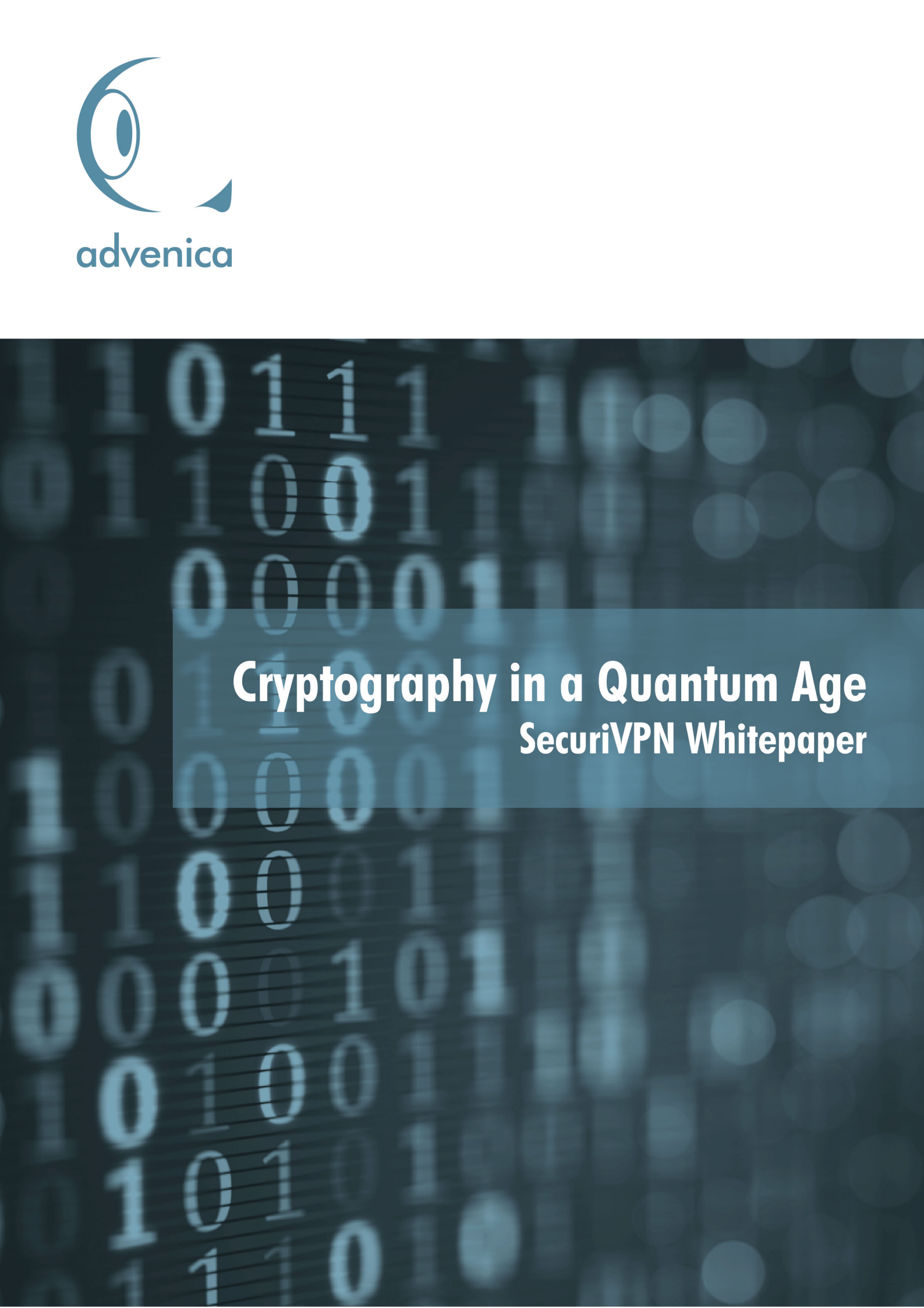
advenica
Cryptography in a Quantum Age
SecuriVPN Whitepaper

Document version: 17385v1.3 Whitepaper - Cryptography in a Quantum Age

TABLE OF CONTENTS

1. CRYPTOGRAPHY OVERVIEW

Cryptography has its roots in the Greek *kryptós graphein*, literally translated to *the study of secret writing*. Today, cryptography is all about secure communication over an insecure public channel. This is achieved by protecting the confidentiality and integrity of information. Without it, anyone could read a message or forge a private conversation.

