Quantum Safety in OT protocols

© *Rob Hulsebos*

Version 26 March 2026

Introduction

There is a lot of interest in "Q-Day" – the day that quantum computers are able to break our current network encryption algorithms and divulge our secrets. So-called "quantum-safe" encryption algorithms have already been developed; they are already in our popular browsers at this very moment.

I find a lot of documentation about all sorts of cryptographic algorithms, key exchanges, hashes, PKI, authentication, Diffie-Hellman, elliptical curves, etc. But I am not a mathematician and it is easy to get lost.

Instead, I'm a user. I use OT-protocols like Modbus/TCP, DNP3, IEC104 (and many others) and an IT-protocols as well like SMB, Radius and SNMP (and many others too). Are *they* quantum safe? If not, will they be soon? It is difficult to find more details about it without getting drowned in a lot of jargon.

Below, I have alphabetically listed various protocols, with comments about their quantum safety.

> ***Note**: this is "Work In Progress". Expect updates – there any many more protocols, and the protocols themselves are evolving, as is my knowledge of this subject. If your-favourite-protocol isn't listed here, please let me know, it might be added to the list.*

Alphabetical overview

CIP Security

CIP Security (for Ethernet/IP devices) is built on TLS or DTLS (1.2 or higher). It allows use of either RSA or ECC public/private key pairs, and SHA-256.

TLS1.2 and DTLS1.2 are quantum-unsafe, as are RSA and ECC.

DNP3

DNP3 is by origin (1980's) a serial protocol. Given this origin, encryption was not part of the protocol. Later, the Ethernet-based version was developed, which is basically a 1:1 port of the serial DNP. So, both offer no encryption.

DNP3-SA (Secure Authentication) is an extension to add authentication and message integrity checking. In principle it is a software independent of DNP3, so could be used by other protocols too (but it is unknow whether anyone has done that). It is possible to use authentication-only mode.

> *DNP3-SA is part of "DNP3 Security", a.k.a. IEEE-1815, of which the current version is 6 (launched 2020); version 5 is from 2012. Versions 1, 3 and 4 were for internal use only and never brought on the market (a security analysis discovered vulnerabilities). Version 1 was published in 2007; V2 was the first version brought on the market (2010).*

Version 5 does not do encryption yet; V6 is the first version to offer full encryption at the protocol level. It is expected to be available this year (2026), so we'll see SA V5 for a long time to come.

The algorithm used by DNP3 SA 6 is AEAD-AES-256-GCM (AEAD = Authenticated Encryption with Associated Data). This is seen as quantum-safe. However, in this paper it is argued otherwise (https://ieeexplore.ieee.org/document/11104845) due to a reliance on classical cryptography and static configurations. Instead, the paper proposes a quantum-safe variant based on ML-KEM with ECC-Diffie-Hellman).

DNP3-SAB

This is the broadcast variant of DNP-SA. See DNP3.

DTLS

DTLS is the UDP-based version of TLS. DTLS is used in many Machine-to-Machine standards, but the older versions of DLTS are not quantum-safe.

DTLS Version 1.3 is based on TLS 1.3, as specified in RFC9147 (successor of older RFC6347). Because DTLS and TLS are very similar, the exceptions are needed for running on UDP (instead of TCP).

DTLS is also used inside CIP Security (see elsewhere).

IEC 60870-5-104 (IEC104)

Standard IEC 60870-5-104 is not encrypted. A security extension is available that uses TLS1.2 or TLS1.3, as per IEC 62351-3. Usage of TLS1.3 is considered quantum-safe but only if a strong enough algorithm is chosen like AES-256 (if products may allow the choice for other algorithms this should be considered carefully).

IEC 61850 with GOOSE

Standard GOOSE is not encrypted. The encryption as specified in companion standard IEC 62351-6 use classical algorithms as RSA and ECDSA, which are not quantum-safe.

Developments on quantum-safe algorithms may not influence the real-time behaviour needed in GOOSE (typical 3 msec for certain critical operations). Some proposed algorithms are FALCON/FN-DSA, RSS and BLAKE-2S. But many existing devices probably lack the CPU-power needed.

IEC 61850 with MMS

The MMS protocol used in IEC 61850 itself uses TCP/IP which is often not encrypted. When encryption is used, it is according AES, considered quantum-safe with key lengths >= 256.

