
## About the Article

As satellites become linchpins of modern life and targets in geopolitical conflict, the security of space infrastructure is increasingly under threat. Erik Lewerenz and Heiko Radde examine the growing vulnerabilities in satellite communications and call for urgent adoption of post-quantum cryptography. Drawing on real-world case studies and in-orbit tests, they showcase how new cryptographic standards can future-proof satellites before quantum computers render today's defences obsolete.

## About the Author

**Heiko Radde** is an Embedded Software Engineer at Berlin Space Technologies, where he designs, implements, and tests firmware for satellite systems. He specializes in bare-metal C and C++ programming and contributes to all stages of development across various device subsystems. With a focus on innovative space missions, he has worked on projects like the AFR satellite, integrating advanced technologies such as post-quantum cryptography into aerospace platforms.

**Erik Lewerenz** works at Berlin Space Technologies GmbH, and has previous experience with aerospace, having worked at the German Aerospace Center. In March 2025 he graduated with a Master's in Computational Science, from the Technical University of Braunschweig.

The usage of satellites is deeply ingrained within our modern life. Be it in communication solutions like satellite TV, navigation applications like GPS or earth observation for accurate weather predictions - all these services would not exist or be severely degraded without a fleet of operational satellites. At the same time, as shown in figure 1, our use of and reliance on space based systems is accelerating: While the United Nations Office for Outer Space Affairs (UNOOSA) reported around 150 yearly new satellites between 1960 and 2012, that number exploded to 2588 in 2023! [Out24] This increased reliance on satellite based services increases the need for strong cyber security as well. Currently these systems are out of scope of most adversaries - terrestrial targets are often still easier or more valuable. But this is starting to change, both in the domain of hybrid warfare as well as with commercial hackers like ransomware groups. Militaries the world over deem satellites as valid targets, even commercial constellations like Viasat in the hours before the Russian invasion of Ukraine in February 2022. As the value generated by these commercial constellations increases we're certain to see an increased interest by non governmental players in pursuit of monetary gains as well. [Erw25] [FT22] [Pee22]

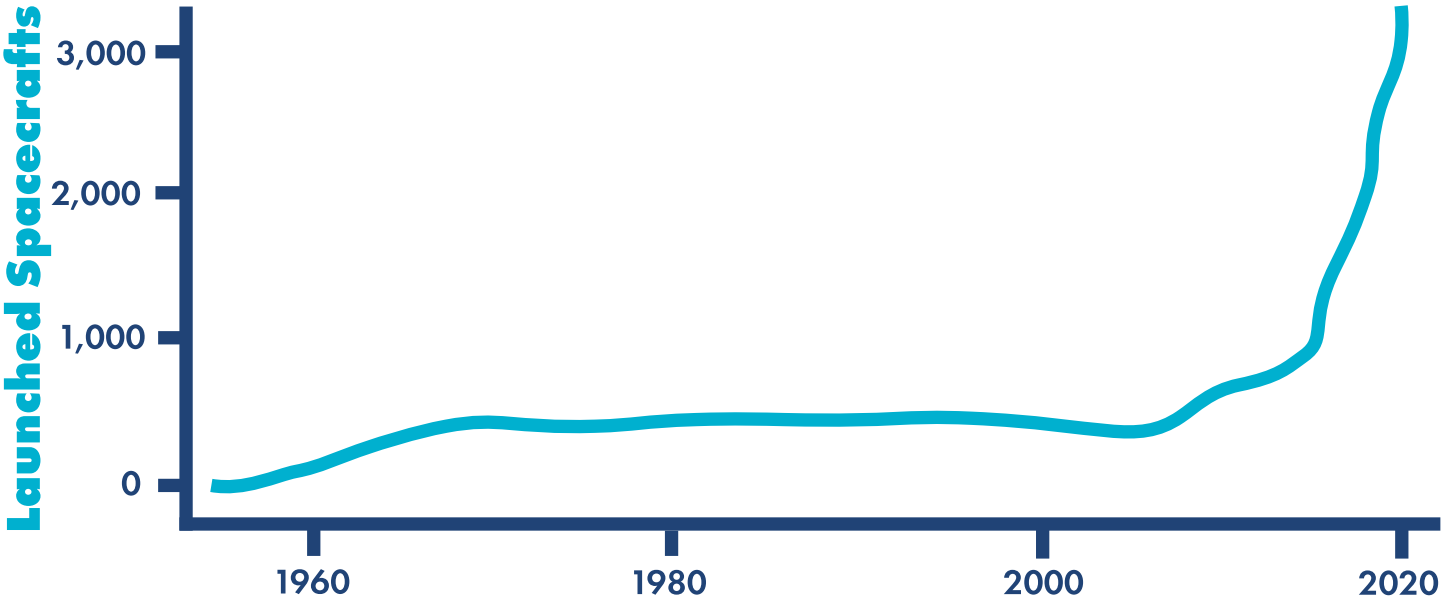

Figure 1: Number of launched satellites per year since 1957 [Out25]

## 1. Current state of satellite security

The level of cyber security on satellites is not uniform, it varies widely from mission to mission. While military and surveillance satellites, as well as spacecrafts for human spaceflight and other large institutional (scientific) satellites have used security-by-design principles for the longest time, this has often not been the case for smaller or commercial satellites. The whole topic of cyber security for spaceborne systems has not received interest of the industry or scientific community for a long time. This is starting to change now and the first scientific studies paint a worrying picture: For their 2023 IEEE paper [Joh+23] Johannes Willbold et al. conducted a survey of companies, academic institutions, and government as well as international institutions that are involved within the satellite development cycles[1]. In it only 53% of respondents stated that any defences against unauthorised access by third parties exist on their satellites. But even halve of those took the position that the secrecy of the satellite commands is suffi- cient security. Only the other halve implemented proper cryptography based access control

[1] The study focusses on the security deployed on the satellite systems themselves. It doesn't cover the security of the ground based equipment, which is in general handled like any other digital (server) equipment by the operators.

