes means that there is an edge in the network between two ground stations.

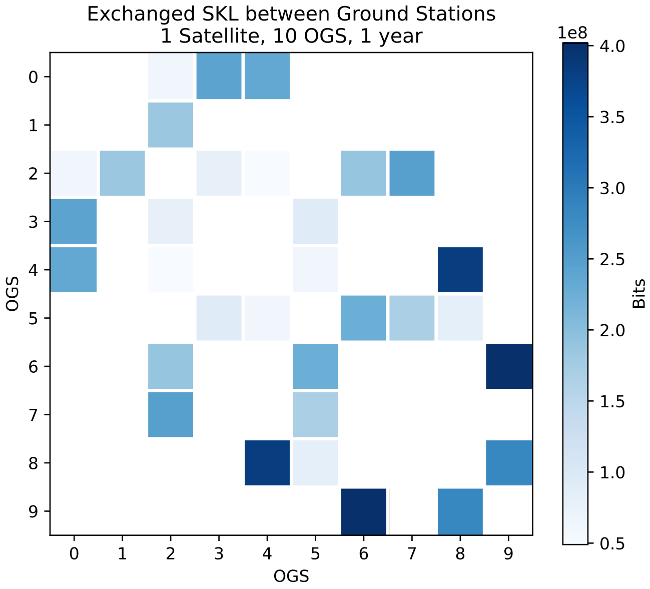


Figure 7.16.: Average SKL exchanged key during a year between different ground stations. White boxes means there is no established connection between the corresponding two OGS

8. Conclusions

In this work, we have presented a simulation framework for analyzing satellite-based QKD systems, with a focus on a two-height, single-orbital plane constellation. We use a SSO to allow night time operations throughout the year. We propose this architecture as an alternative to traditional Walker constellations, aiming to optimize inter-satellite quantum links and enable scalable, global QKD networks. Through our simulations, we assessed the key distribution performance with a varying number of satellites and ground stations.

As expected, our results demonstrate that the total quantum key generated over a year by different optical ground stations exhibits a linear growth trend, highlighting the predictable scaling potential of the system. Furthermore, we found that it is possible to fit to a very high degree of accuracy the length of the secure key with respect to the maximum elevation angle of satellite passes, using a third-degree polynomial. This offers an useful approximation for future optimization efforts of the simulator. Also, ground stations located at higher latitudes tend to generate bigger amount of key, benefiting from a greater number of satellite passes which results from the use of SSO.

Regarding the parameters for the satellite orbits, our simulations identified an optimal separation between the two orbital heights in the constellation at approximately 65 km, balancing inter-satellite visibility and key throughput.

To explore the system's applicability to real-world scenarios, we also implemented a basic end-user requirement model, enabling analysis of how user demand could influence constellation design and performance.

Future work will focus on integrating atmospheric cloud coverage statistics to better estimate realistic link availability, employing the key-vs-elevation-angle polynomial fitting to accelerate simulation runtimes, further studies on the satellite network capacities to give service to different number of users, and extending the simulator to include alternative QKD protocols such as BBM92. These directions aim to further refine the simulator's predictive accuracy and broaden its relevance to practical mission planning and deployment.

A. Simulation Parameters

Time parameters	
Start time	2024-11-26 00:00:00 UTC
End time	2025-11-26 00:00:00 UTC
Time step	10.0 s

Table A.1.: Time parameters

Common physical parameters	
Wavelength	1500 nm
Min. elevation for valid pass	20°
Min. max. elevation for valid pass	30°
Max. distance for valid inter-sat link	100 km

Table A.2.: Common physical parameters

Satellite parameters	
Orbital parameters	
Height	500km-580km
Optical terminals	
Transmitter	
Pointing bias	10^{-6} rad
Pointing jitter	10^{-6} rad
Beam Waist	38 mm
Half divergence	$22.0 \cdot 10^{-6}$ rad
Receiver	
External radius	42.5 mm
Internal radius	10 mm
Internal efficiency	0.8
QKD terminals	
Transmitter	
Repetition rate	10^{-6} rad
Polarization misalignment	0.1 rad
Receiver	
Dark rate	3000
Efficiency	0.2
Gating time	10^{-9}s
Dead time	$100 \cdot 10^{-9}$ s