For use inside substation, usage of MAC Sec (see below) can be a solution.

Future versions, according to IEC 62351-4, may use quantum-safe encryption.

IPSec

Standard IPsec/IKEv2 is not quantum-safe because it relies on RSA/DH/ECDH. The new version which uses post-quantum cryptography (like ML-KEM and ML-DSA) are quantum-safe.

MACSec

This is a protocol used at Ethernet-level, encrypting all traffic at layer 2. For encryption, it uses AES-Galois/Counter Mode (GCM), which is considered quantum-safe if the key length is long enough).

Since the MACSec encryption is done at layer 2, having no encryption at higher levels is not necessarily a weakness. But note that MACSec cannot work in routed networks or outside a LAN, severely limiting its usefulness in making all communication fully quantum safe.

Modbus/TCP

Modbus/TCP doesn't use any authentication and encryption at all, this is major weakness of this protocol. Despite these weaknesses, it is a very popular protocol, and many connect their equipment directly to internet (according to Shodan).

In 2018, Schneider published the specification for a "Modbus/TCP Security" or "Modbus/Secure" extension using TLS1.2 (because 1.3 didn't exist at the time) using AES128. Both TLS1.2 as AES128 are not considered quantum-safe.

Also, I haven't seen much implementations for Modbus/TCP Security in products. Rumour has it that it is supported in some recent Schneider products (but I haven't verified), and (surprisingly) in Siemens S7-1500/ET200 (but unsure whether this complies with the Schneider specification). And there is even an open-source implementation here: **https://github.com/digitalpetri/modbus**.

Concluding: in general, Modbus/TCP communication cannot be considered quantum-safe.

OPC/UA (except PubSub)

OPC/UA supports three different security modes; only "SignAndEncrypt" uses encryption. Unlike many others, OPC/UA does not rely on TLS (except for in the PubSub architecture). It allows vendors some freedom to implement various encryption schemes.

At this moment, OPC/UA cannot be considered quantum safe, but prototypes of a new version will probably use standard NIST-approved algorithms like ML-KEM and ML-DSA.

OPC/UA PubSub

When enabled, it uses (symmetric) AES-CTR with a 128-byte key. This is considered too short to be quantum safe, however with current quantum technology it is considered safe for decades to come.

Signature signing is via HMAC-SHA2-256, which is generally considered quantum-safe.

OpenSSH

This implementation supports a number cryptographic key agreement algorithms considered to be quantum-safe since release 9.0 (April 2022). In OpenSSH 9.9 a second algorithm was added, which also became the default in OpenSSH 10.0 (April 2025).

To warn users and to encourage migration, OpenSSH 10.1 show a warning message when a non-quantum-safe algorithm is in use:

```
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
```

OpenSSL

This is an open-source toolkit for TLS1.3, DTLS 1.2 and QUIC 1.0. OpenSSL v3.5 has been released April 8, 2025 and support ML-DSA, ML-KEM in TLS1.3, both considered to be quantum-safe.

Profinet

Encryption of real-time I/O data is possible in Security Class 3; in Security Class 2 has device authentication (between I/O controller and I/O device); Security Class 1 has no cryptographic protection.

Both class 2 and 3 use EAP-TLS for authentication and key management, and AES-GCM or ChaCha20-Poly1305 with 256-bit keys. AES-GCM-256 and ChaCha20-Poly1305 are both considered quantum safe,

The Profinet User's Group states that TLS1.3 will be used with EAP-TLS and "cipher suites that are in good standing" as of Q3 2025) which promises to make it quantum safe. However, there is no implementation on the market yet.

> *However, standard TLS is not capable of handling the real-time requirements of ProfNet. Additionally, the long-lasting network connections in OT (sometimes multiple years), which may cause overflow of internal counters and allow counters values to be reused (potentially allowing message replay attacks). To prevent this, a new cryptographic key must be renegotiated, but without interference on the real-time communications).*

RADIUS (Remote Dial-in User Service)

The RADIUS protocol has various shortcomings regarding encryption, such as sending data "in the clear" and the usage of MD5 as authentication algorithm. It is not considered quantum-safe.

RadSec , Radius/TLS, Radius1.1

This is an implementation of Radius over TLS (port 2083), offering end-to-end encryption.