Using a cipher and performing the process of encryption, information is scrambled and transformed from *plaintext* into *ciphertext*, i.e. information is made secret. Decryption turns scrambled and unreadable *ciphertext* back into *plaintext*. In both encryption and decryption, the so called *encryption key* controls the ability to hide and reveal information. Modern cryptography can be divided into two major categories, namely *symmetric key* and *asymmetric key* (also known as *public key*) *cryptography*.

In symmetric key cryptography, the same key is used for both encryption and decryption. The key needs to be kept a secret by everyone who is sending and receiving private messages. The figure below depicts a simple encryption schema. The major challenge of symmetric key cryptography is the process of distributing the key to the communicating parties without exposing it to external threats.

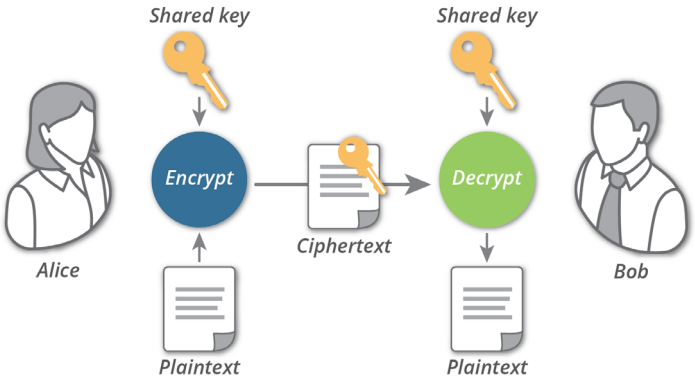

Figure 1 - Symmetric key cryptography.

Asymmetric key cryptography uses two different but interrelated keys, one key for encrypting and another key for decrypting information. The keys are known as *public* and *private keys*. As the name implies, only one of the keys is intended to be kept secret. Anyone who intends to send encrypted information uses the public key of the receiver. On the other side, the receiver must use the second key, his own private key, to decrypt the information (see the figure below). The mathematical relationship between the public and the private key is such that calculation of the private key is computationally infeasible from the public key.

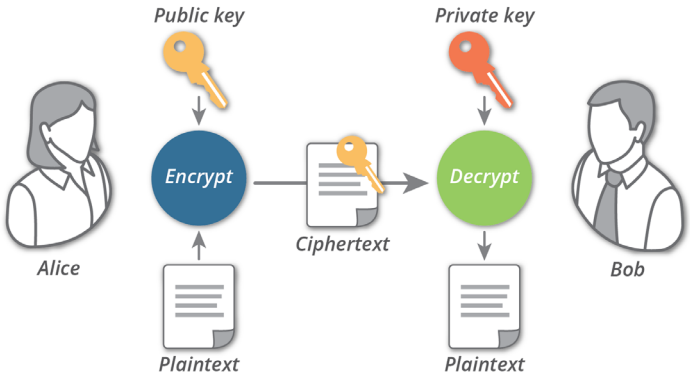

Figure 2 - Asymmetric key cryptography.

Cryptography enables:

- Data confidentiality – sensitive information is kept secret
- Data integrity – information is protected against modification when sent over a public communication channel
- Authentication – communicating parties are who they claim to be.

In practice, cryptography combined with security protocols are the very essence, or cornerstones, of modern secure communications. Should cryptography fail, all sensitive information ever sent over public channels become readable to everybody with capability to passively intercept and record the information.

1.1 THE KEY DISTRIBUTION CHALLENGE

Let's assume that *Alice* wants to send a secret message to *Bob*. She uses a symmetric key to encrypt the message. *Alice* sends the ciphertext message to *Bob*, who knows the key and uses the same symmetric algorithm, e.g. AES, to decrypt the message. In this simplified scenario, both *Alice* and *Bob* had access to the required key. As long as they are the only ones who know the key, no one else is able to read the encrypted message.

But how was the key distributed to *Bob*? *Alice* could select a key and physically deliver to *Bob*. If *Alice* and *Bob* have communicated previously, *Alice* could use a previous key to encrypt a new key and sent it to *Bob*. The challenge is scalability, if there are 10,000 people who want to communicate with each other, everyone need 9,999 different keys to establish separate and confidential communication channels.

