

Key Structures and (Systems for) Key Information Handling in Virtual Private Networks (VPN)

Table of content

1.	Introduction	1
2.	Systems with Public (PKI) and Symmetric Key Distribution	2
2.1.	Advantages of Symmetric Key Distribution against PKI Systems.....	2
2.2.	Advantages of PKI against Systems with Symmetric Key Distribution	3
2.3.	What is the choice?	5
3.	Key Structure and Key Control System in ViPNet Technology.....	5
3.1.	General description of the ViPnet virtual network structure defining the Key Structure.....	5
3.2.	Symmetric Key Distribution Subsystem	6
3.3.	Subsystem of public key distribution.....	7
3.4.	General Functions of the Key Distribution Technologies in ViPNet.....	8

1. Introduction

The various methods providing information security to corporation networks can only be efficient in certain conditions. The modernization (and thus the complexity) of operating systems and applications is constantly moving on eventually leading to today's inability to guarantee the immunity to all new types of network attacks aimed on undermining the security policy or gaining access to classified information including key, user and password information, especially if the attacking source is hidden.

Any corporation network is possibly exposed to internal threats. In comparison to outside threats, such attackers usually have both motivation and an aim, whereas attempts to gain access from outside are often of a more random character. A successful attack from the inside usually results in much more damage than external attacks.

Another difficulty is to provide security to mobile users, who use their laptop to connect to the network from the most various places. The use of wireless technology (Wi-Fi) is impossible, unless you take precaution measures by using a reliable security solution securing each computer individually.

In this situation it is very difficult to guarantee security by only encrypting the different types of traffic on the application or even more lower levels (like SSL, TLS ...) or by only using network firewalls and intrusion detection systems deployed on the borders of

the networks.

The only way to provide a high level of security in such an environment is the implementation of the highest possible level of control over all incoming traffic and its encryption when communicating with other computers. This would provide a solution to the main two difficulties: it will not be possible to hide the source of an attack – resulting in the maximum risk for the attacker(s) to be discovered and, a highly reliable mechanism to filter traffic cryptographically.

Such a solution is provided by the virtual private network technology (VPN), working on the network layer providing encryption to any traffic between separate computers as well as networks, and which are integrated with a personal and network firewall. Such technologies provide the maximum possible level of traffic control, incoming from the network regardless of the location/source and type of attack. There are different definitions for a VPN. We will concentrate only on solutions providing transparent traffic encryption regardless of the application used.

This is also the reason why VPN solutions appear more and more often throughout the world. These solutions are primarily based on the IPSEC protocol suite, which became a de-facto standard mainly because it can be implemented throughout a wide range of computer technology and hardware appliances.

And this is the point when we come to

the main question of this article: which key structure is more suitable for setting up virtual private networks with encrypted circulating traffic. As you may know, IPSEC is a multipoint protocol, defining many different questions needed to organize secure connections. IPSEC is oriented on the PKI technology with public key distribution. The protocol also allows the use of symmetric keys however, without providing a solution for their distribution and use.

Sadly, a couple of years ago, when

computers were not as developed as now, stereotype emerged about the distribution of symmetric keys being a supposedly difficult, hardly scalable and not automatable task. This led to systems implementing a public key infrastructure being more popular, including in VPN technology, without looking at the problems occurring with the use of a PKI. We will try to compare the two key distribution (and thus cryptography) philosophies on the following pages.

2. Systems with Public (PKI) and Symmetric Key Distribution

While looking on the key systems, we will primarily concentrate on methods to allow a confidential exchange of information, e.g. information encryption.

As a brief reminder: In systems providing a confidential link between two points based on symmetric key distribution, both participants need to exchange the key, which will be used to encrypt information prior to establishing a connection. In general, each participant will have his own set of keys to connect to other points.