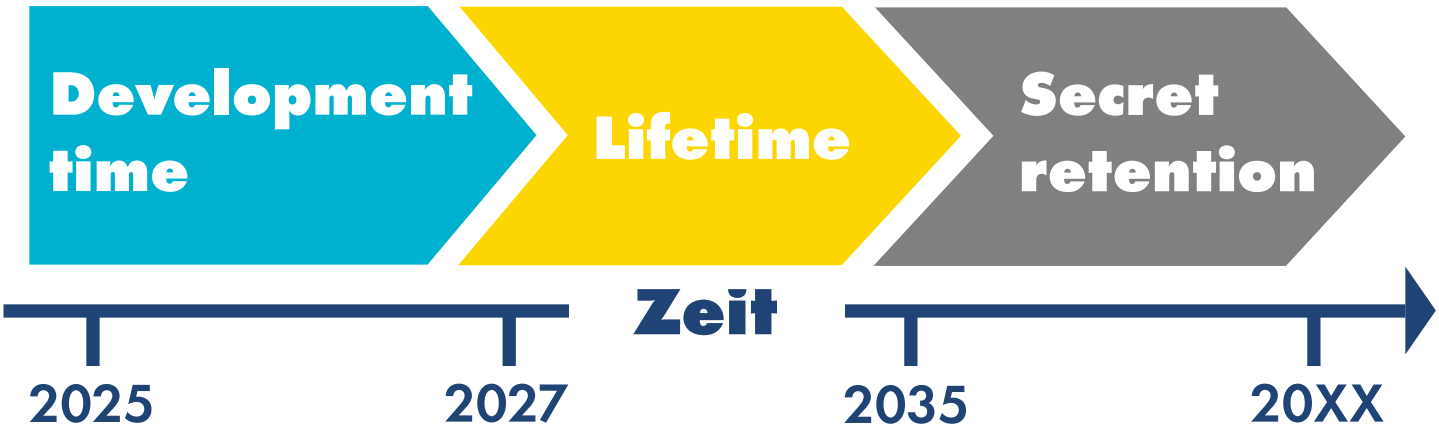


Figure 2: Development and life cycle of satellite system

on their systems. This shows the prevalence of security-by-obscurity in the industry. This is a fallacy: the protocols and commands used are often standardised and thus public knowledge. Access to powerful groundstations isn't a barrier either - radios and antennas capable of commu- nicating with satellites in low-earth-orbit (LEOS) are available for less than 2000 . [PM22]

In general all available research points in the same direction: while the ground segments are better secured because they're handled like any other server-based infrastructure, we've been lucky in regards to the space segment. Even then some attacks are happening: be it satellite TV signal hijacking for propaganda reasons or the destruction of satellite internet terminals on the eve of the Russian invasion of Ukraine in 2022, and others. [PM22] [Poi24] But terrestrial targets are often still easier to access by hackers and the importance of spaceborne systems has reached a critical state only within the last 15 years, thus haven't been in the public eye for much time yet. In addition any attack on this critical infrastructure will be investigated with intense scrutiny, increasing the risk for non state level threat actors. Still, the trend is clear: the number of cybersecurity incidents related to space systems is steadily increasing. The mindset of security-by-obscurity has affected the various standardis- ation authorities within the space domain as well. Organisations like MITRE have only indicated the importance of cybersecurity for space systems but so far have not provided guidance or standards for this domain. [Fal18] The same applies to the most common communication protocol standards. Here the Council of the Consultative Committee for Space Data Systems (CCSDS) has only added security layers as optional part of the protocols in 2015. [CCS22] It also recommends not using key exchange mechanisms, which are standard practice for any secured terrestrial application. Instead CCSDS is recommending static pre-shared keys, again citing the difficult ac- cess to satellites as reason. [CCS11] But the more data is encrypted using the same key, the higher the probability for an attacker to deduce the key out of the data he listened to. [KG16] Berlin Space Technologies has long been aware of these threats and falla- cies, and thus deploys transport security (encryption and authentication) as well as a key exchange mechanism on all their platforms since their second project. These are standardised and widely used cryptographic algorithms, also used e.g. for securing internet traffic. Here we've been at the forefront for systems of newspace companies, e.g. deploying key exchange mechanisms in- stead of static keys earlier than most of the industry. Additional second-layer defence mechanisms like access control levels are in the works for platforms destined to be used on satellite-as-a-service (SaaS) projects. But all these security mechanisms rely on the confidence in the key ex- change mechanism. Here a fundamental threat is now appearing for the whole cyber security domain: Quantum computers. Traditional key exchange algo- rithms are asymmetric cryptographic schemes that rely on the impracticality of reversing ab mod p = c when large pri-

me numbers are used. But the quantum computers in development today will be able to solve loga c efficiently for large prime numbers, as soon as the early 2030s. [Sic25] This poses a pressing problem for the satellite industry: as shown in figure 2, if a system starts development today and is launched 2 years later, then one has to expect the availability of a usable quantum computer within its lifetime (average 7 years). But even after its demise, its data might need to remain secure; attackers that recorded data in the past shouldn't be able to decode it later for some time. Thus we have to deploy post-quantum-secure cryptography on our systems today, even without guidance by the space standardisation authorities like CCSDS.

## 2. Communication in the space environment

Spacecrafts are in essence embedded systems: They have limited processing power, limited storage sizes[2] and don't provide direct (physical) user access. Thus their software has to be highly optimised and can only offer a fraction of the processing speed of modern office computers. A few more special circumstances have to be observed when designing any communication related procedure or feature for spacecrafts:

***Asymmetric Channels*** Communication between spacecrafts and their ground- station(s) is done via radio. Typically the telecommand[3] channel is slower by several orders of magnitude than the telemetry channel.

***Intermittent Connectivity*** Typically a spacecraft can only communicate with a groundstation via radio signals when they are within line of sight. In the popular (polar) low earch orbit (LEO) each of these connections (passes) will only last a few minutes. A few passes will happen in shorter succession every halve day.

***Propagation Delay*** Spacecrafts in LEO are between 200km and 3000km (most around 500km - 800km) , while those on geostationary orbit are 35.768km distance above the ground. When the spacecraft is not di- rectly overhead then the distance between it and the groundstation will be even larger. Communicating over these large distances, combined with the data conversions between radio signals and digital results in a higher propagation delay than typical terrestrial applications.

***Remote Location*** Spacecrafts are by their very nature remote and inac- cessible to direct manual manipulation. With physical access being impossible care has to be given to prevent loss of access (e.g. because of data corruption loss of the crypto key).

***Noisy Signals*** The relative low power radio signals are susceptible to signal noise over the long distances between groundstation and spacecraft, resulting in data corruption or loss. Other radio sources, terrestrial (cities, mobile phones, etc) or spaceborne (e.g. nearby satellites or solar storms) will lead to varying error rates within the signal.

**KEM:**
**A triple of algorithms (KeyGen, Encaps, Decaps)**

## 3. Cryptography Basics

To construct the current and future cryptosystems employed by BST we make use of several so-called cryptographic primitives. One can think of them as the building blocks out of which we construct a secure system.

