














Use-Case Denial of Service Attack on Actual Quantum Key Distribution Nodes

Patrik Burdiak¹^a, Emir Dervisevic²^b, Amina Tankovic²^c, Filip Lauterbach¹^d, Jan Rozhon¹^e,
Lukas Kapicak¹^f, Libor Michalek¹^g, Dzana Pivac²^h, Merima Fehric²ⁱ, Enio Kaljic²^j,
Mirza Hamza²^k, Miralem Mehic²^l and Miroslav Voznak¹^m

¹Department of Telecommunications, Faculty of Electrical Engineering and Computer Science, VSB – Technical University of Ostrava, 17. listopadu 2172/15, 708 33 Ostrava, Czech Republic

²Department of Telecommunications, Faculty of Electrical Engineering, University of Sarajevo, Zmajca od Bosne bb, 71000, Sarajevo, Bosnia and Herzegovina

Keywords: QKD, KMS, Suricata IPS/IDS, DoS, DDoS.

Abstract: QKD integration into traditional telecommunication networks is anticipated in the upcoming decades in order to maintain adequate levels of communication security. QKD establishes ITS (Information-Theoretic secure) symmetric keys between the two parties, which they may use to sustain secure flow of data even in the post-quantum era. Since QKD-keys are a valuable and scarce resource, they must be carefully maintained. This paper investigates DoS attacks on actual QKD equipment, in which an adversary with access to QKD services depletes the reserves of QKD-keys maintained at the KMS system. As a result, safety precautions are proposed in order to prevent this scenario and maintain operational QKD service.

1 INTRODUCTION


Quantum key distribution (QKD) is an emerging technology that has matured sufficiently to be considered for integration in traditional telecommunication networks. QKD, unlike any other technology, provides a long-term solution to secret key agreements that is not compromised by new advances in (quantum) computing (Bennett and Brassard, 1984). When used correctly with quantum-resistant symmetric cryptosystems, it ensures secure data transmission even in quantum computing environments. A recent


area of research regarding the integration and application of QKD within telecommunication networks is concerned with its security against traditional network attacks, such as DoS (Denial of Service) attacks, which would render the technology inoperable (Dervisevic et al., 2022). This study provides a solution to a specific DoS attack on the Key Manager System (KMS) component, which is critical to the operation of QKD technology. The solution is evaluated in real-world environment using commercially available QKD equipment and Suricata IPS/IDS (Intrusion Prevention and Detection System) services.


This paper is organized as follows: Section 2 is describing current state of the art, in Section 3 is described essential parts of QKD system, Sections 4 and 5 are focusing on test bed environment and attacking scenario. In Section 6 reader can find proposed security measurements that could be implemented by various technologies and Section 7 is representing results of our experiment. (Mehic et al., 2022b).


2 STATE OF THE ART


Currently most attacks on QKD systems were performed on quantum channel (Hugues-Salas et al.,


