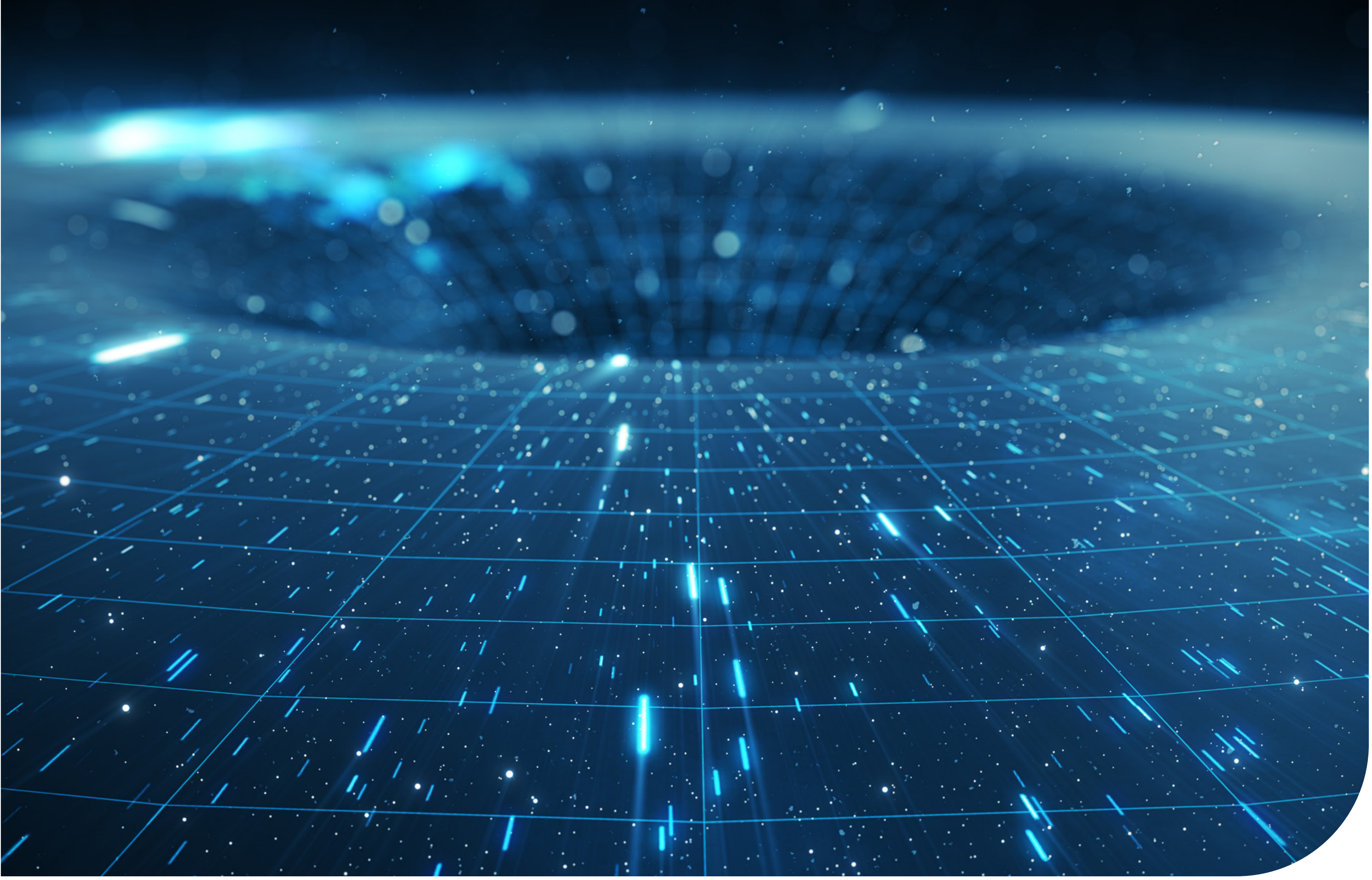