### 3.1. Block Ciphers

Block Ciphers are pseudorandom functions[5] . A Block Cipher consists of two functions Encrypt and decrypt. Each of these Messages receives a shared secret key K and the (encrypted) message as inputs. A Block Cipher is defined by the following basic equation

***Decrypt (Encrypt (m, K), K) = m.***

BST uses the Advanced Encryption Standard (AES) with

[2] Higher processing power and larger storage are directly linked to power higher require- ments. This in turn would require larger batteries and solar cells, increasing the costs to launch the system.
[3] Uplink from groundstation to the spacecraft
[4] Downlink from spacecraft to the groundstation
[5] Pseudorandom: deterministic but nearly impossible to predict without knowing the input parameters

a key size of 256bit to encrypt all communication. Usually Block Ciphers are rather simple func- tions and thus very efficient. More and more they are implemented in hard- ware instead of software. Since Block Ciphers encrypt a fixed size message into a fixed size ciphertext it is rather insecure to encrypt a message piece by piece. This approach is called Electronic Code Book (ECB). Instead one relates the different blocks to one another. BST uses the Galois Counter Mode (GCM) See figure 3 for a comparison of the two modes.

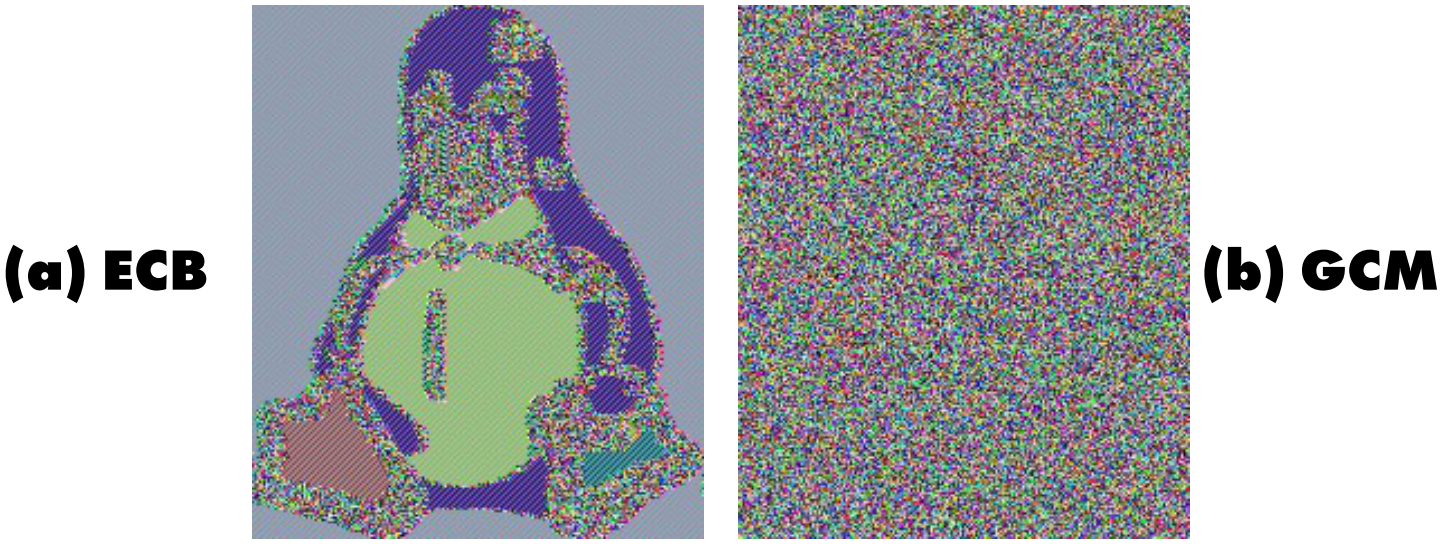


Figure 3: Identical image encrypted using the ECB and GCM modes

### 3.2 Diffie Hellman Key Exchange

To ensure both parties have the same secret key one can either distribute the keys pre mission through a secure channel. or perform a cryptographic key exchange via an unsecured channel. The drawback of the first approach is that a compromised of the system is non recoverable. Furthermore keeping keys around for a long time increases the likelihood of them being broken.cA solution to these problems is the Diffie Hellman Key Exchange. Both parties openly share information and generate a key using these open and their (distinct) secret information. There are several approaches to imple- ment Diffie Hellman. One can either use finite arithmetic using prime num- bers or points on elliptic curves. BST first used prime numbers but is now switching over to elliptic curves due to the better performance and higher se- curity they offer. The figure 4 displays the steps involved in a Diffie Hellman Key Exchange.