^a <https://orcid.org/0000-0002-9739-9278>


^b <https://orcid.org/0000-0002-7981-7739>


^c <https://orcid.org/0000-0001-8570-8339>


^d <https://orcid.org/0000-0002-0176-1288>


^e <https://orcid.org/0000-0003-4768-6073>


^f <https://orcid.org/0000-0002-4310-5123>


^g <https://orcid.org/0000-0002-1117-5477>


^h <https://orcid.org/0000-0001-5368-5582>

ⁱ <https://orcid.org/0000-0001-7813-535X>

^j <https://orcid.org/0000-0003-1902-2608>

^k <https://orcid.org/0000-0002-1084-232X>

^l <https://orcid.org/0000-0003-2697-1756>

^m <https://orcid.org/0000-0001-5135-7980>

2018; Li et al., 2018; Dai et al., 2022). These studies are mainly focused on fiber based QKD systems and vulnerabilities of transmission of qubits within optical fibers. In (Hugues-Salas et al., 2018), a Software Defined Networking (SDN) application is developed to perform real-time monitoring of quantum channel performance, i.e., Secret Key Rates (SKR) and Quantum Bit Error Rates (QBER), which can be used to detect DDoS attacks. If the DDoS attack is detected, the SDN application relocates the quantum channel to one of the spare fiber links. Article (Li et al., 2018), authors are proposing DoS attack based on manipulation of quantum channel. As result even slight manipulation of the channel parameters is leading to underestimation of the secure communication and to intentional termination of communication. In (Dai et al., 2022), authors are focusing on detection of LDoS (low-rate denial-of-service) in the CV-QKD (continuous-variable QKD) communication progress. This detection is performed on quantum channel and focused on parameters of this channel. Articles (Dervisevic et al., 2022; Mehic et al., 2022a), are pioneering research on DoS attacks on KMS. Article (Dervisevic et al., 2022) will be more described in Chapter 5. This article focuses on the same problem but with real-time devices.

3 ESSENTIAL PARTS OF QKD SYSTEM

QKD technology establishes symmetric key material in ITS (Information-Theoretic Security) secure manner between the two distant sites. This key material is established via QKD link. QKD link is divided into two separate channels, public channel and quantum channel. Purpose of quantum channel is propagation of photons where keys are encoded into quantum states of this photons. Laws of quantum physics guarantees, that this channel is secure from eavesdropper by non-cloning theorem and Heisenberg principal (Vagenas et al., 2019). Public channel is used for post-processing applications, connection between QKD nodes via public channel can be performed by public internet. Access from public internet means that QKD node on his public channel interface is exposed to attacks that are well known like Dos attack, DDos attack and so on. Every QKD protocol like BB84 (Bennett and Brassard, 1984), BB92 (Bennett, 1992), COW (Gisin et al., 2004) are using both of these channels for establishing key material. Essential part of QKD system is Key Management system.

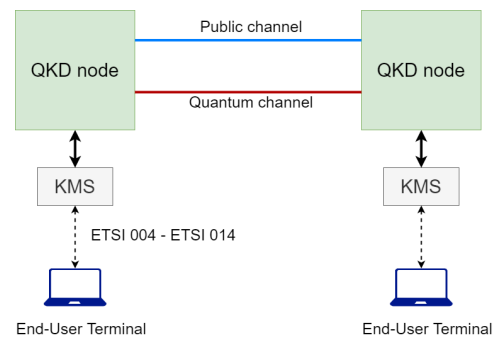


Figure 1: QKD system.

KMS is purposing for storage, management of keys and distribution of keys through QKD network. Secure Application Entity (SAE) is communicating with KMS via REST-API which is described in standards ETSI 004 and ETSI 014 (Mehic et al., 2022c). In Figure 2 we can notice that the communication between encoders/decryptors and QKD nodes takes place using the ETSI 014 standard. The ETSI 014 (ETSI, 2019) standard defines REST-API (Representational State Transfer-Application Programming Interface) for communication between QKD node that contains KMS (Key Management System) and SAE (Secure Application Entity). We can imagine SAE as a service that contains an encoder/decryptor and is used for communication with the QKD node. The REST-API uses the HTTPS protocol, the keys are transferred in JSON format together with the key identifier (ID). The key identifier serves as a pointer to the key store in the QKD node. Encryptors/decryptors exchange such an identifier. This step ensures that a key with the same ID will be used for encryption and decryption. We have SAE Alice and SAE Bob as in Figure 2.

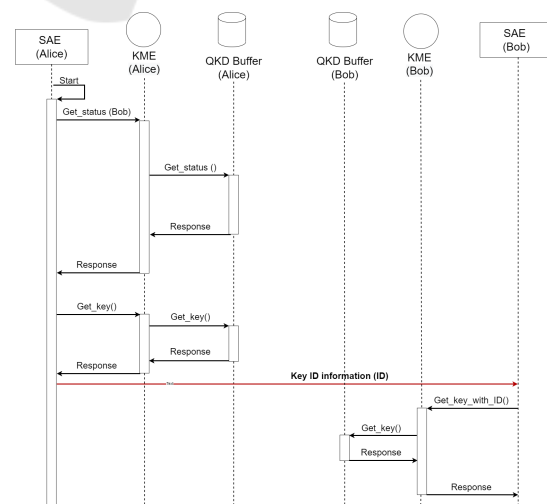


Figure 2: ETSI-014.