Some other key distribution alternatives:

- a third party can select and deliver the key to *Alice* and *Bob*
- if *Alice* and *Bob* have secure communications with a third party called *Claire*, *Claire* can relay the key between *Alice* and *Bob*

The challenge with all the above alternatives is secure key distribution. The security is compromised if a man-in-the-middle can access the key while it is being distributed.

Can an asymmetric algorithm, such as RSA, solve the problem of secure key distribution? In asymmetric algorithms two keys are used, a public and a private one. The public key is available for everyone, while the private key is kept secret and known only by the user. When the message is encrypted with the public key, only the corresponding private key

can decrypt it. It is also computationally infeasible to calculate the private key from the public key. Thus, when *Alice* wants to distribute a symmetric key to *Bob*, she takes *Bob*'s public key and uses it to encrypt the symmetric key. Only *Bob* can then decrypt the encrypted symmetric key, because he is the only one who knows the corresponding private key.

The general problem with asymmetric algorithms is the performance. They are quite slow compared with symmetric algorithms. However, as discussed above, asymmetric algorithms are perfect for solving the problem of secure key distribution. In fact, today asymmetric algorithms are widely used to securely distribute symmetric keys. Once secure key distribution is achieved, *Alice* and *Bob* can use a symmetric algorithm, e.g. AES, and the symmetric key to make the communication confidential.

Use of an asymmetric algorithm also solves the scalability problem. Every communicating party needs only one public key and one private key to establish secure and confidential communication with the other party.

We can conclude that communication systems today use:

- Symmetric encryption to provide confidentiality of the message.
- Asymmetric encryption to securely distribute the symmetric key.
- Asymmetric encryption to solve the scalability problem related with symmetric encryption.

1.2 QUANTUM COMPUTING, A SHORT OVERVIEW

Quantum computers are devices that process information using physical phenomena unique to quantum mechanics, obeying the laws of quantum mechanics.

In a classical computer the basic unit of information is the bit, which is a two-state device that can represent the values 0 and 1. In quantum computing, the fundamental unit, qubit, can hold both a 0 and a 1 value at the same time; this is known as a superposition of two states. Exploring this fact that the qubit can exist in multiple states simultaneously, the quantum computer, with a single quantum processor, is able to perform multiple computations. This is known as quantum parallelism, and it is the key property in the power of quantum computers. It gives quantum computers an advantage over classical computers in that they can perform very rapid parallel computations without the need of having several processors linked together. Thus, a quantum computer is able to solve certain problems like searching and factoring much faster than it would take a classical computer to solve the same problem.

Algorithms that exploit quantum effects, such as superposition, are known as quantum algorithms. The most well-known quantum algorithms are Shor's algorithm and Grover's algorithm. The Shor's algorithm for factoring runs exponentially faster with respect to any known classical computation while the Grover's algorithm for searches (e.g. searching unstructured databases) runs quadratically faster than the best possible classical algorithm for the same task.

The widely used asymmetric cryptography (e.g. RSA) relies on the assumption that prime factorization of large numbers takes extremely long time to solve and thus it would take a very long time for information to be decrypted. However, this assumption is challenged by the speed of the Shor's algorithm.

In summary, Shor's and Grover's algorithms threaten many widely used cryptosystems that base their security on the premises that certain computational problems are extremely

difficult to solve. The quantum algorithms can solve these classes of problems quickly enough to jeopardize the security of the information that is protected by cryptography.

1.3 QUANTUM COMPUTING IMPACT ON CRYPTOGRAPHY

Quantum computers are able to quickly reverse calculate private keys, a task that is considered impossible for a conventional computer. By breaking the cryptographic keys, an eavesdropper is able to decrypt private communications and pretend to be someone whom they are not. In essence, quantum computing compromises the very principal of secure communication – confidentiality, integrity and authentication.

Some of our most widely used cryptosystems such as RSA and Elliptic Curve Cryptography have already been demonstrated to be insecure in the presence of a quantum computer. Everything transmitted over a public communication network is vulnerable to both eavesdropping and modification. These issues do not only impact information encrypted in this manner in the future, they also apply to information already transmitted over a public network.

