Quantum Computing and the future of security

Ewa Anna Szyszka · 2020-08-06

The internet security industry relies heavily on the complexity of mathematical problems, which are impossible to solve in a reasonable time frame by classical binary computers. It might take your computer hundreds of years to crack a message protected by cryptographic algorithms. Even if you had access to a computer farm and better computational resources, you might need a couple of decades to decrypt a secured message.

With the arrival of quantum computers and popularization of their use, this might soon change. The very same cryptographic algorithms securing websites, financial transactions, and your sensitive data might quickly be put in peril by quantum supremacy. What does quantum computers' arrival mean for the future of security, and what vulnerabilities of security systems would popularization of quantum computation bring? To answer those questions, let's dive into a short IT security history.

History of Web Security

HTTP stands for HyperText Transfer Protocol, and was initially developed at CERN by Tim Berners-Lee as a communication tool for scientists [1]. Nowadays, it is a widely used client-server communication protocol and the key building block of the web. As the need for a more secure browsing experience increased with the rise of online banking and sensitive information handling, the **HTTPS** protocol was created. HTTPS is used as an encryption standard, providing a secure browsing experience as opposed to HTTP.

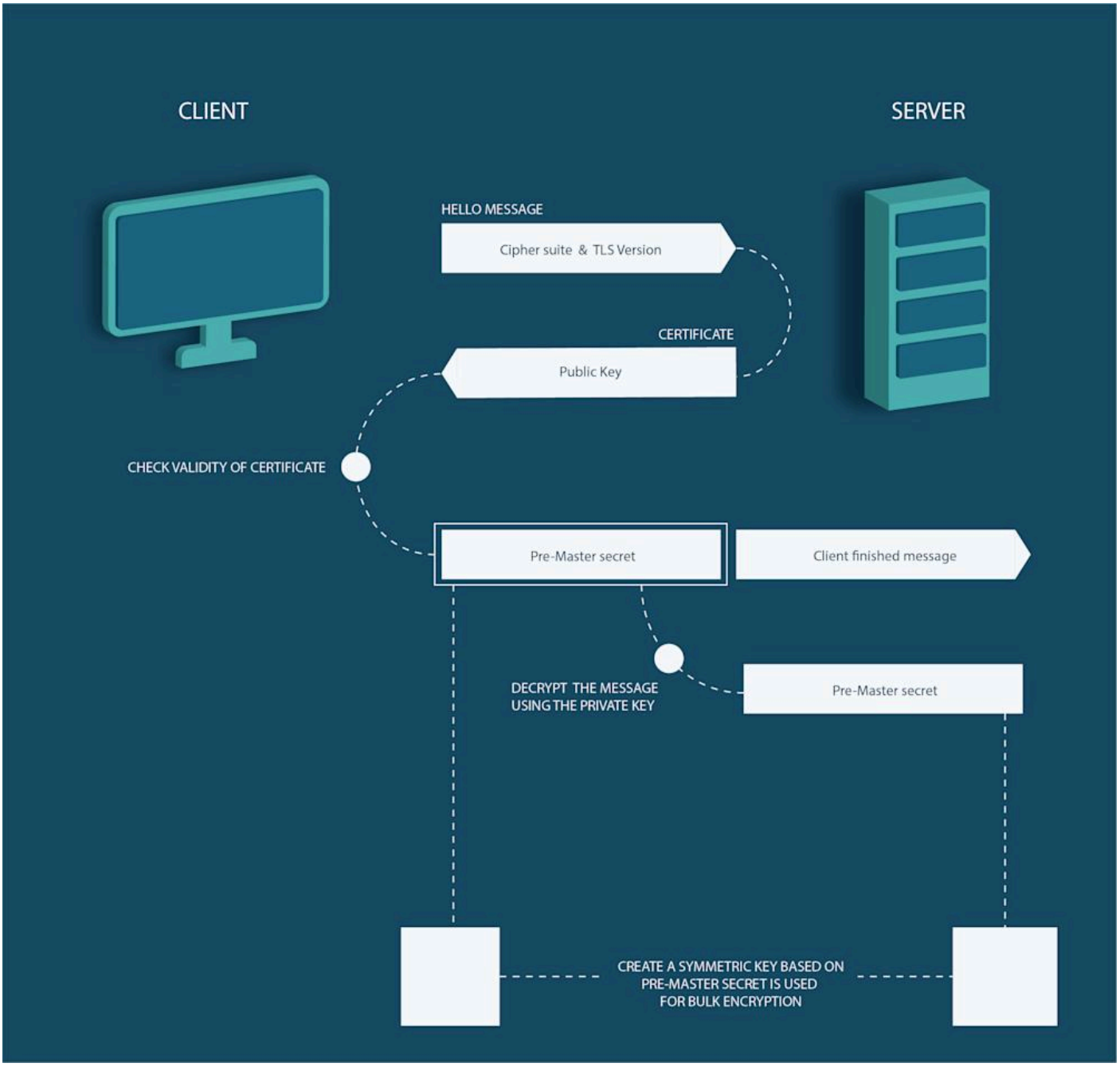
The *S* in *HTTPS* stands for **Secure Sockets Layer (SSL)**, which evolved into **Transport Layer Security (TLS)** over time. It is a protocol responsible for securing the communication between the server and the client. HTTPS security relies on **TLS handshakes**, which are illustrated below.