SAE Alice first asks her KMS using the *Get_Status(Bob)* function for status of QKD buffer where the keys are stored. The function returns information about the assigned KMS entity, the size of the key that can be delivered to the SAE, the number of keys stored in the buffer and the maximum number of keys that can be delivered in one response. Subsequently, Alice requests a block of keys (or just one key) using the function *Get_Key()*, as arguments of this function are the number, size of keys and SAE ID. The SAE ID serves as the SAE identifier. As soon as Alice receives the block of keys with the corresponding ID, she forwards this ID to Bob. Bob then requests his KMS for a block of keys with the corresponding ID from Alice using the *Get_Key_with_ID()* function. From the QKD theory, we know that for a given QKD connection, both communicating parties have the same QKD buffer, i.e. they contain the same keys. Thus, as a result, Alice and Bob have the same keys for encryption and decryption.

4 TEST BED

Our experiments were performed with *IDQ Cerberis 3* hardware located at VSB-Technical University of Ostrava. Two QKD nodes *IDQ Cerberis 3* are interconnected by quantum channel and service channel(public channel). Attenuation of quantum channel which is provided by single mode optical fiber is 10 dB. Both of this device alongside with encryptors *IDQ Centauris* are included within OpenQKD¹ project and NATO Quantum5² project.

5 ATTACKING SCENARIO

In article (Dervisevic et al., 2022), authors investigated attacking scenario where attacker is sending invalid requests to KMS and KMS must process them. With high number of requests KMS must proceed these requests in queue, queue have finite size, so attacker can full fill this queue with invalid requests and valid requests could be dropped. Authors of these article are describing next situation where attacker can deplete all key material from key storage of QKD nodes by using end-user terminal with valid certificates. Authors performed this experiment in *QKD-NetSim* (Mehic et al., 2017) simulator, by generating *Get_key* https requests on KMS in specific period of time. From this observations we can see that KMS

has no defensive mechanism against this type of attack. With higher number of requests key storage is depleted faster a this means a big security problem for stability of QKD system. When key storage's are empty, other end-users can not get their own keys for encryption. This article deals with situation where attacker has valid certificates and wants do deplete key storage in KMS, performed with real-time QKD devices.

6 SECURITY PRECAUTIONS

In this section we are proposing method or approaches which could be implemented into KMS or firewall for protecting key storage from depleting key material. We are proposing Formula 1, for threshold value. This threshold value representing maximum number of observed key in specific period of time for one user.

$$threshold = \frac{\frac{total\ amount\ of\ keys[-]}{number\ of\ valid\ end\ users[-]}}{\frac{total\ amount\ of\ key\ material[bits]}{secret\ key\ rate[bit/s]}} [keys/s] \quad (1)$$

In this formula we have to consider that all SAE served by QKD system will ask for the same length of the key. In our experiment end-users SAE are consuming 256 bit keys. In the next chapter is described use-case how can be this formula interpreted into IPS/IDS service.

6.1 Implementation in Software Based IDS/IPS

Software based IDS/IPS is representing service running on typical PC or virtual machine running on server hardware. Advantage of software IPS/IDS is that most of them are for free and you can easily install this system on already running network infrastructure. Typical represent of IDS/IPS software is **Suricata** (Waleed et al., 2022). In Figure 3, is shown position of Suricata IPS/IDS within network topology.

We used this IDS/IPS system for implementing our Formula 1. Our IDQ system has this parameters:

- Total amount of keys: 1000
- Total amount of key material: $256 \cdot 1000 = 256000bit$
- Secret key rate = $2000bit/sec$

In our experiment we consider that QKD system is operating with forty active end-user terminals. With

¹<https://openqkd.eu>

²<https://www.quantum5.eu>

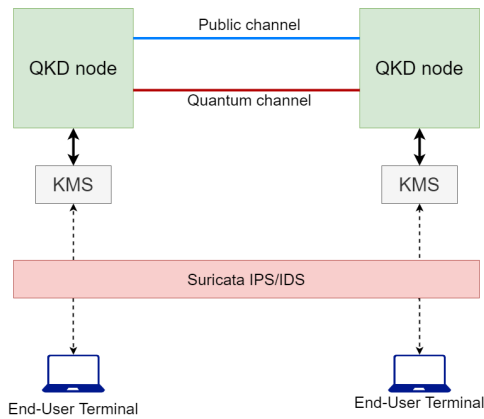


Figure 3: Implementation of Suricata IPS/IDS.