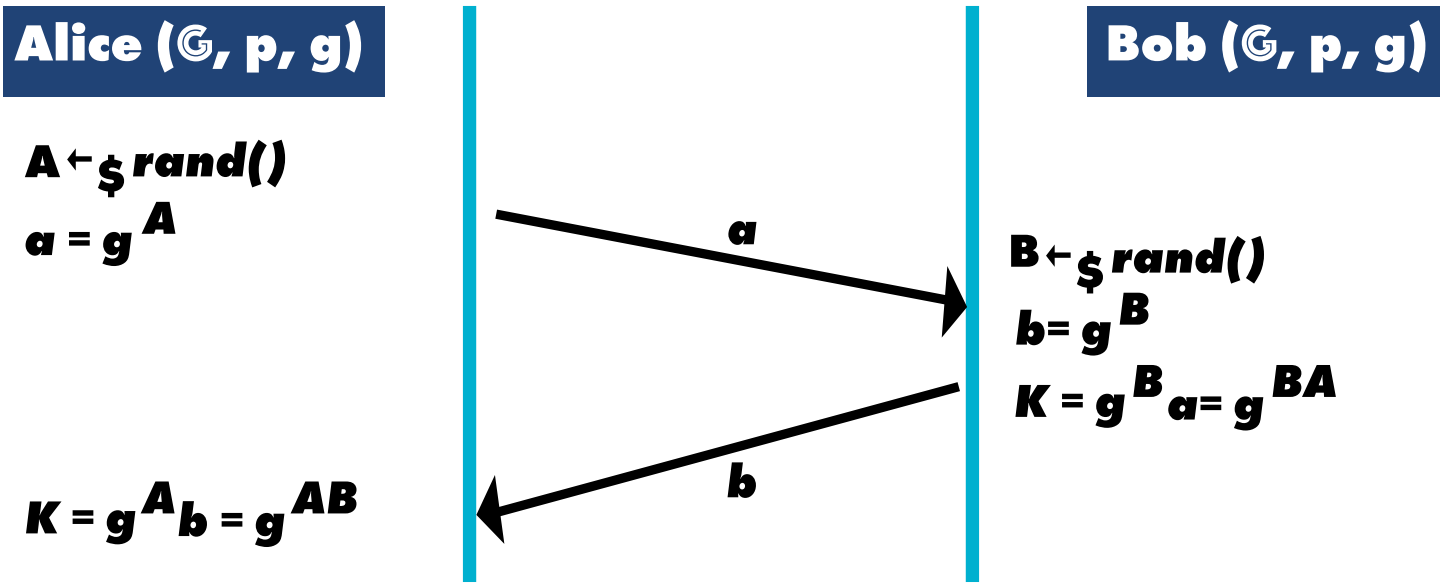


Figure 4: Diffie Hellman Key Exchange between Bob and Alice

### 3.3. Hash Functions

Hash functions are pseudorandom functions that take any input and output a fixed length of pseudorandom noise. In BSTs architecture they are used mostly for deriving a secret key out of raw material. Most modern Hash functions fall in one of two categories.

- Merkle-Damg˚ard Construction: The input message is split into blocks of a fixed length. A pseudorandom function is applied iteratively to each block. The final value is calculated using a truncation function.

- Sponge Construction: The input message is again split into blocks of a fixed length. During each iteration a state vector internal to the hash function, the sponge, is permuted according to the absorbed data. Finally a fixed size of data is squeezed from the internal state.

BST uses Hashing Algorithms of both types. Namely the SHA-2 and SHA-3 families of hashing algorithms standardised by NIST.

## 4. How Quantum Computers Attack Cryptography

Two Algorithms have so far been identified for quantum computers that threaten (BSTs) current cryptography. As of time of writing no quantum computer has been constructed that can efficiently run these algorithms. Nonetheless we can already estimate how exactly cryptographic primitives may be attacked and design future cryptographic systems accordingly.

### 4.1. Grovers Iteration

First described by the American Mathematician Lov K. Grover in 1996 [Gro96] this algorithm can be used to speed up non structured search by a factor of $\sqrt{N}$ . Essentially a non structured search is brute forcing finding a key. Imagine there are 1024 possible keys. If we just try each possible key at a time we would expect to find the correct key after around 500 steps. Using grovers iteration we would expect to find the key after around 25 steps; figure 5 illustrates this scale.

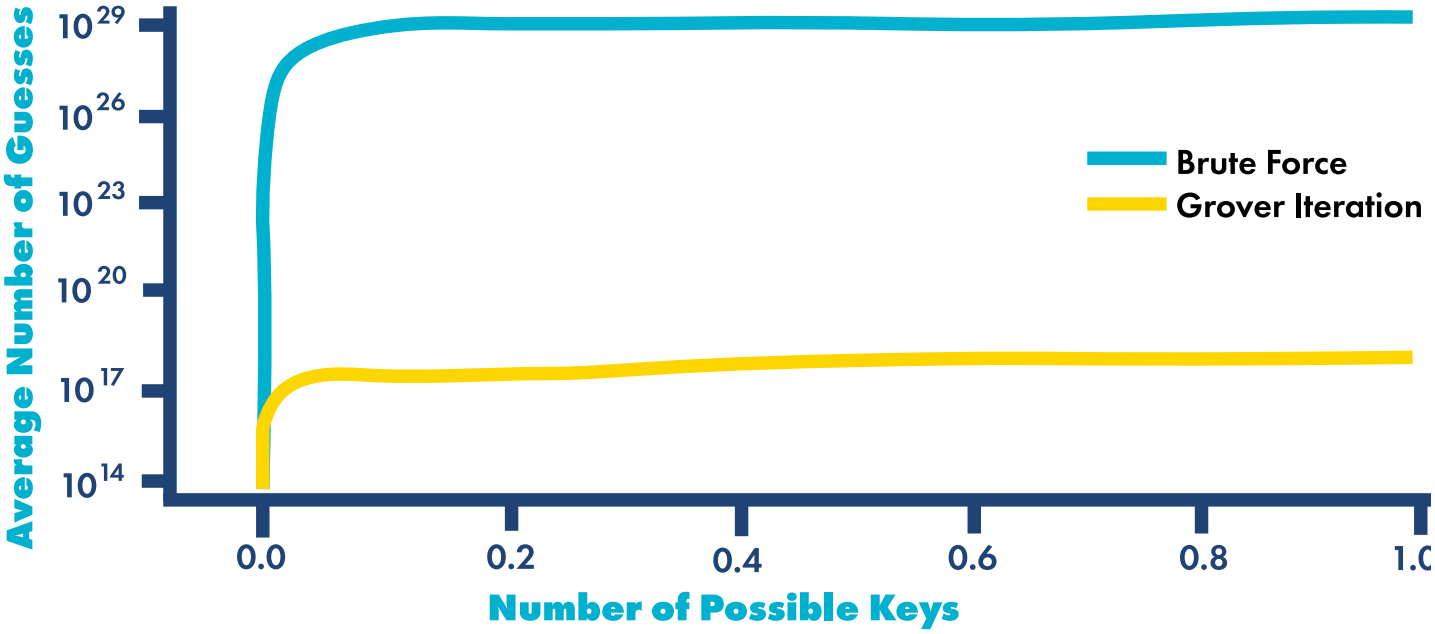


Figure 5: Speedup of Grover Iteration vs Brute Force

### 4.2. Shors Algorithm

Shors Algorithm was first published by American Mathematician Peter W. Shor in 1994 [Sho94]. It can be used to break the Hard Logarithm Problem associated with the Diffie Hellman Key Exchange. Shors Algorithm provides a significant speed-up compared to the currently fasted known factorization algorithm (Field Number Sieve), as indicated in figure 6.