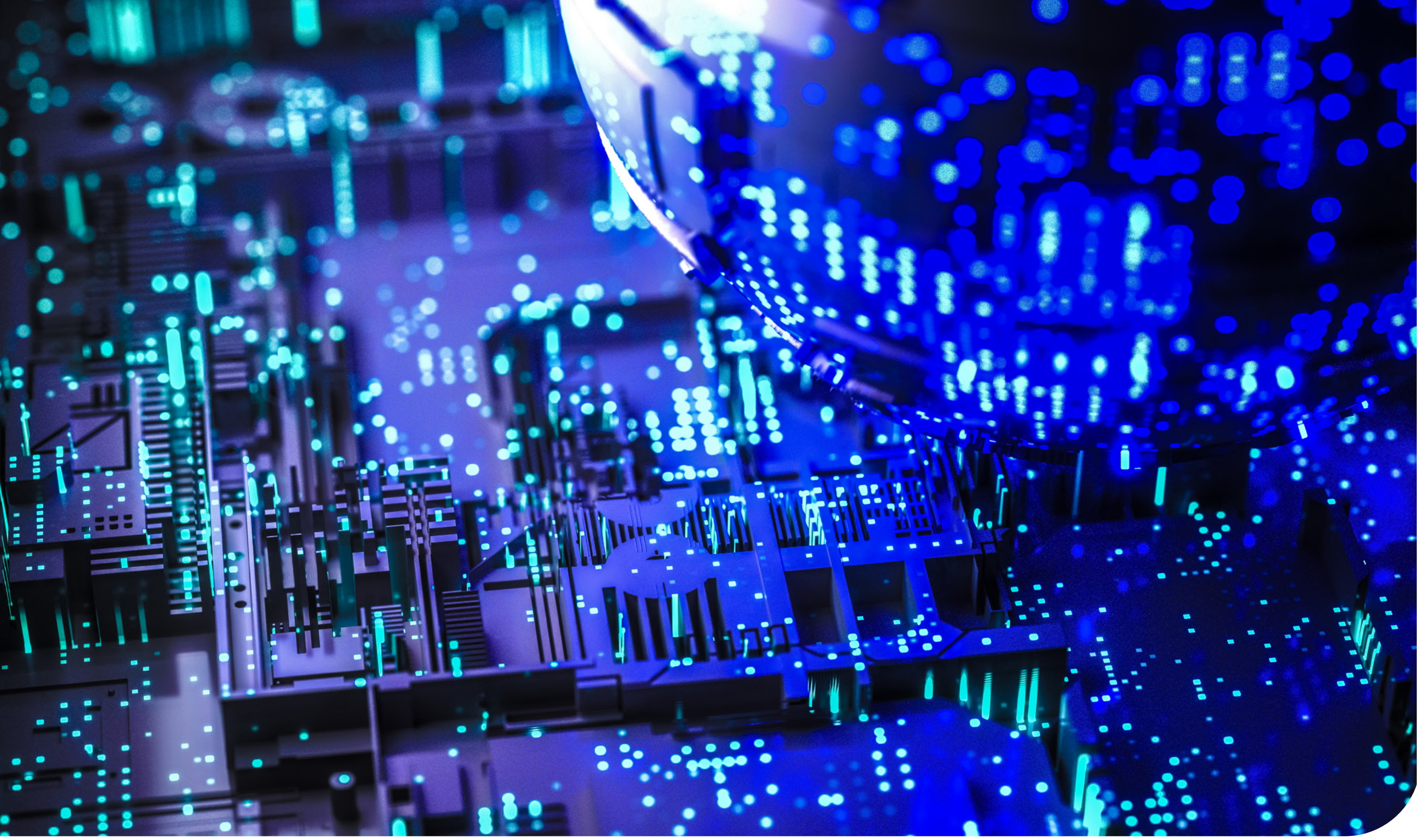