these parameters we can easily compute threshold value for our KMS.

$$threshold = \frac{1000}{\frac{40}{256000}} [keys/s] \cong 11 [keys/minute] \quad (2)$$

As result from Formula 2, we can observe that in our experiment our SAE can obtain eleven keys by minute. This keys could be used for encryption of data for any time period. That means that secure tunnel between two end-users terminals can use specific key for one minute, and the other secured tunnel can using one specific key for one hour. This time validity of key is set by end-user terminals. In this implementation, Suricata has internal rules which will not allow to get more than eleven keys by size 256 bit per minute by end-user terminal. After obtaining eleven keys per minute, this end-user terminal will not obtain new keys for specific period of time. For our experiment this time was 1024 seconds. This period represent, time for full filling key storages with key material. This period is derived from knowledge that secret key rate is 2k bit/s and total amount of key material is 256k bits.

6.2 Implementation in Hardware Devices

Security implementation with hardware-based firewall is similar to software defined firewall and IPS system. The main differences are in TCP/IP model layer-based security. Hardware firewalls are fully designed for a traffic packet real-time analysis. It reaches higher throughput compare to software defined firewall (Konikiewicz and Markowski, 2017) and better efficiency if we analyze traffic for second, third and fourth TCP/IP (Alani, 2014) model traffic. The First TCP/IP model layer security could be performed by MAC address security a port security to

ports where device is connected. We can avoid some ARP attacks etc.

The Second layer of TCP/IP model (Alani, 2014) is IP based. The basic rule for KMS security could be IP white and black listening. These rules could be pre-defined by system administrators. If we know trusted IP address list for KMS clients, it is simple to prepare rule for access to KMS keys. Otherwise, we can prepare list for non-trusted clients, which can be blocked by default. In case the KMS device is available for access through the internet, we must implement other security mechanisms to protect this device. Hardware firewall device must have protection against DOS and DDOS attacks. When this device detects multiple requests from one or multiple IP addresses and reaches preconfigured threshold, this address must be black-listed permanently, or system administrator must be notified about security incident.

The Second layer security is TCP or UDP traffic based. KMS key exchange use HTTPS traffic over TCP connection. Hardware firewall device can limit traffic on this type of traffic. Other traffic is blocked by the default. The worst scenario for hardware firewall (Krishna and Karthik, 2022) is attack performed by the trusted device which is white-listed, and it has certificates for requesting keys. KMS device provides keys for a multiple KMS clients and in case of one client deplete key buffer, other devices will not have keys for their purposes. Security for this attack scenario is not simple because it is not possible to analyze encrypted traffic directly. Direct traffic analysis is possible on non-encrypted traffic and it could be done using proxy server. Administrator must observe normal traffic load between the KMS device and KMS clients and prepare rules for TCP connections. If limit for normal load is reached, according rule the traffic for this KMS client is blocked.

In every scenario, hardware firewall must be equipped with some notification technology. Administrator must be notified about security incident minimally via e-mail, or there has to be implemented support for SNMP or RSYSLOG (Bresnahan and Blum, 2019) which allow system administrator or fully automatized system make some approaches to prevent attack. System could be distributed into two systems. For traffic analyzing and for security rules application can be used hardware firewall device, which provide better network performance and fully automatic system for determination attackers could be software based, like is depicted in Section 6.1.

7 RESULTS

In this chapter we are representing output data from experiment based solution described in Section 6.1. In this experiment we are performing attack on KMS where end-user terminal with running SAE is trying to consume all keys from KMS key buffer. Key requests were generated from software-based encryptor, this keys could be used for application which is performing encryption SAE for creating secure tunnel between two end-user terminals. Both ends of secured tunnel have to use at the same time symmetric key with same ID. Signalization about used ID is transmitting between SAE on both ends.