Systems with public key distribution follow a different approach. Each user has an own pair of keys. One key is secret, generated by the user and has to be kept confidential. The second key, which is generated using the secret (private) key, is public and transferred to anyone else wishing to establish a (trusted) connection with the user (by using public storage like a webserver or during the communication process). However, there is a huge requirement for this key: the origin of it needs to be 100% guaranteed, e.g. you need to be sure that the key you just requested is truly from the user you are establishing a connection with. If it's not possible to be sure, there is also no confidentiality. As will be shown later, a reliable solution to this problem undermines the advantages of public key distribution. To establish a secure connection with a specific user, both sides use the public key of the other side and their own private key to generate a common symmetric key, which is then used to encrypt or decrypt the information sent between them.

2.1. Advantages of Symmetric Key Distribution against PKI Systems

2.1.1. It is commonly known that systems with symmetric key distribution are providing *many more levels higher resistance to intrusion methods*, than systems based on a public key infrastructure. Let's take the keys as an example: if somebody would want to reach the same level of security provided by a symmetric system with a key length of 128 bit (this is the absolute minimum used in symmetric systems today), a key length of more than 2000 bit would be needed in a PKI based system.

2.1.2. Encryption algorithms used in PKI systems, are based on the computing complexity of some classic mathematic tasks. *An unexpected breakthrough in mathematics, potentially at any moment, is able to corrupt the whole asymmetric cryptography.*

Systems with symmetric key distribution always incorporate a secret element (key), which is initially owned by both sides wanting to establish a secure connection between them. And this is exactly the cause, why symmetric systems are much more resistant to different analysis methods as well as being more stable. The algorithm DES is well known, however it was not breached by finding a hole in the algorithm, but by having the ability to try out all possibilities and eventually find out the key by using today's technology, which advanced heavily in comparison to the times DES was developed in. However, Triple DES (3DES) is not susceptible to „brute force“ attacks anymore and everybody is confident that it will stay like this for the next couple of decades. The same

is applicable to the Russian algorithm GOST 28147-89, which is counted as one of the most powerful algorithms of our time, which is almost as resistant to attacks and analysis as it was for already several decades.

It is known that it is not impossible that some relation between the keys in a key pair, or a weakness in an algorithm's operation, might be found which would allow decryption without either key, or using only the encryption key. The security of asymmetric key algorithms is based on estimates of how difficult the underlying mathematical problem is to solve. Such estimates have changed both with the decreasing cost of computer power, and with new mathematical discoveries.

Weaknesses have been found for promising asymmetric key algorithms in the past. The "knapsack packing" algorithm, for instance, was found to be insecure when an unsuspected attack came to light. Recently, some attacks based on careful measurements of the exact amount of time it takes known hardware to encrypt plain text have been used to simplify the search for the most likely decryption keys. Thus, the use of asymmetric key algorithms does not ensure security; it is an area of very active research discovering and protecting against new and often unexpected attacks. A relatively new addition to the list of asymmetric key algorithms is the elliptic curve cryptography. While it is more complex computationally, many also believe it to represent a much more difficult mathematical problem than either the factorization or discrete logarithm problems.

2.1.3. As a prerequisite to functioning properly, systems based on a PKI need special authentication sessions and key generation processes, which take their time. Besides, it is hard to synchronize the initialization of the application with the authentication session, which usually has to happen first. This is a crucial fact when choosing a VPN, which needs to be as transparent as possible to other applications (especially network dependant), because any delays may result in outages. This is exactly the cause, why there are almost no IPSEC solutions allowing client-to-client connections or which are suited for LAN use.

Symmetric systems, distribute the keys beforehand, this results in no special, long authentication sessions being needed; a connection can be established instantly, without

disturbing possible network services, etc.

2.1.4. The presence of special authentication sessions when using a PKI is hard to hide; this adds another surface to base attacks on raising the overall potential instability of a system. Disturbing just a few symbols from the relative huge amount of information, which is transferred in this process, leads to the inability to establish a connection. This means that most systems based on a PKI are vulnerable to easy hideable attacks.