Configuring Post-Quantum MACsec in Cisco Switches

Summary

A quantum computer could break essentially all of the public key cryptography standards in use today: RSA, Diffie-Helman (DH), and Elliptic Curve Diffie-Hellman (ECDH). Since these algorithms are widely used for key exchange in various encryption protocols, a quantum computer could threaten data encryption protocols of today. Someone could store encrypted communications today and decrypt them later, if, and when, a quantum computer was available. In this whitepaper, we discuss the quantum-resistance of MACsec, which is standard for authenticating and encrypting packets between two MAC-layer, directly connected devices. We explain how quantum-resistant MACsec can be deployed in Cisco routers today, and further enhancements that can be made to improve the protocol's quantum-resistance.



Contents

Introduction

Advances and attention to quantum computing have raised security concerns among IT professionals. If a real-world quantum computer was built, it could implement quantum algorithms to break today's cryptography. These algorithms are Shor's[1] that solves the Discrete Logarithm Problem (DLP), one Shor's variant[2] that addresses the Elliptic Curve Discrete Logarithm Problem (ECDLP) and Grover's[3] that searches for discrete inputs with certain outputs.

Shor's algorithm and its variant could be used to break protocols like TLS, SSH, or IKEv2, which use DH and ECDH for key exchange of keys used in symmetric key encryption and RSA, ECDSA for authentication. Although authentication cannot be attacked by a quantum computer retroactively, someone could store encrypted communications today and decrypt them if, and when, a quantum computer was available by using Shor's algorithm. Thus, Shor's algorithm poses a threat to existing encryption protocols for transferred data with long lifetimes. Additionally, symmetric key cryptography used today is threatened by Grover's algorithm, which halves the effort to break symmetric algorithms. In other words, AES-256 offers 256 bits of classical security, and Grover drops it to 128 bits of post-quantum security.

At the time of this writing, it is clear we need to prepare for a post-quantum set of public key algorithms. NIST, ETSI, BSI, IETF, and other bodies have been working in standardizing post-quantum algorithms and using them in encryption protocols. At the same time, multiple vendors like Cisco, Microsoft, Cloudflare, Google, AWS have been looking into public key protocols.

While we are waiting for the standardization of post-quantum algorithms, not all protocol options would be susceptible to a quantum computer. In this paper, we focus on Media Access Control Security (MACsec) and provide configuration options that ensure quantum resistance of the encrypted data in MACsec.

MACsec is an IEEE 802.1AE standards-based[4,5,6] Layer 2 hop-by-hop encryption protocol that provides data confidentiality and integrity for media access (MAC) independent protocols over wired networks by using out-of-band methods for key establishment. Before establishing a MACsec secure session, the MACsec Key Agreement (MKA) protocol is used as the control protocol. MKA selects the ciphersuite to be used for encryption and to exchange the required keys and parameters between peers. MKA uses Extensible Authentication Protocol over LAN (EAPoL) defined in IEEE 802.1X[7,8] as the transport protocol to transmit MKA messages that distribute the keys. MKA provides authentication using a pre-shared key (PSK) or the 802.1X Extensible Authentication Protocol (EAP) and EAP-Transport Layer Security (EAP-TLS) framework.

The MKA/MACsec key hierarchy includes a Connectivity Association Key (CAK) established by a key agreement method (or out-of-band configuration). A Security Association (SA) defines a security relationship between members of the association. An SA is secured with a Security Association Key (SAK), forming a Secure Channel (SC). A SAK is cryptographically derived from a CAK or randomly generated by the MKA key server. SAKs are distributed to the peers by the key server using MKA messages in destination multicast MAC address EAPoL Protocol Data Units (PDU), called MACsec Key Agreement PDUs (MKPDU). These MKA messages carrying MACsec encryption keys are cryptographically encrypted and authenticated.

Thus, to provide quantum-secure MACsec, we need to ensure

1) the CAK establishment is quantum-secure, and the SAK key derivation is quantum-secure with enough entropy,

2) the encrypted MKA messages are not susceptible to quantum computer decryption.

3) And the MACsec data encryption is quantum-secure.