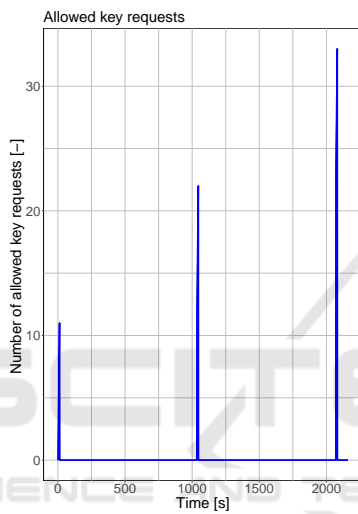


Figure 4: Allowed key requests.

In Figure 4 are shown allowed key requests from SAE captured on Suricata system in time slot of 2160 seconds. Experiment was running for 2 hours but we are choosing this specific time slot for better representation of output data. In Figure 4 reader can see three peaks. During the time of all of three peaks SAE obtains 33 keys from KMS with time difference of 1024 seconds between peaks. Period of one peak was exactly 11 seconds, every second SAE was asking KMS for new key. Formula 1 was implemented into Suricata system which allows SAE to obtain 11 keys per minute. After this threshold value was reached SAE which represented specific IP address of end-user terminal, would not obtain next keys for time period of 1024 seconds. In Figure 5 is shown detailed peak, where on y axis can be seen increasing number of observed keys for time period of 11 seconds. Very important part of this experiment is examining the behavior of implemented solution for blocking key requests.

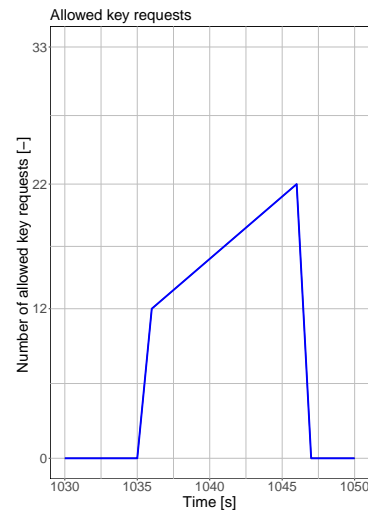


Figure 5: Detailed peak.

In Figure 6, is shown number of blocked requests during the whole experiment, from this graph is obvious that all key requests between peaks when threshold value was activated were blocked. The density of blocked requests is different due to the simulation of different amounts key requests.

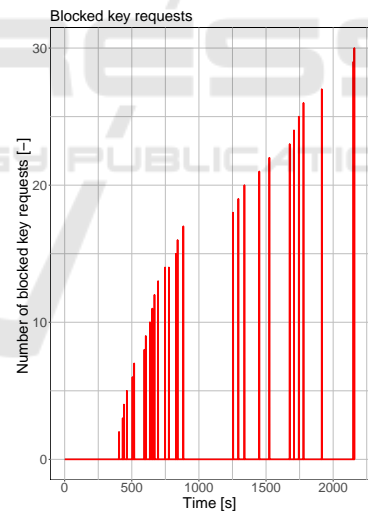


Figure 6: Blocked key requests.

8 CONCLUSION

QKD is promising area for advanced security in post-quantum era. In incoming decades we can expect that threat by quantum computers against today used asymmetric cryptography will be growing. This paper represents real-time QKD nodes and very sophisticated attack on them when attacker has a valid certificates and can very easily deplete keys from this nodes.

With the growing need for implementation QKD technology into conventional networks, arises question which security measurements should be taken. We investigate previous research about security of KMS systems. This article is describing real-time QKD nodes and DoS attack on KMS. In previous chapters we represented security measurements against DoS attack on QKD nodes. This security measurements are described within various technologies, with usage of our proposed threshold formula. From this threshold formula network administrators can easy calculate how many keys can be consumed by end-users or this formula can be implemented to protocol, service and so on. This security approach could be used when KMS is using for communication with SAE standard ETSI 014. With usage of standard like ETSI 004 which is using reservation of keys for specific period of time security measurements should be different. Subsequently we have implemented this solution to Suricata IPS/IDS service and performed experiment with actual QKD nodes. The Experiment showed, that SAE obtain only maximum amount of keys for specific time period other key requests were blocked.