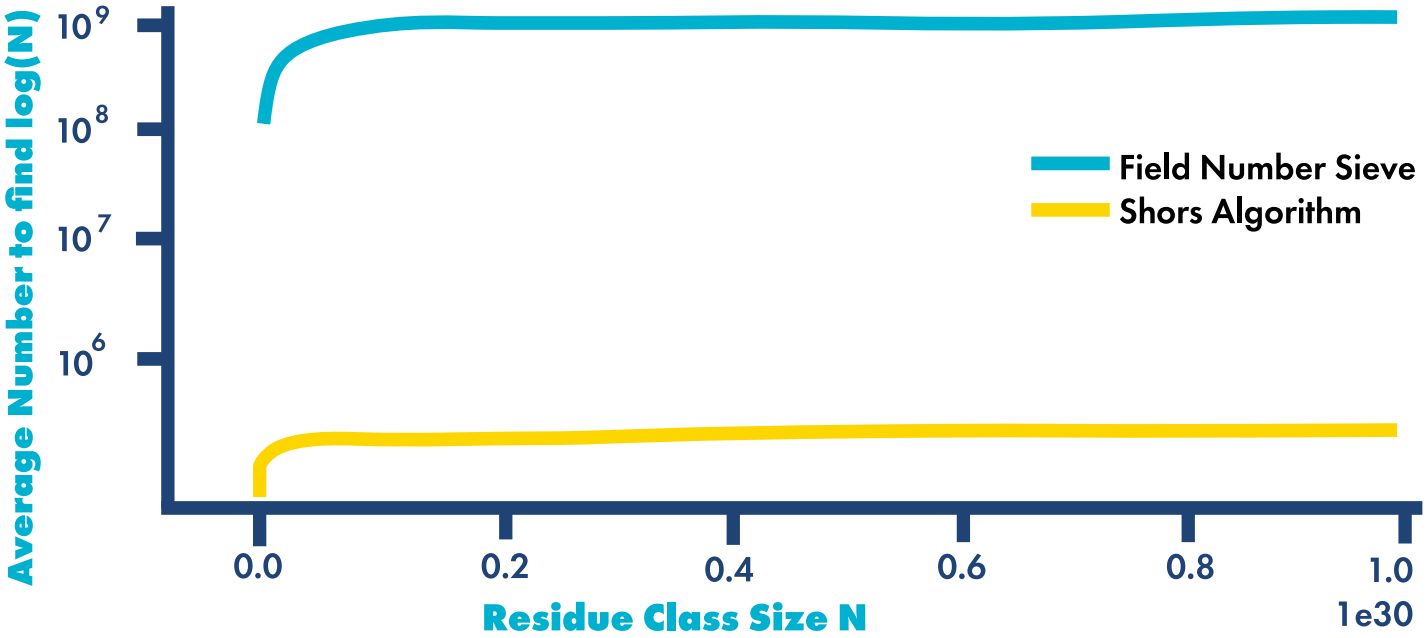


Figure 6: Field Number Sieve vs Shors Algorithm

### 4.3. What Now?

Grovers Algorithm can be used to attack Hashing Algorithms as well as Block Ciphers. However since the speed-up is "only" $\sqrt{N}$, doubling the key size aswell as retiring short hashes is actually enough to combat the risk. Shors algorithm in turn presents a much bigger challenge for modern cryptography. Virtually all key exchanges and digital signature schemes in use today rely on elliptic curves. The speed up accomplished by using Shors Algorithm is enough to completely obliterate any security. See table 4.3 for an overview of the consequences a quantum computer would have for BSTs cryptography.

| Algorithm | Type | Purpose | Impact of CRQC Longer |
|---|---|---|---|
| AES | Symmetric | Encryption | Keys needed Larger |
| SHA-2 | Symmetric | Hashing | Output needed |
| RSA | Asymmetric | Digital Signature | No longer secure |
| Diffie-Hellmann | Asymmetric | Key Establishment | No longer secure |

## 5. Quantum Proofing BST Satellites

### 5.1. Key Encapsulation Mechanisms (KEM)

In 2017 NIST started a competition to find a replacement for elliptic curves which is quantum proof. It was decided to use a new kind of cryptographic primitive for key exchange purposes. A Key Encapsulation Mechanism is a special type of Asymmetric Scheme (or Public Key Scheme). A KEM is a triple of algorithms KEM = (KeyGen, Encaps, Decaps), satisfying

- The key-generation algorithm receives as input a security parameter s and outputs a public key pk and a secret key sk They are both assumed to be pseudorandom.
- The Encapsulation algorithm receives as input the public key pk and outputs the ciphertext c and the shared secret shs which are assumed to be pseudorandom.

- the Decapsulation algorithm receives as input the ciphertext c and the secret key sk and returns the shared secret shs.

Several underlying mathematical problems have been studied to build such KEMs. As of 2025 two flavours of KEMs are available and (about to be) standardised by NIST.

#### 5.1.1. Code Based Schemes

First introduced in the 1970s these schemes are well studied. They are as- sumed to be very secure however have not been used before due to the large sizes of their keys. The Hamming Quasi Cyclic KEM submitted to NIST has been chosen as one of the winners of the NIST PQC competition and is going to be standardised next year.

#### 5.2.2. Lattice Based Schemes

Based on so-called hard lattice problems. Lattice Based Schemes are of great interest due to their comparatively small key sizes. The Israeli Regev showed in 2024 that given they are secure against classical computers they are also secure against quantum computers [Reg24]. Note however that they are not yet well studied.

**Spacecrafts are in essence embedded systems**

### 5.2. The Hybrid Approach

Since Elliptic Curve Cryptography can no longer be considered secure once a Quantum Computer goes online we should no longer rely on it as a sole means of encryption. However, against classical adversaries Elliptic Curves remain a robust and well proven cryptographic scheme. This is in contrast to Post Quantum Cryptography which is still a young field of research. Thus there is not a lot of experience with these technologies and there is a considerable risk that Quantum Schemes will be broken after all (maybe even by classical computers). When designing a cryptosystem for use today we should of course take this into account. Therefore we decided to follow the approach outlined by Barbosa et al. [Bar+24], called X-Wing. The idea is to combine both classical and post quantum cryptography in such a way that if either is broken the other retains its full security.

### 5.3. Authenticated Key Exchange

Consider again the Elliptic Curve Diffie Hellmann outlined in figure 4. We imagine an attacker Eve who can intercept and alter messages sent between Bob and Alice. Eve can intercept the message A· G sent by Alice and instead send E1 · G. Bob now computes K1 = B· E1 · G and sends B· G back to Alice. Eve again intercepts the message and instead sends E2 ·G to Alice. Now Alice computes K2 = A · E2 · G. Essentially both Alice and Bob have unwittingly completed a key exchange with Eve who is in possession of both K1 and K2. This is problematic because Eve is now able to decrypt all messages sent from either Bob or Alice and reencrypt them. Bob and Alice meanwhile are unaware that their communication is compromised.