That way, someone storing data today would be unable to extract the CAK or SAK by using Shor's algorithm from the EAP or MKA stored communications. He also could not decrypt the MACsec encrypted flows using Grover's algorithm.

Post-quantum MACsec

Cisco switch devices implement and support MACsec. They can be configured to provide quantum-secure MACsec tunnels without requiring any additional upgrades. In summary, we only need to use 64 hex character random PSKs, AES-CMAC-256 as the key derivation function (KDF), and AES-GCM-256 authenticated encryption for the tunnel.

Below we address the three requirements that ensure post-quantum MACsec tunnels as presented in the Introduction.

1) Initially, we have to ensure that the CAK establishment is quantum-secure, and the SAK key derivation is quantum-secure with enough entropy. According to MKA, the CAK can be configured (PSK) or derived dynamically from the master-secret (MSK) of the EAP-TLS authentication step.

In Cisco IOS, the hex string of the PSK is configured under a key chain. There are two key-size options for the PSK, 32, and 64 hex characters which equal to 128 and 256 bits, respectively.

Below we show the relevant configurations for a 256-bit PSK in IOS-XE and NX-OS.

```
key chain ms-keys macsec
 key 01
  key-string <64-charatecter hex string (256-bits)>
  cryptographic-algorithm aes-256-cmac
  ! More MACsec configuration lines omitted for brevity.
```

IOS-XE MKA Pre-shared Key configuration

```
key chain ms-keys macsec
 key 1
  key-octet-string <64-charatecter hex string (256-bits)> cryptographic-algorithm
  AES_256_CMAC
  ! More MACsec configuration lines omitted for brevity.
```

NX-OS MKA Pre-shared Key Configuration

The key chain is subsequently applied in the configuration of the interface that will terminate the MKA/MACsec tunnel.

The SAK used to encrypt the MACsec data is either generated randomly from the MKA server or generated from the CAK by using AES-CMAC as the KDF with CAK as the key. Thus, using a quantum-secure KDF that generates a long enough SAK with enough entropy would ensure that the SAK generation is quantum-safe. When using AES-CMAC-256 in the configurations above, the SAK is generated using AES-CMAC in counter mode as defined in Section 5.1 of NIST SP800-108 with n=2, which generates 256 bits of pseudorandom output. Assuming the CAK key used has 256 bits of entropy, AES-CMAC-256 is a quantum-secure pseudorandom key derivation function. Thus, configuring randomly generated 256-bit PSKs for MACsec in Cisco switches will give you quantum-safe key derivation for your MACsec encrypted tunnel symmetric key.

Note about provisioning the PSK: If we want to be pedantic, we ought to make sure that the PSKs are configured on the switches in a quantum-secure way. Theoretically, someone could capture the switch configuration communications and decrypt them if, and when, a quantum computer was available in order to extract the configured CAKs and decrypt all other previously captured encrypted MACsec traffic. Given that at the time of this writing, there are no quantum-safe TLS or SSH options, you would need to make sure that configuring the tunnel PSKs is done in a way that reduces the risk of this configuration being captured and harvested later.

Note about EAP-TLS authentication: Other than PSKs, MKA/MACsec allows for the CAK to be generated from the execution of an Extensible Authentication Protocol over LAN (EAPoL) method. That EAP (IETF RFC3748) method is EAP-TLS (IETF RFC5216), as specified in[7]. An EAPoL-Key exchange occurs between the 802.1X supplicant and the authenticator to negotiate a cipher suite, exchange security parameters, and manage keys. Successful completion of all three tasks results in the establishment of an SA. After the authentication is successful, the EAP-TLS MSK is used as the CAK, which generates the SAK, as explained above. The MSK generated from EAP-TLS is derived by using TLS-PRF-48 (Section 2.3 of IETF RFC5216). TLS-PRF-48 uses a pre_master_secret key, which is the shared-secret from the TLS negotiation. At the time of this writing, the industry has been looking into post-quantum TLS key exchange[9, 10, 11, 12, 13], but there is no standardized quantum-secure TLS key exchange which means that the MSK derived from EAP-TLS is not quantum-resistant. Consequently, the derived from the MSK, MACsec SAK would be vulnerable to Shor's algorithm. To ensure quantum-secure MACsec tunnels with EAP-TLS authentication, you would need to wait until we had a standardized key exchange in EAP-TLS. Additionally, although not as urgent as key exchange, post-quantum authentication in EAP-TLS[14] will also need to be standardized for a fully post-quantum MACsec tunnel.