Table A.3.: Satellite parameters

OGS parameters Receiver optical terminal	
External radius	400 mm
Internal radius	150 mm
Internal efficiency	0.256 (includes fiber coupling)
Receiver QKD terminals	
Dark rate	90
Efficiency	0.9
Gating time	10^{-9}s
Dead time	10^{-6} s

Table A.4.: OGS parameters

Abbreviations

AO Adaptive Optics

BA Barabasi-Albert

BSM Bell State Measurement

DFT Discrete Fourier Transform

EB Entanglement-Based

FFT Fast Fourier Transform

FSO Free Space Optics

ISS International Space Station

LDPC Low Density Parity Check

LEO Low Earth Orbit

MDI Measure Device Independent

OGS Optical Ground Station

PDF Probability Density Function

PNS Photon-Number-Splitting

PM Prepare and Measure

PSI Power Scintillation Index

QKD Quantum Key Distribution

QBER Quantum Bit Error Rate

RAAN Right Ascension of the Ascending Node

SDP Simplified Deep Space Perturbations

SGP Simplified General Pertutbations

SKL Secure Key Length

SSO Sun Synchronous Orbit

TLE Two-Line Element

TN Trusted Node

WCS Weak Coherent State

WGS84 World Geodesic System 1984

List of Figures

List of Tables

Bibliography

[AB02] R. Albert and A.-L. Barabási. "Statistical mechanics of complex networks." In: *Reviews of Modern Physics* 74.1 (Jan. 2002), pp. 47–97. DOI: 10.1103/revmodphys.74.47.

[AP05] L. C. Andrews and R. L. Phillips. "Laser beam propagation through random media." In: *Laser Beam Propagation Through Random Media: Second Edition* (2005).

[Ash+24] A. Ashkenazy, Y. Idan, D. Korn, D. Fixler, B. Dayan, and E. Cohen. "Photon Number Splitting Attack – Proposal and Analysis of an Experimental Scheme." In: *Adv. Quantum Technol.* 7.7 (2024), p. 2300437. DOI: 10.1002/qute.202300437.

[Bat+20] R. Bate, D. Mueller, J. White, and W. Saylor. *Fundamentals of Astrodynamics*. Dover Books on Physics. Dover Publications, 2020. ISBN: 9780486497044.

[BB84] C. H. Bennett and G. Brassard. "Quantum Cryptography: Public Key Distribution and Coin Tossing." In: *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing*. Bangalore, Dec. 10, 1984, pp. 175–179.

[BBM92] C. H. Bennett, G. Brassard, and N. D. Mermin. "Quantum cryptography without Bell's theorem." In: *Phys. Rev. Lett.* 68 (5 Feb. 1992), pp. 557–559. DOI: 10.1103/PhysRevLett.68.557.

[Ben+05] M. Ben-Or, M. Horodecki, D. W. Leung, D. Mayers, and J. Oppenheim. "The Universal Composable Security of Quantum Key Distribution." In: *Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings*. Vol. 3378. Lecture Notes in Computer Science. Springer, 2005, pp. 386–406. DOI: 10.1007/978-3-540-30576-7_21.

[Boa04] R. J. Boain. *A-B-Cs of sun-synchronous orbit mission design*. Version V2. 2004. DOI: 2014/37900.

[Bra+00] G. Brassard, N. Lütkenhaus, T. Mor, and B. C. Sanders. "Limitations on practical quantum cryptography." In: *Physical Review Letters* 85.6 (Aug. 2000), pp. 1330–1333. DOI: 10.1103/physrevlett.85.1330.

[Bro02] C. D. Brown. *Elements of spacecraft design*. American Institute of Aeronautics and Astronautics, Inc., Jan. 2002. DOI: `10.2514/4.861796`.

[BS94] G. Brassard and L. Salvail. “Secret-Key Reconciliation by Public Discussion.” In: *Advances in Cryptology—EUROCRYPT ’93*. Ed. by T. Helleseth. Berlin: Springer, 1994, pp. 410–423.