Systems with symmetric key distribution, do not know such problems. If somebody would want to attempt a similar attack, the traffic of a whole session would need to be distorted, which on the other hand is easy to find out.

2.1.5. There is also a non-scientific argument: *No single governmental or army structure, as well as systems dealing with governmental or armed forces classified information is using a PKI to encrypt this information.*

2.2. Advantages of PKI against Systems with Symmetric Key Distribution

2.2.1. It is seen as the main advantage of a PKI system, that there is no need for a secret key transfer. However, this results in the need to have a secure and reliable authentication process which, if it is not secure enough renders the system insecure. This task is currently solved by implementing a full-scaled digital signature infrastructure, based on digital certificates, which are certified by a third party (certificate authority), which is trusted by all parties participating in the information transfer. In this case both sides wishing to establish a secure link to each other will sign their public keys with their digital signature during the authentication session. Only if the signatures are valid and the certificates are trusted, the symmetric session key is generated, which the traffic encryption is finally based on. The most various ways exist to authenticate using digital signatures; however, it is always needed to validate the digital certificates.

This results in all participants needing a digital certificate, to be able to establish a secure connection to another participant. Such a certificate is not kept secret; however it is required to validate the identity of the recipient

to obtain (at least the first) the digital certificate to guarantee that the certificate is owned by the intended owner and not somebody else. This can happen either by going personally to the certificate authority, or by means of a trusted and secure communication channel.

Additionally, the root certificates, which are used to sign all other certificates, need to be trustfully obtained and afterwards kept safe from attempts to replace the certificates by different ones, in which case the whole security system is compromised.

In the case of systems based on symmetric key distribution, it is also possible to receive a secret key, which can then be used to establish a trusted connection which is used to transfer all other needed keys.

This means, that both philosophies require a trusted way to receive initial key information. The only difference is that the symmetric variant requires you to keep the keys secret, and the asymmetric variant requires you to provide methods to authenticate and keep the information secure from replacement, which is a task of similar difficulty. Both ways require you to keep the private keys secret: in the case of a PKI – the key generated by the user itself, and in the case of a symmetric key infrastructure – the key obtained in the CA.

Furthermore, two users would need to receive and be able to validate the other side's certificate, which is a rather responsible task. There is a wide-spread anecdote about a group of people were able to receive a certificate on the name of Microsoft from a highly respectable certificate authority and used this certificate to put malicious programs on the computers of the internet users. This is explained easily by the fact that it is very hard to automate the validation of certificates, e.g. the validation itself is easy, however, it is up for the user to decide manually who owns the certificate, often resulting in errors from the user's side.

2.2.2. It is often stated, that a public key distribution system needs much less keys than a system using symmetric keys. This would be true, if no authentication procedures would be needed between two communicating parties. As a matter of fact, each computer needs to store the same amount of digital signatures as other users it will be communicating with. It is of course possible not to store the signatures and

receive them each time a connection is about to be established from a central storage place or directly from the other side, however this bears the effect for a VPN using such an approach that the authentication process takes longer resulting in some applications not working anymore, and, since certificates are rather long, this also results in additional network load.

In a symmetric key distribution system, the same number of keys is needed as user the computer will be communicating with. The security of these keys can easily be guaranteed by encrypting them with a single personal key. The difference to a PKI is that the certificates on all computers are the same, whereas the symmetric keys of each user are different. However, this is more a question of the key distribution approach, which will be explained later. *The size of a symmetric key is no more than 32 bytes long, whereas a certificate is much larger – 1500-2000 bytes.*

2.2.3. PKI systems have no "main" owner of the key information. The secret part of the key is formed directly by the user resulting in it known only to him and not the administrators.

In a symmetric system, the keys are generated in a central place and may potentially be used by the administrator to gain access to information transferred between users. Additionally, there is the risk of a compromitiation of the key center, which may well include the whole key set used by the system.

However, something similar can also happen in a PKI system. The administrator of the certificate authority is capable of issuing a certificate for any user, and similar to the symmetric system send out information in the name of somebody else, or perform a "man-in-the-middle" attack intercepting information transferred between users.