2) Secondly, after establishing the CAK and deriving the SAK, we need to ensure the encrypted MKA messages distributing the SAK are not susceptible to quantum computer decryption. MKA messages are encrypted with a Key Encryption Key (KEK) and authenticated with an Integrity Check Key (ICK). The key encryption algorithm is AES Key Wrap, as defined in IETF RFC3394 and NIST SP 800-38F. The authentication algorithm is AES-CMAC, which produces an Integrity Check Value (ICV).

The KEK and ICK are generated using AES-CMAC as a KDF in counter mode. When using AES-CMAC-256 as in the key chain configurations above, the KEK and ICK are generated using AES-CMAC in counter mode as defined in Section 5.1 of NIST SP800-108 with n=2 which generates 256 bits of pseudorandom output. Thus, assuming the CAK key used has 256 bits of entropy, the KEK and ICK keys are generated in a quantum-secure, pseudorandom way, which provides 256-bits of entropy when AES-CMAC-256 is used. That leads to a quantum-safe distribution of the MKA symmetric key used to encrypt the MACsec tunnel.

3) Finally, the data encryption of the MACsec tunnel needs to be quantum-secure. In MACsec, the derived SAK is used as the symmetric key for AES-GCM authenticated encryption. Depending on the derived SAK length, AES-GCM-128 or 256 can be used in Cisco switches. Assuming we have used 64-character hex string as in the key chain configurations above with AES-CMAC-256 as the cryptographic algorithm, AES-GMC-256 will offer quantum resistance.

Below we show the MKA policy configuration lines for AES-GCM-256 in IOS-XE and NX-OS.

```
mka policy ms-p
 macsec-cipher-suite gcm-aes-256
 ! More MACsec configuration lines omitted for brevity.
```

IOS-XE MKA Policy Configuration

```
macsec policy 1
 cipher-suite GCM-AES-256
 ! More MACsec configuration lines omitted for brevity.
```

NX-OS MKA Policy Configuration

The MKA policy is then applied under the MACsec interface configuration of the Cisco switch.

Then, you have a quantum-resistant MACsec tunnel. In summary, we used 64 hex character random PSKs, AES-CMAC-256, as the KDF and AES-GCM-256 authenticated encryption of the tunnel.

Readers should note that MKA/MACsec includes a very similar process for group establishing group CAKs and deriving pairwise KEK and ICK keys in order to distribute the SAKs generated from the group CAK to the members of the group. The equivalent configurations would use the same algorithms and PSKs.

Note about using AES-CMAC-128 in the key chain and AES-GCM-128 in the MKA policy, respectively: As it was argued in NIST's PQ Cryptography FAQ regarding AES key lengths (added 11/18/18), AES-128 can be considered secure for decades to come even if there was a real-world quantum computer. This argument is mainly because of the cost of a quantum computer and because Grover's algorithm is not parallelizable as proven in[19] and discussed in[20]. Thus, there is a reasonable argument to be made for using AES-CMAC-128 in the MACsec key chain with 32-hex character PSKs and AES-GCM-128 MKA policies instead of AES-CMAC-256, 64-hex character strings with AES-GCM-256 in the configuration examples above. Arguably, these options could provide quantum-resistance in MACsec for a long time. It would be up to a device administrator to make the decision between being practical (128-bit AES) or conservative (256-bit AES), but in this whitepaper we chose to be conservative because there are no practical drawbacks to using AES-CMAC-256 and AES-GCM-256 over their 128-bit equivalent in Cisco switch platforms.