ACKNOWLEDGEMENTS

The research leading to the published results was supported under the NATO SPS G894 project “Quantum Cybersecurity in 5G Networks (QUANTUM5)” and partly under the H2020 project OPENQKD and grant agreement No. 857156. The work was also supported by the Ministry of Science, Higher Education and Youth of Canton Sarajevo, Bosnia and Herzegovina under Grants No. 27-02-35-35137-29/22 and 27-02-35-35143-6/22.

REFERENCES

- Alani, M. (2014). *TCP/IP model*, pages 19–50.
- Bennett, C. H. (1992). Quantum cryptography using any two nonorthogonal states. *Physical review letters*, 68(21):3121.
- Bennett, C. H. and Brassard, G. (1984). Proceedings of the IEEE international conference on computers, systems and signal processing.
- Bresnahan, C. and Blum, R. (2019). *Implementing Logging Services*, pages 473–486.
- Dai, E., Huang, D., and Zhang, L. (2022). Low-rate denial-of-service attack detection: Defense strategy based on spectral estimation for cv-qkd. In *Photonics*, volume 9, page 365. MDPI.
- Dervisevic, E., Lauterbach, F., Burdiak, P., Rozhon, J., Slivová, M., Plakalovic, M., Hamza, M., Fazio, P., Voznak, M., and Mehic, M. (2022). Simulations of denial of service attacks in quantum key distribution networks. In *2022 XXVIII International Conference on Information, Communication and Automation Technologies (ICAT)*, pages 1–5. IEEE.
- ETSI (2019). Quantum key distribution (qkd); protocol and data format of rest-based key delivery api (etsi gs qkd 014).
- Gisin, N., Ribordy, G., Zbinden, H., Stucki, D., Brunner, N., and Scarani, V. (2004). Towards practical and fast quantum cryptography. *arXiv preprint quant-ph/0411022*.
- Hugues-Salas, E., Ntavou, F., Ou, Y., Kennard, J. E., White, C., Gkounis, D., Nikolovgenis, K., Kanellos, G., Erven, C., Lord, A., et al. (2018). Experimental demonstration of ddos mitigation over a quantum key distribution (qkd) network using software defined networking (sdn). In *Optical fiber communication conference*, pages M2A–6. Optica Publishing Group.
- Konikiewicz, W. and Markowski, M. (2017). Analysis of performance and efficiency of hardware and software firewalls. *Journal of Applied Computer Science Methods*, 9.
- Krishna, T. V. and Karthik, P. (2022). Dominance of hardware firewalls and denial of firewall attacks (case study blacknurse attack). *International Journal of Science and Research (IJSR)*, 11:28–33.
- Li, Y., Huang, P., Wang, S., Wang, T., Li, D., and Zeng, G. (2018). A denial-of-service attack on fiber-based continuous-variable quantum key distribution. *Physics Letters A*, 382(45):3253–3261.
- Mehic, M., Maurhart, O., Rass, S., and Voznak, M. (2017). Implementation of quantum key distribution network simulation module in the network simulator ns-3. *Quantum Information Processing*, 16(10):1–23.
- Mehic, M., Rass, S., Dervisevic, E., and Voznak, M. (2022a). Tackling denial of service attacks on key management in software-defined quantum key distribution networks. *IEEE Access*, 10:110512–110520.
- Mehic, M., Rass, S., Fazio, P., and Voznak, M. (2022b). Modern trends in quantum key distribution networks. *Quantum Key Distribution Networks*, pages 209–223.
- Mehic, M., Rass, S., Fazio, P., and Voznak, M. (2022c). Quality of service architectures of quantum key distribution networks. In *Quantum Key Distribution Networks*, pages 73–107. Springer.
- Vagenas, E. C., Farag Ali, A., and Alshal, H. (2019). Gup and the no-cloning theorem. *The European Physical Journal C*, 79(3):1–5.
- Waleed, A., Jamali, A. F., and Masood, A. (2022). Which open-source ids? snort, suricata or zeek. *Computer Networks*, 213:109116.