In the case of the keys in the center being compromised, the PKI would win undoubtedly, since the certificate authority does not keep any keys which could help decrypt information. From this point of view, systems implementing both symmetric and asymmetric key distribution (introduced later in this document) would have a huge advantage, however in such a case the PKI system is enormously simplified, since it does not require the many security precautions needed for such systems anymore.

2.3. What is the choice?

2.3.1. The previous reasoning makes clear, that both philosophies have its drawbacks.

Systems with Public Key Distribution:

- Potentially have very serious security issues in the far or maybe near future
- Require very close attention from the user to assure that the certificate is owned by the person the user wants to communicate with and not by a person with a similar surname
- Require a substantial amount of time for the authentication process on each connection establishment. This leads to a decreased transparency of the VPN technology for any network applications (some not functioning properly because of the delays)
- Are easily attacked by means of communication disturbance
- Force the use of VPN technology mainly through remote access through the VPN server which a connection was established with before. In any case a full blown PKI is a difficult and costly task, when done implementing all security measures

Systems with symmetric key distribution are free of these side-effects if a reliable system for key control is available. The only negative point is the lack of sufficient defense for the key center, if the case of the key compromitation is counted as possible.

2.3.2. The above problems can be solved by

using a hybrid key structure making use of both symmetric and asymmetric keys.

The base module is a centralized or distributed management center subsystem managing the symmetric key structure of the corporate network, providing a high level of network stability and a reliable way to manage the key structure.

Another module is the „public“ key distribution subsystem. This system functions under the protection of the symmetric key structure rendering the above problems unimportant. Because authentication sessions are not needed anymore, each connection is established momentarily, which is a very important point in VPN technology. Additionally it becomes possible to generate keys not know to the administrator on a regular basis. This makes it impossible for the administrator to gain access to user information and lowers the effects of a possible key compromitation in the center. In this case the “public” key exchange session is hidden in the general encrypted traffic and becomes very difficult to attack.

As a result of the operation of the two subsystems a communication key is generated for the corresponding network nodes, which the traffic encryption is based on.

The public key distribution subsystem also implements all standard mechanisms needed for digital signatures, which also work under the protection of the symmetric key distribution subsystem. The digital signature is using the same algorithms as the encryption in the public key distribution, which also render the above issues with digital signatures unimportant

3. Key Structure and Key Control System in ViPNet Technology

3.1. General description of the ViPnet virtual network structure defining the Key Structure

The creation of ViPNet network objects, the connection between them, the generation of symmetric key information, generation of the initial key distributions, remote update and replacement of key information, centralized generation of digital signatures and the issueing of digital certificates for the public keys of the digital signature, generated on the

nodes is, handled by the ViPNet[Administrator] application, which is part of the [Network Control Center] and [Key and Certificate Authority] (key center). The network control centers of different virtual networks can interoperate with each other to organize a secure cross-network interaction for their own network's nodes.

3.1.1. ViPNet network objects, which are registered in the NCC are:

- Network nodes (basically, these are computers) includes a keyset for the

communication with other nodes

- Groups (a group of users of a network node; several groups may be registered on a network node) – includes a keyset for the communication with other groups
- users (basically the users of the computer, who can be registered in several groups on several nodes) – have the ability to receive digital certificates, own the group keys, including the keys of the network node level

A typical situation is a single user per computer. In this only the network node needs to be registered; the corresponding group and user will be created automatically.

3.1.2. The interaction possibilities with information resources or other users, i.e. the presence of the corresponding keys in the users' keysets, are defined by the configuration of the connections between groups in the NCC. Two network objects may only interact if the corresponding connection is allowed in the NCC.

The creation of several groups on a network node only makes sense if the computer is used by more than one user and there is the need to separate access between users. If the document exchange needs to be legally reusable, each user has the ability to generate an own digital signature and obtain a digital certificate from the ViPNet certificate authority.