For organizations and institutions with interest in keeping secret information safe from adversaries, it is absolutely essential to be forward thinking when it comes to information security. This involves considering how long information needs to stay secure. Organizations need to have a very clear view of the practical consequences of a certain category of information becoming public knowledge long before it was originally intended to.

Security solutions that are known to be highly vulnerable and can be easily broken by a quantum computer include:

1. Cryptosystems RSA, DSA, DH, ECC, ECDH, ECDSA and other variants of these ciphers.
2. Any security protocols that derive security from the above public key ciphers.
3. Any products or security systems that derive security from the above protocols.

Virtually all security products and protocols today that utilize public key cryptography use the above mentioned ciphers. The table below highlight the impact of quantum computing on effective key strengths of some popular ciphers.

Algorithm	*Key length*	*Impact*
RSA-1024	1024 bits	0 bits
RSA-2048	2048 bits	0 bits
ECC-256	256 bits	0 bits
ECC-384	384 bits	0 bits

Table 1 - Impact of quantum computing on effective key strengths.

Fortunately, not all types of cryptographic keys are breakable by quantum computers. Moreover, symmetric cryptographic algorithms in use today are perfectly safe to use. The most widely used symmetric cipher, AES, is one of the ciphers considered quantum-safe. A quantum attack merely weakens AES by reducing the key size, in contrast to asymmetric ciphers whose keys are completely compromised. Furthermore, AES adapts to a quantum attack by increasing its key size to rectify the threat introduced by quantum computing.

Symmetric key ciphers like AES are believed to be quantum-safe, whereas many public key ciphers like RSA are known not to be. A protocol that relies exclusively on ciphers like RSA is vulnerable to quantum attacks, but a protocol that can adapt to use quantum-safe ciphers is in itself considered quantum-safe. In protocols and applications where public key cryptography is preferred over symmetric key cryptography (usually to overcome the difficulty of key distribution and key management problems), quantum safe cryptographic ciphers must substitute RSA or ECC in order to resist quantum attack.

1.4 QUANTUM KEY DISTRIBUTION

Quantum Key Distribution (QKD) is a method used in order to produce a perfectly random key which is shared by a sender (*Alice*) and a receiver (*Bob*) while making sure that nobody else has a chance to learn about the key, e.g. by eavesdropping the communication channel used during the process.

In reality, *Alice* and *Bob* use two channels: a classical public channel and a quantum channel. First, *Alice* sends qubits to *Bob* over the quantum channel e.g. an optical fiber. *Bob* is measuring those qubits, obtaining a sequence of bits. The sequence depends on a coding system between two parties and chosen measurements which have random characteristic. Then *Alice* and *Bob* communicate over the public channel, in order to agree or disagree on *Bob*'s received bits.

The security of QKD relies on a fundamental characteristic of quantum mechanics: the act of measuring a quantum system disturbs the system (Heisenberg uncertainty principle). Thus, an eavesdropper, *Eve*, trying to intercept a quantum exchange will inevitably leave detectable traces. *Alice* and *Bob*, can then decide to discard the corrupted information. Once *Alice* and *Bob* have agreed upon the shared symmetric encryption key, they use the normal channel to transfer the data encrypted with the key.

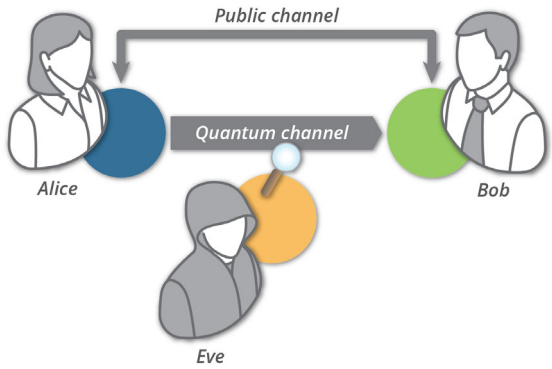

Figure 3 - Quantum key distribution.

Since QKD relies on a property of physics, secure communication is achievable when symmetric encryption keys are transmitting on a quantum channel. However, currently there are still some limitations of QKD. One of the limitations is the need of authentication of the communicating parties in order to prevent man-in-the-middle attack. That is, if *Alice* and *Bob* are to establish a secure communication channel, there needs to be a way for *Alice* to verify whether she really is communicating with *Bob* or whether *Eve* is pretending to be *Bob*. *Bob* needs to prove his identity by signing his communications so that *Alice* can verify his signature using his public key. Therefore, signing is a requirement for secure QKD. Another, practical, limitation of QKD is the distance limitation of about

250 km. One of the reasons for this limitation is because quantum repeaters do not currently exist.