Currently BST solves this problem by signing messages using RSA. The security of RSA is also compromised by a future Quantum Computer as it is based on the Log Hardness assumption as well. There are several PQC signature schemes available and standardised for use by NIST. All of them are Lattice Based and require large amounts of data to be transmitted, a problem for the limited bandwidth available to spacecrafts (see section 2). We chose a different approach which was suggested by the authors of the original Kyber paper [Bos+18]. Instead of using signature schemes we use static keys for both Alice and Bob and a session key to authenticate both parties implicitly. We combine three KEMs to construct the Triple KEM key exchange protocol. Both Alice and Bob generate a static public/private keypair. The public keys are shared pre mission. Whenever required (ideally for every session) Alice generates a session key pair and shares the public key with Bob. Bob encapsulates both Alice's static key and the session key, while Alice encapsulates Bob's public key. The entire key exchange can be accomplished with just two messages. Alice sends Bob's encapsulated static key and the session key. Bob then decapsulates his static key and encapsulates both Alice's static key and the

session key. He then sends both these encapsulated keys to Alice who de- capsulates them. In the end both parties have the same set of three shared secrets which they can combine into a master shared secret. See figure 7 for an overview of this[Bos+18].

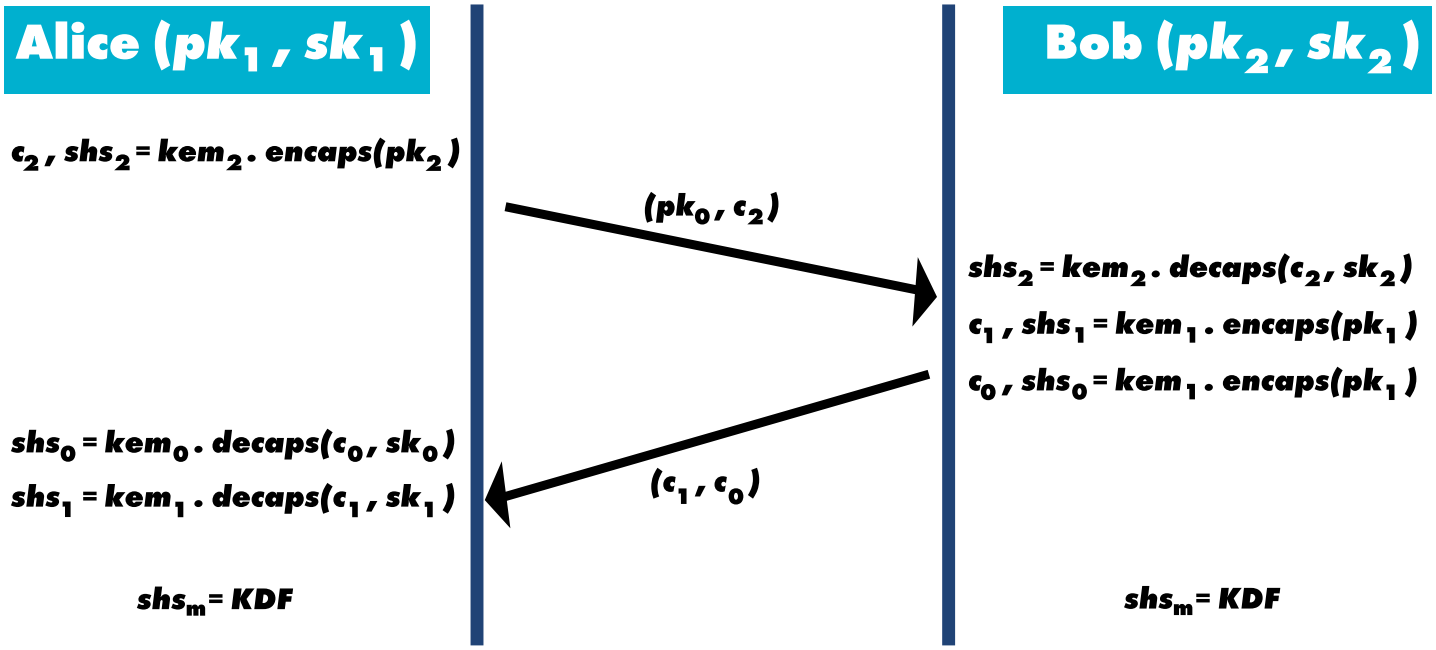


Figure 7: Triple KEM Key Exchange

For the key derivation we need a robust KEM combiner (similar to X- Wing).[Bar+24][GHP18] It is necessary to include the ciphertexts into the Key Derivation Function (KDF) to prevent chosen ciphertext attacks. We follow a similar approach to the X-Wing construction where we do not include the entire ciphertext into the KDF but only the Elliptic Curve Part. This reduces the stack size during key derivation. Without this optimisation key derivation might fail due to the low performance of satellite CPUs.

## 6. Testing in space

### 6.1. BST Triple KEM and AFR Satellite Test

The AFR (ABA First Runner) is an earth observation satel- lite build by BST and Azista BST Aerospace (ABA). It's owned and operated by ABA, but BST is able to access it in order to test new software and procedures in orbit. A basic firmware containing the Triple KEM protocol was uploaded to an in orbit demonstrator board on the AFR satellite. The triple KEM key exchange has been verified between ground and satellite software in April 2025. The uplinking and downlinking of data was conducted via the German