There are many more possible options available like the creation of subgroups, hidden groups and much more; this topic is not covered in this document.

3.1.3. A unique identifier is assigned for each created object by the NCC. After the removal of an object, the identifier is not put back into the pool of available identifiers. The identifiers fully determine the registered network objects, guarantee their individuality, allow building the whole symmetric key infrastructure for object interaction, including objects from different virtual networks, and authenticate users during their logon process on specified network nodes.

3.1.4. As was said above, each user has the ability to obtain a digital certificate, which can be used for different means: guarantee the legal reusability of documents, by signing them, organize the access to concrete information

resources and other tasks. We will not go into more detail about this subsystem; however, we would like to point out that the security of this subsystem is also guaranteed by the symmetric key distribution system.

3.2. Symmetric Key Distribution Subsystem

3.2.1. The task of this system is to generate a symmetric exchange key for each connected pair of groups on the network nodes. Connection information is transferred from the NCC to the KC in the according directories. Of course the key center does not save all created keys; instead it generates a set of keys for each object of the network on demand. These keys for a given group pair are based on encrypting the corresponding pair of identifiers with a master key.

The master key (32 bytes) is kept in the key center at maximum possible confidentiality, since a compromised master-key results in all keys being compromised, which are derived from the master key. *It's cryptographically proven that it is not possible to compute the master key from keys derived from it, since the same sym. encryption algorithms are used for generation.*

Each network node (the groups of a node) receives the needed *exchange keys* with other nodes (groups) encrypted on the security key. *Security Keys* are basically exchange keys, used by the key center to perform exchange key updates for the group of a given node.

The security key is encrypted on the personal keys of the users. Personal Keys – are basically exchange keys between the key center and the specific user of a given node, it is used to separate access to the available collectives of the nodes, the user is registered on, as well as for updating the security key of each user.

3.2.2. The Master – key is generated using a random number generator (hard- or software) and implements an echeloned cryptographic protection system for the safe storage and use of the key. For the given virtual network, three master keys are generated, each 32 bytes long and needed to generate the key of the following levels:

- The mentioned above *group exchange keys*

- *Security keys of the exchange keys* which are used to encrypt the exchange keys when transferring them to the corresponding nodes
- *personal user keys*, used to encrypt the security keys of the exchange keys and other personal key information including the private keys of the digital certificate

3.2.3. *The set of exchange keys for a group is encrypted on the according security keys of the exchange keys*, which is unique for each group and can thus be freely transferred through the communication channels and saved on the hard drive of the according node. Since the length of a symmetric key is not big, the transfer and storage of a key set for a given node is not a problem at all. If we take a key set consisting of 1000 keys, it only takes 40-50 Kbyte not counting system information.

3.2.4. *Security Keys of the Exchange Keys* are stored on the harddrive in the personal directory of a user. The number of these keys equals to the number of groups, a user is registered in. Usually it is one key of 32 bytes length. The same folder contains the root certificate directory of its network and keys needed to initialize the random number generator. *Security keys of the exchange keys are encrypted on the personal keys of each user* and can also be transferred through the communication channels when updating key information.

3.2.5. *The personal keys of each user are encrypted on the according password key* and stored on the user's personal key storage container (i.e. floppy, smart-card, touch memory tablet, USB dongle, etc. in some cases also stored on the harddrive).

3.2.6. *Password key* – a sequence of bytes, 32 bytes long, derived from caching the password.

3.2.7. *Password* – a sequence of alphanumeric characters between 9 and 32 characters long. A password may be either random or personal. A personal password is set up by the user itself or the administrator of the key center, whereas a random password is generated from a so-called random, easy to remember password phrase consisting either of 3 or 4 words. During password generation it is also possible to select

how many characters from the beginning of each word of a phrase should be taken to form the password (3 or 4).