Client and server are communicating via symmetric encryption, which means they are using the same key to encrypt the message and decrypt it. For this to happen, the Client and Server need to first exchange keys between each other.

To do so, the client sends a *Hello* message to the server, and both agree on a **Cipher Suite**. A Cipher Suite is a set of algorithms that allow the client and server to exchange keys and to encrypt and verify data.

One such key exchange algorithm is the Rivest–Shamir–Adleman (RSA) algorithm, which makes use of the fact that it is difficult to find the prime factors of any large number.

Once the keys are exchanged, the bulk encryption of data is perfomed using a symmetric key, created using the so called pre-master secret.



To illustrate how RSA works, let's start with the assumption that two parties want to communicate securely. They both want to achieve two tasks, namely, being able to encrypt and decrypt messages. The server has both encryption and decryption keys. In our simplified example, for encryption, this would be a pair of numbers (5,14). For decryption, we use a different pair of numbers (11,14).

The encryption numbers are publically available, but the decryption numbers are available only to the server.

If we want to send a simple message, we need to convert it into numbers first. This way, we are able able to manipulate it mathematically. For example, we can use the letter's index in the English alphabet as a text to number conversion.

Next, these numbers will be raised to the power of the first key, and a modulo operation using the second key is applied. The manipulated values would then be sent on the network. In our example below, a letter *B* would thus be converted to the number 4.

The decrypting party then uses the decryption keys to convert the value 4 back to the initial message.

We first take the number to the power of the first key, and then and apply a modulo operation using the value of the second key.

```
————— ENCRYPTION TASK —————
Encryption Keys: (5,14)
Text: B
Text as number: 2
Encryption: 2 pow 5mod 14 = 32mod14 = 4
```

```
————— DERYPTION TASK —————
Decryption Keys: (11,14)
Cipher: 4
Reverse the cipher: 4 pow 11 mod 14 = 4194304 mod14 = 2
Numer as text: B
```

This process illustrated above can be taken advantage of by quantum computers, thereby putting in peril many security systems such as the TLS handshakes.[3]

Although there are many alternatives to the RSA algorithm, the RSA algorithm's case illustrates an important issue that the popularization of quantum computers will raise. In the next section, we will take a closer look at specific examples of how quantum computing would interfere with the current security paradigms.

Future of Web Security

In 2019, Google Engineers made a breakthrough with the invention of the Sycamore processor. This processor can compute specific tasks in 200 seconds what would take a regular silicon-based processor 10,000 years and more [2]. Although quantum computing is still in an early stage and can only be applied to particular tasks, you get a sense of the revolution that awaits us.

As mentioned in the first part, many security algorithms rely on the inability of modern computers to calculate the factors of large numbers in a short time. The RSA algorithm, for instance, could be cracked

by a brute-force attack. Just try random numbers until you eventually end up with the correct key.

For modern computers, this task would take thousands of years. However, with the usage of **Shor's algorithm** and its ability to turn those random guesses into more meaningful guesses, alongside with the ability to process information in parallel by quantum computers, our current security solutions might be put in peril.

As quantum supremacy would make our current security measures obsolete, the demand for new security measures and the redefinition of existing cryptographic primitives arises.

Quantum cryptography

Quantum cryptography is a field that defines such new paradigms. More specifically, **Quantum Key Generation (GKG)** is a technology that allows sending information encoded as photons.

Photons are polarized either horizontally (H), vertically (V) or diagonally +45 degrees, and -45 degrees. Thus, binary information would be translated to photons in such a way that V and +45 encoding would, for example, mean 0 and H, and -45 means 1.

Those photons are sent to the receiver using one of the encoding systems mentioned above. The receiver needs to use the correct encoding either V/H or +45/-45 degrees, otherwise, the information would be altered [4].

If an eavesdropper tried to listen to the information being sent, it would be protected by the physic law called **Heisenberg Uncertainty principle**. It states that we cannot know the speed and the position of an object at the same time, since by measuring we would alter the objects properties. Therefore, if an eavesdropper would try to listen in to the communication, it would change the photons, and both communicating parties would be informed.

Sources

[1] CERN (2020) **The birth of the Web** https://home.cern/science/computing/birth-web

[2] David A., Burkett, Brian (October 2019). **Quantum supremacy using a programmable superconducting processor**. Nature. 574 (7779): 505–510. doi:10.1038/s41586-019-1666-5

[3] Eddie Woo (2014) **The RSA Encryption Algorithm** https://www.youtube.com/watch?v=4zahvcJ9glg

[4] Mizutani, A., Sasaki, T., Takeuchi, Y. et al. **Quantum key distribution with simply characterized light sources** npj Quantum Inf 5, 87 (2019). https://doi.org/10.1038/s41534-019-0194-3

This article was published on the Search Guard blog: https://search-guard.com/blog/quantum-computing-security/