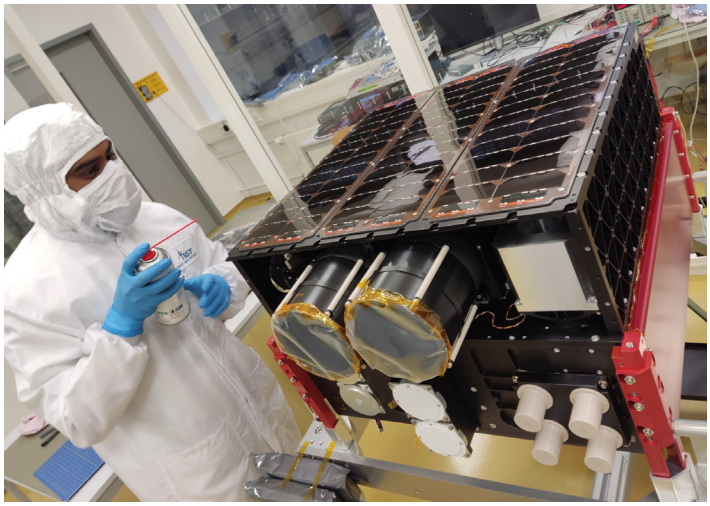

Figure 8: The AFR satellite dur- ing final assembly at BST

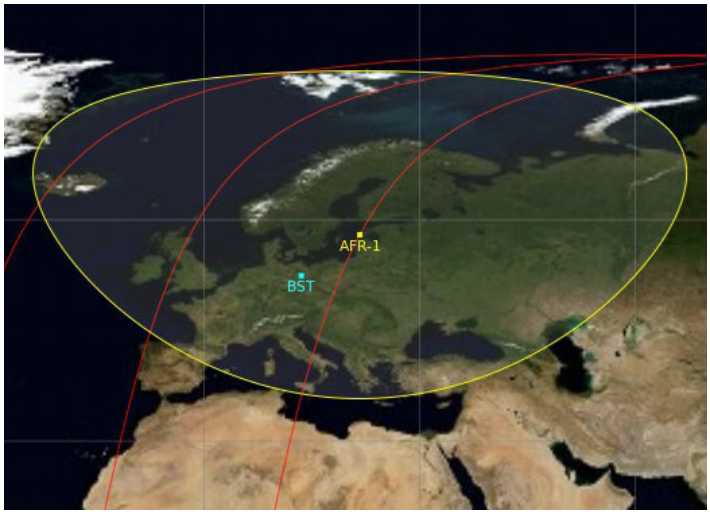


Figure 9: Orbit of the AFR satel- lite during a pass over the BST groundstation

Aerospace Center (DLR) groundstation in Neustrelitz using S-Band. AFR makes two to three passes over Neustrelitz every twelve hours. Each pass is between seven to eleven minutes long with 90 minutes between each one.

There are several code based algorithms available. See table 6.1 for sizes in bytes of the cryptographic data that needs to be stored and transmitted.

| Algorithm | Hardness | Public Key | SecretKey | Cypher |
|---|---|---|---|---|
| ML-KEM768 [Sta24] | Lattice | 1184 | 2400 | 1088 |
| HQC-192 [al20] | Code | 4522 | 4586 | 8978 |
| BIKE-L3 | Code | 24659 | 3602 | 40973 |
| Classic McEliece6960119 [al22] | Code | 1047319 | 13948 | 194 |
| ECDH (classical) | DLog | 32 | 32 | – |

We chose HQC-192 and ML-KEM768 for our implementation. We tested triple KEM with three X-Wing Hybrid KEMs built from Curve25519 [Ber06] and ML-KEM768. We also verified the protocol using HQC-192 using a wired connection. The structure of the Triple KEM implementation and the necessary commands can be seen in figure 7.

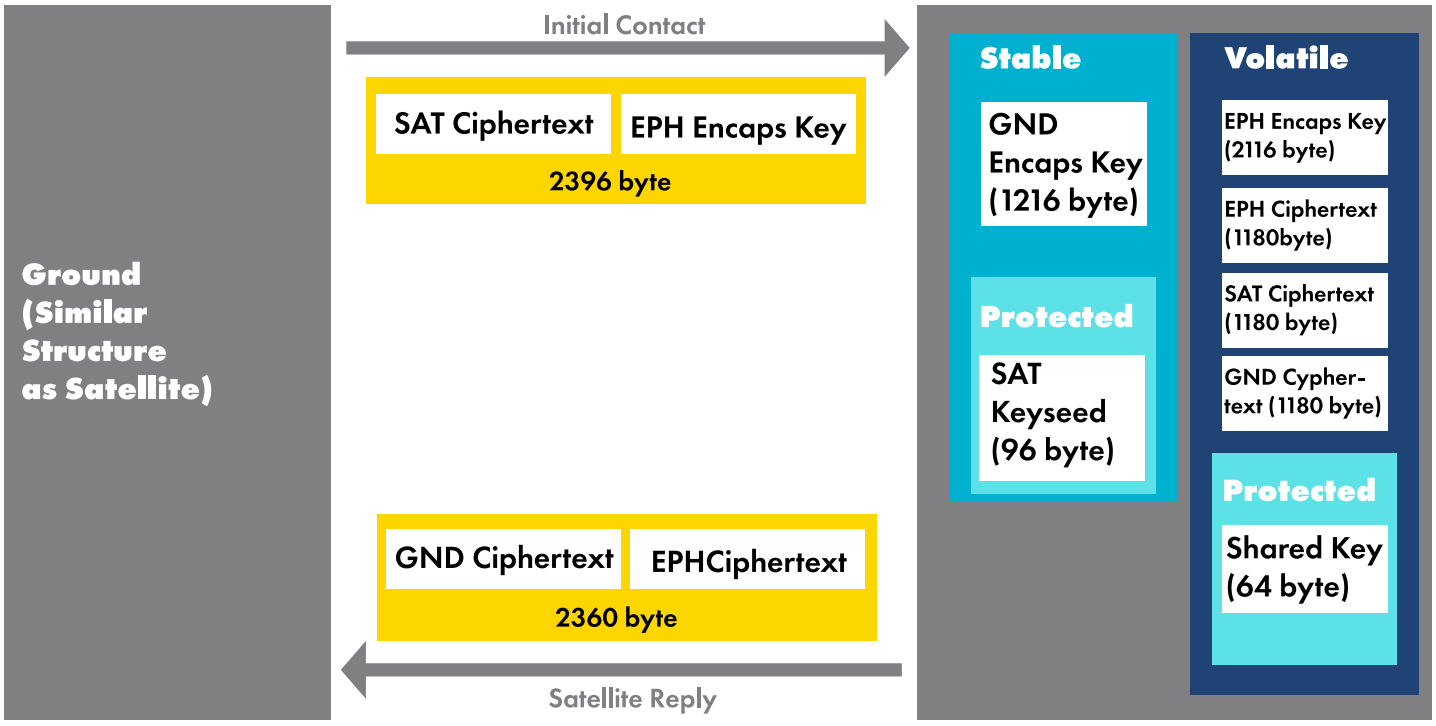


Figure 10: BST tailored Triple KEM showing commands and their sizes as well as the hierarchy of data in satellite firmware

The computation time of the cryptographic operations on the satellite was lower than the round trip time of communication with the satellite (< 500ms).

## 6.2. Conclusions and Future Developments

Since BSTs satellite computers are not performance bound regarding post quantum cryptography it was decided to

migrate the system to session keys. This means everytime we contact the satellite new keys are generated auto- matically. This provides greater protection in case keys are leaked and makes it harder to compromise keys in the first place. In addition to that we are developing a system which allows us to define access level rights. Intuitively certain commands (e.g. controlling the satellite propulsion or writing into the memory) should only be allowed to be run by privileged users (i.e. the owners of the satellite) while other commands (e.g. using the payload, requesting satellite telemetry) can also be executed by third parties who use the satellite as a service. We designed our software in such a way that it is easy to switch around the PQC algorithms in use. This cryptoagile approach allows us to respond quickly and seamlessly in case a weakness were discovered in one of the PQC algorithms. Lastly, one might wonder why we even bother with post quantum cryp- tography if there is a secure key exchange mechanism using Quantum Key Distribution (which does not require a quantum computer). The problem with this approach is that it is not very reliable. It requires many attempts until it succeeds. Furthermore it requires specialised hardware which is not even commercially available (and won't be for the forseeable future). PQC algorithms run on standard hardware and are very reliable. As our test proves it can be deployed right now.