[Cal+24] G. Calistro-Rivera, O. Heirich, A. Shrestha, A. Ferenczi, A. Duliu, J. Eppinger, B. F. Castella, C. Fuchs, E. Garbagnati, D. Laidlaw, P. Lützen, I. D. Marco, F. Moll, J. Prell, A. Reeves, J. R. Nonay, C. Roubal, J. S. Torres, and M. Wagner. *Building Europe’s first space-based Quantum Key Distribution system – The German Aerospace Center’s role in the EAGLE-1 mission*. 2024. arXiv: `2412.03222 [quant-ph]`.

[Fuc+18] C. Fuchs, N. Perlot, J. Riedi, and J. Perdigues. “Performance estimation of Optical LEO Downlinks.” In: *IEEE Journal on Selected Areas in Communications* 36.5 (May 2018), pp. 1074–1085. DOI: `10.1109/jsac.2018.2832831`.

[GW69] G. H. Golub and J. H. Welsch. “Calculation of Gauss quadrature rules.” In: *Mathematics of Computation* 23.106 (Jan. 1969), pp. 221–230. DOI: `10.1090/s0025-5718-69-99647-1`.

[HR80] F. R. Hoots and R. L. Roehrich. *Models for propagation of NORAD element sets*. Tech. rep. Aerospace Defense Command Peterson AFB Co Office of Astrodynamics, Dec. 1980. DOI: `10.21236/ada093554`.

[Kar+16] H. Karttunen, P. Kröger, H. Oja, M. Poutanen, and K. Donner. *Fundamental Astronomy*. Springer Berlin Heidelberg, 2016. ISBN: 9783662530450.

[Las08] Lasunncty. *Diagram of the OSI Model*. `https://commons.wikimedia.org/w/index.php?curid=8971052`. Image by Lasunncty at the English Wikipedia. Licensed under CC BY-SA 3.0. 2008.

[LCQ12] H.-K. Lo, M. Curty, and B. Qi. “Measurement-Device-Independent Quantum Key Distribution.” In: *Phys. Rev. Lett.* 108 (13 Mar. 2012), p. 130503. DOI: `10.1103/PhysRevLett.108.130503`.

[Li+25] Y. Li, W.-Q. Cai, J.-G. Ren, C.-Z. Wang, M. Yang, L. Zhang, H.-Y. Wu, L. Chang, J.-C. Wu, B. Jin, H.-J. Xue, X.-J. Li, H. Liu, G.-W. Yu, X.-Y. Tao, T. Chen, C.-F. Liu, W.-B. Luo, J. Zhou, H.-L. Yong, Y.-H. Li, F.-Z. Li, C. Jiang, H.-Z. Chen, C. Wu, X.-H. Tong, S.-J. Xie, F. Zhou, W.-Y. Liu, Y. Ismail, F. Petruccione, N.-L. Liu, L. Li, F. Xu, Y. Cao, J. Yin, R. Shu, X.-B. Wang, Q. Zhang, J.-Y. Wang, S.-K. Liao, C.-Z. Peng, and J.-W. Pan. “Microsatellite-based real-time quantum key distribution.” In: *Nature* (Mar. 2025). DOI: `10.1038/s41586-025-08739-z`.

[Lia+17] S.-K. Liao, W.-Q. Cai, W.-Y. Liu, L. Zhang, Y. Li, J.-G. Ren, J. Yin, Q. Shen, Y. Cao, Z.-P. Li, F.-Z. Li, X.-W. Chen, L.-H. Sun, J.-J. Jia, J.-C. Wu, X.-J. Jiang, J.-F. Wang, Y.-M. Huang, Q. Wang, Y.-L. Zhou, L. Deng, T. Xi, L. Ma, T. Hu, Q. Zhang, Y.-A. Chen, N.-L. Liu, X.-B. Wang, Z.-C. Zhu, C.-Y. Lu, R. Shu, C.-Z. Peng, J.-Y. Wang, and J.-W. Pan. "Satellite-to-ground quantum key distribution." In: *Nature* 549.7670 (Aug. 2017), pp. 43–47. DOI: 10.1038/nature23655.

[Lim+14] C. C. W. Lim, M. Curty, N. Walenta, F. Xu, and H. Zbinden. "Concise security bounds for practical decoy-state quantum key distribution." In: *Physical Review A* 89.2 (Feb. 2014). DOI: 10.1103/physreva.89.022307.