For more information on all configuration options for MKA/MACsec tunnels, refer to the Cisco IOS XE 3S MACSEC and MKA Configuration Guide, the Catalyst IOS XE 3.10 4500 Series Switch Software Configuration Guide, and the Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 7.x. For a summary of protocol requirements necessary to add quantum resistance to VPN technologies, including MACsec, refer to the ETSI TR 103 617 V1.1.1.

Conclusion

In conclusion, while waiting for post-quantum public key algorithms to be standardized, we could be leveraging quantum-secure algorithms in our tunnels today to be resilient against store-now-decrypt-later attacks. Specifically for MACsec, when quantum computers are a concern for data encrypted in MACsec tunnels, Cisco switches can offer adequate level of protection. We showed how statically configured 256-bit pre-shared keys in Cisco switches and configuring quantum-safe algorithms AES-CMAC-256 and AES-GCM-256 can offer long term protection. We also discussed why 802.1X EAP-TLS authentication for MKA does not provide quantum resistance and the updates that will need to take place to introduce quantum-resistance for MACsec overall.

Contact

Panos Kampanakis – Product Manager – panosk@cisco.com

References

[1] 1997. Peter W. Shor. “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer.” SIAM Journal on Computing 26, 1484–1509. MR 98i:11108. 1994 version: “Algorithms for quantum computation: discrete logarithms and factoring.” MR 1489242. Pages 124–134 in Shafi Goldwasser (editor). 35th annual IEEE symposium on the foundations of computer science. Proceedings of the IEEE symposium held in Santa Fe, NM, November 20–22, 1994. IEEE. ISBN 0-8186-6580-7. MR 98h:68008.

[2] 2003. John Proos, Christof Zalka. “Shor’s discrete logarithm quantum algorithm for elliptic curves.” Quantum Information & Computation 3, 317–344. MR 2004h:81067.

[3] 1996. Lov K. Grover. “A fast quantum mechanical algorithm for database search.” MR 1427516. Pages 212–219 in Proceedings of the twenty-eighth annual ACM symposium on the theory of computing, held in Philadelphia, PA, May 22–24, 1996. ACM Press. ISBN 0-89791-785-5. MR 97g:68005.

[4] IEEE 802.1AE-2006, IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Security.

[5] IEEE 802.1AEbn-2011, IEEE Standard for Local and metropolitan area networks--Media Access Control (MAC) Security Amendment 1: Galois Counter Mode--Advanced Encryption Standard-- 256 (GCM-AES- 256) Cipher Suite”

[6] IEEE 802.1AEbw-2013, IEEE Standard for Local and metropolitan area networks-Media Access Control (MAC) Security Amendment 2: Extended Packet Numbering”

[7] IEEE 802.1X-2010, IEEE Standard for Local and metropolitan area networks--Port-Based Network Access Control”.

[8] IEEE 802.1Xbx-2014, IEEE Standard for Local and metropolitan area networks -- Port-Based Network Access Control Amendment 1: MAC Security Key Agreement Protocol (MKA) Extensions” (Amendment to IEEE Std 802.1X-2010).

[9] Post-Quantum TLS (Microsoft)

[10] Post-quantum confidentiality for TLS (Google)

[11] Post-quantum TLS now supported in AWS KMS

[12] The TLS Post-Quantum Experiment (Cloudflare)

[13] IETF Hybrid key exchange in TLS 1.3 Draft

[14] Post-Quantum Authentication in TLS 1.3: A Performance Study

[15] Cisco IOS XE 3S MACSEC and MKA Configuration Guide

[16] Catalyst IOS XE 3.10 4500 Series Switch Software Configuration Guide

[17] Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 7.x

[18] ETSI TR 103 617 V1.1.1 Quantum-Safe Virtual Private Networks

[19] Christof Zalka, Grover’s quantum searching algorithm is optimal. Review A 60.4 (1999): 2746–2751. Crossref. Web.

[20] Scott Fluhrer, Reassessing Grover’s Algorithm, Cryptology ePrint Archive, Report 2017/811, 2017.