Although the use of TLS1.3 is required, it doesn't automatically ensure usage of quantum-safe algorithms; they need to be added

Radius/DTLS

An implementation of Radius over DTLS (RFC9765) is possible. MD5 usage has been removed.

SMBv3

System Message Block is a Microsoft protocol, also known as "Samba". It has many implementations in non-Windows environments too. Over the years, the protocol evolved from SMBv1 to SMBv2 to SMBv3, and the latter is now at version v3.1.1.

> *Sidestep: SMBv1 and SMBv2 should not be used anymore; they have lots of weaknesses (not even related to quantum safety). SMBv3 also had its share of problems, so use V3.1.1 where possible. See* ***https://learn.microsoft.com/en-us/windows-server/storage/file-server/troubleshoot/detect-enable-and-disable-smbv1-v2-v3?tabs=server***

How is encryption handled in SMBv3? Microsoft says: "Windows Server 2022 and Windows 11 introduce AES-256-GCM and AES-256-CCM cryptographic suites for SMB 3.1.1 encryption. Windows automatically negotiates this more advanced cipher method when connecting to another computer that supports it. You can also mandate this method through Group Policy. Windows still supports AES-128-GCM and AES-128-CCM. By default, AES-128-GCM is negotiated with SMB 3.1.1, bringing the best balance of security and performance."

AES is a symmetric cipher, so considered quantum-safe if the key length is sufficiently large; AES-256 is advised. It can be changed via a group policy, or PowerShell (**https://learn.microsoft.com/en-us/windows-server/storage/file-server/smb-security**).

PowerShell Copy

```
Set-SmbServerConfiguration -EncryptionCiphers "AES_128_GCM, AES_256_GCM" -Confirm:$false
```

This command specifies the encryption ciphers used by the SMB client, and the preferred order without user confirmation.

Migrating to SMBv3 is impossible with older Windows versions that only have SMBv1 and/or SMBv2. Updating these old Windows systems is therefore advised, but in many cases, it may be difficult or impossible (typical for OT).

SNMP

SNMPv3 has the capability for encryption, but is it quantum safe? It depends. For encrypting the data payloads, it supports DES (Data Encryption Standard), 3DES (Triple DES) and AES-128/192/256 (Advanced Encryption Standard). DES and 3DES have since long been retired. AES is generally considered quantum-safe, it being a symmetric algorithm, but it is not entirely immune

to quantum attacks (Grover's algorithm). But if the key length is long enough (> AES-256), it is considered safe for the foreseeable future.

Authentication is done via MD5 or SHA1, both of which must not be used for implementations as they are not quantum safe (and have other weaknesses as well).

Luckily, the SNMPv3 architecture allows usage of TLS (Transport Layer Security) and DTLS (Datagram TLS), see RFC9456 (**https://www.rfc-editor.org/rfc/rfc9456**). But TLS1.3 doesn't automatically ensure usage of quantum-safe algorithms; they need to be added. Contact your vendor for details.

SSL2.0 and SSL3.0

SSL has been superseded by TLS for over a decade, due to serious weaknesses in the encryption. SSL3.0 was officially deprecated in 2015 and SSL2.0 in 2011. SSL1.0 was never published due to serious flaws. SSL is vulnerable due to usage of weak encryption algorithms (like MD5), making it quantum-unsafe.

> *Note: it is confusing that there are products on the market with "SSL" in the name, while in fact they communicate over TLS1.3 (i.e., see WolfSSL below) and may be quantum-safe. Check the documentation of these products.*

S7Plus (OMSPlus)

S7Plus is the successor of the well-known S7 protocol from Siemens. It has a proprietary encryption algorithm, of which no information could be found. Given its age it is very likely not quantum-safe.

TACACS+

This is not quantum-safe due to the use of MD5. A newer version is under development (RFC9887, released December 2025). It is based on TLS1.3, but check to make sure that it has the quantum-safe algorithms in it.

WolfSSL

This is a commercial SSL implementation, using TLS1.3 and DTLS 1.3, and support post-quantum algorithms ML-KEM and ML-DSA (https://www.wolfssl.com/products/wolfcrypt-post-quantum/).

Summary

Many protocols are migrating to, or already have been migrated to TLS1.3. When a quantum-safe cipher is then selected by the client and server, communication is considered quantum-safe.

If you have any suggestions, comments or additions,
please do not hesitate to contact me (email: rh[at]enodenetworks.com).