3.2.8. The encryption of traffic between two objects is realized using the exchange keys, which are always ready for use. Actually, the encryption of each block of information is accomplished using derivated keys:

- Either by the random key encrypted on the exchange key
- Or by caching the exchange key and
- This allows performing exchange key updates once a year

3.2.9. The terms key number and key variant are used to allow the update of keys after expiration or a compromitation.

The key number is changed when modifying the master key, which is used when updating all keys of a virtual network, relying on the given master key.

The key variant is used to update keys on a single network object. In this case, all keys of an object, and the according keys of other objects connected to it are replaced.

3.3. Subsystem of public key distribution

As an option, each node may turn on the use of public key distribution subsystem for any other network node. In this case, each period of time (default is 30 days), an own pair of asymmetric encryption keys is generated: a public and private key. The new key, signed by the digital signature of the user, which is bound to a digital certificate of the certificate authority, is sent to the network nodes, the asymmetric subsystem is turned on for. Based on the public keys of their objects with a valid signature and certificate, and the own private key, an additional symmetric exchange key, which is valid until one of the side generates a new pair of keys. In this case, the resulting exchange key for two objects is generated as a "cryptographic wrap-up" of two keys: the symmetric exchange key generated in the KC and the symmetric key, generated by using the asymmetric keys of the public key distribution protocol.

The replacement of a key does not interrupt active communication sessions, since

our special technology will check that the new key is present on both sides first before switching to the new key.

3.4. General Functions of the Key Distribution Technologies in ViPNet

3.4.1. The described above multi-level symmetric key structure of the virtual network, set up using the ViPNet[Custom] package, provides the ability to set up a highly scalable, secure symmetric key distribution system as well as systems controlling the possible connections between users, able to limit access for information resources.

The symmetric key distribution is completely automatic and does not require user interaction.

The automatic asymmetric key distribution subsystem protected by the symmetric key distribution system, provides the protection of network objects against possible compromitations of the central administrations.

3.4.2. A node may only connect to the virtual private network, if it was registered in the network control center along with the according users and the required connections, and a set of keys was generated, which include the minimum key set to be able to interact with the key center, network control center and the node's coordinator.

The key information in this distribution is protected by the personal key, which is encrypted on the password key itself. The personal key may be part of the key distribution file or stored separately on a personal medium.

3.4.3. After receiving the distribution file, the ViPNet application can be installed on the computer. The computer will then instantly be able to connect to the virtual protected network and interact with other nodes of its own or other virtual networks using any applications, according to the connections allowed in the NCC (e.g. all objects, a corresponding keyset is available for).

3.4.4. If additional network nodes are needed, the new nodes need to be registered in the NCC. This will result in new keys being generated in

the key center and sent out to the affected nodes along with the access directories automatically. Before any key information is sent out to a node from the NCC, it is encrypted with the communication keys of the NCC and the corresponding node; the information is then sent out to the node through existing VPN tunnels (either through coordinators or directly). After receiving new key information, a node automatically updates its existing key database.

A similar process takes place when removing connections. In this case, the nodes remove excess information from the key database.

3.4.5. If it becomes available, that a network node was compromised, all necessary steps are taken in the NCC, which is used to flag the object as compromised. All information needed by the key center is then generated automatically and new keys with a new variant are formed. The new key information is then sent out to the according nodes the normal way from the NCC and then updated automatically.

3.4.6. It is very important to ensure a synchronous update, when updating keys, especially the master key, and ergo all keys derived from it. To achieve this, an information update interval is set on each node, which can be defined in the NCC, which can then also be used to see the update status of each node. The nodes are configured to provide compatibility to the previous active keys for a period of time. These features allow the uninterrupted work of the vpn, even when doing mass key updates.

3.4.7. The asymmetric encryption keys between nodes are updated automatically in defined time intervals, without user or NCC interaction.

3.4.8. The system will warn about the need to update the certificate keys in time. The update is either accomplished from a central place by generating the corresponding keys and certificates in the certificate authority, or by each user, by putting the private key on the media and automatically obtaining the certificate for the public key from the certificate authority.