[Ma+05] X. Ma, B. Qi, Y. Zhao, and H.-K. Lo. "Practical decoy state for quantum key distribution." In: *Phys. Rev. A* 72 (1 July 2005), p. 012326. DOI: 10.1103/PhysRevA.72.012326.

[Ma+16] X. Ma, X. Yuan, Z. Cao, B. Qi, and Z. Zhang. "Quantum random number generation." In: *npj Quantum Information* 2.1 (June 2016), pp. 1–9. DOI: 10.1038/npjqi.2016.21.

[Mak+24] V. Makarov, A. Abrikosov, P. Chaiwongkhot, A. K. Fedorov, A. Huang, E. Kiktenko, M. Petrov, A. Ponosova, D. Ruzhitskaya, A. Tayduganov, D. Trefilov, and K. Zaitsev. "Preparing a commercial quantum key distribution system for certification against implementation loopholes." In: *Physical Review Applied* 22.4 (Oct. 2024). DOI: 10.1103/physrevapplied.22.044076.

[MN96] D. MacKay and R. Neal. "Near Shannon limit performance of low density parity check codes." In: *Electronics Letters* 32.18 (Jan. 1996), p. 1645. DOI: 10.1049/el:19961141.

[NAS00] NASA. *Two-Line Element Set Format (TLE) Definition*. https://web.archive.org/web/20000301052035/http://spaceflight.nasa.gov/realdata/sightings/SSapplications/Post/JavaSSOP/SSOP_Help/tle_def.html. Accessed via Internet Archive on June 10, 2025. 2000.

[Ors+25] D. Orsucci, P. Kleinpaß, J. Meister, I. De Marco, S. Häusler, T. Strang, N. Walenta, and F. Moll. "Assessment of practical satellite quantum key distribution architectures for Current and Near-Future Missions." In: *International Journal of Satellite Communications and Networking* (Feb. 2025). DOI: 10.1002/sat.1544.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. "A method for obtaining digital signatures and public-key cryptosystems." In: *Commun. ACM* 21.2 (Feb. 1978), pp. 120–126. ISSN: 0001-0782. DOI: 10.1145/359340.359342.

[Scr+22] A. Scriminich, G. Foletto, F. Picciariello, A. Stanco, G. Vallone, P. Villoresi, and F. Vedovato. "Optimal design and performance evaluation of free-space quantum key distribution systems." In: *Quantum Science and Technology* 7.4 (Aug. 2022), p. 045029. DOI: `10.1088/2058-9565/ac8760`.

[Sho97] P. W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer." In: *SIAM Journal on Computing* 26.5 (1997), pp. 1484–1509. DOI: `10.1137/S0097539795293172`. eprint: `https://doi.org/10.1137/S0097539795293172`.

[Ste96] G. W. Stewart. *Afternotes on Numerical Analysis*. Society for Industrial and Applied Mathematics, 1996. Chap. 23. DOI: `10.1137/1.9781611971491`. eprint: `https://epubs.siam.org/doi/pdf/10.1137/1.9781611971491`.

[Tom+11] M. Tomamichel, C. Schaffner, A. Smith, and R. Renner. "Leftover hashing against quantum side information." In: *IEEE Transactions on Information Theory* 57.8 (Aug. 2011), pp. 5524–5535. DOI: `10.1109/tit.2011.2158473`.

[Wal84] J. G. Walker. "Satellite Constellations." In: *Journal of the British Interplanetary Society* 37 (Dec. 1984), p. 559.

[Xu+20] F. Xu, X. Ma, Q. Zhang, H.-K. Lo, and J.-W. Pan. "Secure quantum key distribution with realistic devices." In: *Reviews of Modern Physics* 92.2 (May 2020). DOI: `10.1103/revmodphys.92.025002`.

[Yur18] H. T. Yura. "Optical downlink propagation from space-to-earth: aperture-averaged power fluctuations, temporal covariance and power spectrum." In: *Opt. Express* 26.21 (Oct. 2018), pp. 26787–26809. DOI: `10.1364/OE.26.026787`.