## References


[al20]
Carlos Aguilar Melchor et al. Hamming Quasi-Cyclic (HQC).
ISAE Supaero. Jan. 2020.

[al22]
Daniel J. Bernstein et al. Classic McEliece: conservative code- based cryptography: cryptosystem specification. Oct. 2022.

[Bar+24]
Manuel Barbosa et al. X-Wing: The Hybrid KEM You've Been Looking For. Cryptology ePrint Archive, Paper 2024/039. 2024.

[Ber06]
Daniel J. Bernstein. "Curve25519: new Diffie-Hellman speed records".
In: Proceedings of the 9th International Conference on Theory and Practice of Public-Key Cryptography. PKC'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 207–228.

[Bos+18]
Joppe Bos et al. "CRYSTALS – Kyber: a CCA-secure module- lattice-based KEM". In: 2018 IEEE European Symposium on Se- curity and Privacy. NXP Semiconductors. 2018, pp. 353–367.

[CCS11]
CCSDS. Space Mission Key Management Concept. Green Book. CCSDS 350.6-G-1. 1. CCSDS Secretariat, Nov. 2011.
https://ccsds.org/wp-content/uploads/gravity_forms/5- 448e85c647331d9cbaf66c096458bdd5/2025/01//350x6g1.pdf.

[CCS22]
CCSDS. Space Data Link Security Protocol. Blue Book. CCSDS 355.0-B-2. 2. CCSDS Secretariat, July 2022.
url: https://cc sds.org/wp- content/uploads/gravity_forms/5- 448e85c 647331d9cbaf66c096458bdd5/2025/01//355x0b2.pdf.

[Erw25]
Sandra Erwin. GPS disruption and satellite maneuvers now hall- marks of modern warfare. Apr. 2025.
url: https://spacene ws.com/gps- disruption- and- satellite- maneuvers- now- hallmarks-of-modern-warfare/.

[Fal18]
Gregory Falco. "The Vacuum of Space Cyber Security". In: Sept.2018. doi: 10.2514/6.2018-5275.

[FT22]
Foreign, Commonwealth & Development Office and The Rt Hon Elizabeth Truss. Russia behind cyber-attack with Europe-wide im- pact an hour before Ukraine invasion. May 2022.
url: https://www.gov.uk/government/news/russia-behind-cyber-att ack-with-europe-wide-impact-an-hour-before-ukraine- invasion.

[GHP18]
Federico Giacon, Felix Heuer, and Bertram Poettering. KEM Com- biners. Cryptology ePrint Archive, Paper 2018/024. 2018.

[Gro96]
Lov K. Grover. "A fast quantum mechanical algorithm for database search". In: Proceedings of the Twenty-Eighth Annual ACM Sym- posium on Theory of Computing. STOC '96. Bell Labs. New York, NY, USA: Association for Computing Machinery, 1996, pp. 212– 219.

[Joh+23]
Johannes Willbold et al. "Space Odyssey: An Experimental Soft- ware Security Analysis of Satellites". In: 2023 IEEE Symposium on Security and Privacy (SP). 2023, pp. 1–19. doi: 10.1109/ SP46215.2023.10351029.

[KG16]
Karthikeyan Bhargavan and Ga¨etan Leurent. Sweet32: Birthday attacks on 64-bit block chiphers in TLS and OpenVPN. Aug. 2016.
url: https://sweet32.info/.

[Out24]
United Nations Office for Outer Space Affairs. 2023 Annual Re- port. Government Report. June 2024.
url: https : / / www . un oosa . org / res / oosadoc / data / documents / 2024 / stspace / stspace90_0_html/UNOOSA_Annual_Report_2023.pdf.

[Out25]
United Nations Office for Outer Space Affairs. Online Index of Objects Launched into Outer Space. May 2025.
url: https:// www.unoosa.org/oosa/osoindex/search-ng.jspx?lf_id=.

[Pee22]
Walter Peeters. Cyberattacks on Satellites: An Underestimated Political Threat. 2022.
url: https://www.lse.ac.uk/ideas/ projects/space- policy/publications/Cyberattacks- on- Satellites.

[PM22]
James Pavur and Ivan Martinovic. "Building a launchpad for satellite cyber-security research: lessons from 60 years of space- flight".
In: Journal of Cybersecurity 8.1 (June 2022), tyac008. issn: 2057-2085. doi: 10.1093/cybsec/tyac008.
eprint: https://academic .oup .com /cybersecurity/article- pdf /8 /1 / tyac008/50476484/tyac008.pdf. url: https://doi.org/10.1093/cybsec/tyac008.

[Poi24]
Cl´emence Poirier. "Hacking the Cosmos: Cyber operations against the space sector. A case study from the war in Ukraine". In: CSS Cyberdefense Reports (2024). doi: 10.3929/ethz-b-000697348.
eprint: https://ethz.ch/content/dam/ethz/special-intere st/gess/cis/center-for-securities-studies/pdfs/cyber- reports- 2024 - 10 - hacking- the- cosmos .pdf.
url: https://doi.org/10.3929/ethz-b-000697348.

[Reg24]
Oded Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. Jan. 2024.

[Sho94]
Peter W. Shor. "Algorithms for Quantum Computation: Discrete Logarithms and Factoring". In: Proceedings 35th Annual Sym- posium on Foundations of Computer Science. Bell Labs. 1994,pp. 124–134.

[Sic25]
Bundesamt für Sicherheit in der Informationstechnik. Status of quantum computer development. BSI TR-02102-1. 2.1. Jan. 2025.
url: https://www.bsi.bund.de/SharedDocs/Downloads/DE/ BSI/Publikationen/Studien/Quantencomputer/Entwicklung stand_QC_V_2_1.pdf? blob=publicationFile&v=3.

[Sta24]
National Institute of Standards. FIPS 203, Module-Lattice-Based Key-Encapsulation Mechanism Standard. National Institute of Stan- dards and Technology (NIST). 100 Bureau Drive Gaithersburg, MD 20899, USA, Aug. 2024.