2. FUTURE-PROOF SECURE COMMUNICATION

Previous chapters highlighted that essentially all communication systems today use asymmetric encryption algorithms for secure key distribution, even though they are known to be highly vulnerable and can be easily compromised by a quantum computer. Quantum Key Distribution is often mentioned as a possible supplant when its limitations have been sorted out. This leaves us with the obvious question: is it at all possible today to realize truly future-proof secure communication?

In practice, future-proof implies quantum resistance. Symmetric cryptographic algorithms possess this particular property. Thus, the question is if symmetric algorithms can be used for secure key distribution.

Suppose once again that *Bob* and *Alice* want to communicate using symmetric cryptography. They have never met and thus have not established a shared secret key in advance. How can they agree on a secret key? A solution is to use a trusted Key Distribution Center (KDC). The KDC is a server that shares a different secret symmetric key with every user. The KDC knows the secret key of each user and each user can communicate securely with the KDC using this key. When *Alice* and *Bob* are users of the KDC they only know their individual key, *kAlice* and *kBob*, respectively (see the figure below). Using *kAlice* to encrypt her communication with the KDC, *Alice* sends a message to the KDC saying she wants to communicate with *Bob*. We denote this message, *kAlice (Alice,Bob)*.

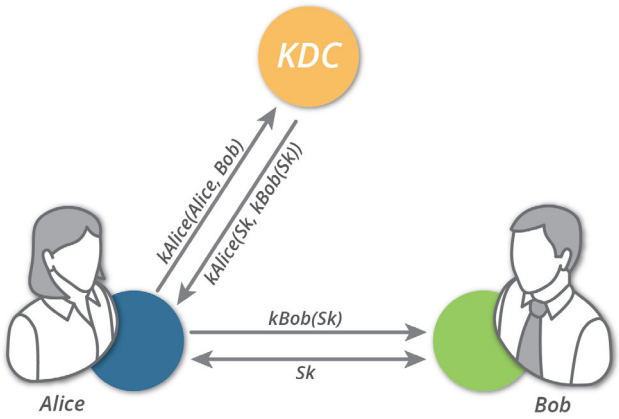

Figure 4 - Key Distribution Center.

The KDC, knowing *kAlice*, decrypts *kAlice (Alice,Bob)*. The KDC then authenticates *Alice* and then generates a session key, *Sk*. This is the shared key that *Alice* and *Bob* can use to perform symmetric encryption. As the name implies, *Alice* and *Bob* will use this key for only this one session. The KDC now needs to inform *Alice* and *Bob* of *Sk*. The KDC thus sends back an encrypted message to *Alice* containing the *Sk*. In reality the message also contains the *Sk* encrypted with *Bobs* key, *kBob*. The message from the KDC to *Alice* is thus *kAlice(Sk,kBob(Sk))*.

Once *Alice* receives the message from the KDC, she extracts *Sk* from the message. Now she knows the one-time session key, *Sk*. *Alice* also extracts the session key encrypted with *Bobs* key, *kBob(Sk)*, and forwards it to *Bob*. He decrypts the received message using *kBob*. Now *Bob* also knows the one-time session key, *Sk*. Since both *Alice* and *Bob* have the

secret session key, they can use it for secure communication protected by symmetric encryption algorithm.

Of course, the above described method is grossly simplified in order to highlight the very core essence with a Key Distribution Center, secure key distribution can be realized with symmetric encryption algorithms.

Today, there are only a handful commercial solutions utilizing KDC for secure key distribution. Advenica's network encryptor product line, SecuriVPN, is the only solution that utilizes a KDC specially developed to ensure long-term sustainable communication privacy. Unlike today's most popular and widespread technologies, that are no longer suitable to protect information that has extended lifetime, SecuriVPN is a perfect solution for a wide range of industries who rely upon the ability to keep sensitive information secure in the past, present and future.

advenica

Roskildevägen 1
SE-211 47 Malmö, SWEDEN

Phone +46 40 60 80 401

E-mail helpdesk@advenica.com
URL www.advenica.com