A Survey on Automatic Configuration of Virtual Private Networks

Michael Rossberg, Guenter Schaefer

Telematics/Computer Networks Group
Ilmenau University of Technology
Germany

Abstract

Virtual private networks (VPN) offer a secure data exchange over public networks. Despite being cheaper than leased lines, growing sizes and dynamic behavior of VPN nodes, e.g., for mobility or reasons of denial-of-service-attacks, make a manual configuration of large, dynamic VPN expensive.

Consequently, a number of different VPN auto-configuration approaches have been invented and partially deployed over the last decade. This article identifies a comprehensive set of objectives to be fulfilled by IP-based VPN auto-configuration, explains and groups mechanisms, and analyzes their strengths and weaknesses with regards to the objectives. Finally, it identifies potential future directions of autonomous VPN deployment.

1. Introduction

Over the last decade, the Internet has advanced to a low-priced and globally available communication medium. So it is only a consequence that companies and governmental institutions are changing their strategy and switch from dedicated leased lines to the more open and more flexible, as well as cheaper paradigm of communicating even internal, possibly confidential information via the Internet. Additionally, the Internet also raises the desire of geographically distributed communities without large funding for secure, cheap communication and exchange of files.

Both scenarios can be supported by the creation of virtual private networks (VPNs) overlay networks on top of the Internet Protocol (IP) layer. Every participant in such a VPN is given a certificate or password that enables him to securely communicate with others by presenting a compatible certificate or the same password. Security in the sense of VPNs primarily concerns confidential data transmission, but often also integrity protection and authentication.

Even though security is naturally handled very well by VPNs, many operational problems remain: Where shall a VPN device connect to? Which VPN device represents which IP address range within the VPN? Over which path data shall be relayed through the VPN, if no direct connection through the network exists? — All of these questions must be covered by a VPN configuration mechanism. However, VPN standards like IP security (IPsec), Transport Layer Security (TLS), and Point-to-Point Tunneling Protocol (PPTP) do not address the configuration from a macroscopic point of view, but rather rely on the static, manual configuration of each VPN association.

This manual configuration approach has several drawbacks. First, the administrative overhead grows quadratically with the number of VPN devices, if each VPN device shall be able to communicate with every other VPN device. This will not only lead to higher expenses, but also to more errors caused by human failure. Second, the robustness of the VPN is not as high as it could be, e.g., in case of partial failures of the transport network some VPN devices could redirect traffic for other devices that cannot reach each other directly anymore. Even though IPsec could support such a resilient behavior by utilizing nested security associations, a manual reconfiguration prohibits a timely reaction. Third, manually configured security associations cannot be adopted with sufficient flexibility to support mobile VPNs appropriately. It is not possible to just configure security associations between two mobile devices as both regularly change their external IP addresses.

The large administrative overhead and the limited flexibility of manual configuration approaches lead to a demand for the automation of VPN configuration. Several entire differently structured mechanisms try to fulfill this need. However, each one is focusing on special scenarios.

Thus, the contribution of this tutorial style article is threefold. First, the article motivates and describes a consistent set of general, unbiased, and comprehensive objectives for automatic configuration of VPNs. Second, a number of diversely structured configuration approaches is systematically presented and evaluated following these objectives. We especially focus on the evaluation of security targets, and identified sevreral potential weaknesses. Finally, current open issues in VPN auto-configuration are derived from the presented approaches and discussed.

All presented configuration approaches focus on IP-based Customer Edge VPNs, and build overlay networks. VPNs that make use of provider support, e.g., Multiprotocol Label Switching (MPLS) VPN, are configured transparently to the customer, so that they do not have to perform the configuration themselves. Providers on the side require a set of entire differently structured objectives to be fulfilled. Therefore, re-

garding end-user auto-configuration approaches it is plausible to simply focus on overlay VPN. Furthermore, approaches on data link layer are not discussed, i.e., for VLAN configuration. These have very different objectives and may use local broadcast mechanisms for example.

The rest of this tutorial style survey is organized as follows: In section 2 we present details on common use cases, the resulting network topologies of VPNs and utilized protocols. The following section 3 covers the objectives for automatic configuration techniques, which will lay the foundation for the description and analysis of several systems in section 4. We close the article with an overview of open issues in section 5 and a conclusion in section 6.

2. VPN Deployment Scenarios, Topologies & Protocols

As many of the presented auto-configuration approaches have a certain scenario in background, common use cases for VPNs and the corresponding VPN protocols will be presented in this section.

2.1. Deployment

Depending on the type of organization of the VPN operator the following three use cases are known.

2.1.1. Commercial & Non-governmental organizations

Especially for smaller companies the main application of VPN is the inclusion of remote workers, sometimes referred to as "road warriors" or nomadic nodes. The resulting topologies for such *remote access scenario* are quite simple (see Figure 1): each of the nomadic nodes connects to a dedicated VPN access concentrator. All intranet traffic is tunneled to the access concentrator by the nomadic node and vice versa. This includes the traffic between nomadic nodes themselves so that this traffic is always relayed and re-encrypted in the access concentrator.

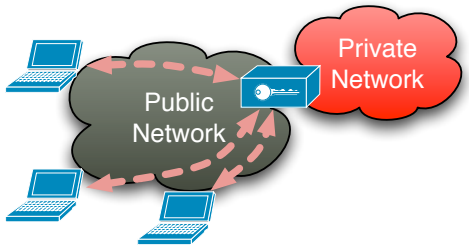

Figure 1: A remote access scenario to connect mobile nodes

Another use case for VPN, often found in larger organizations, is the *site-to-site* VPN that creates and maintains virtual connections between different locations, such as company sites. In contrast to nomadic node scenarios these are mostly static, that is their Internet connection does not change in terms of public IP address, bandwidth, delay, etc. Furthermore, the VPN gateways connect whole networks, and do not simply integrate single hosts. With regards to security the participating VPN devices are usually considered to be more trusted than nomadic nodes as they are run in a more secure environment. Nonetheless, according to [1] about half of the examined VPN follow a similar centralistic paradigm as outlined in the nomadic node scenario. In this case the different sites connect to a VPN concentrator in the headquarters, resulting in a so-called *Hub and Spoke* architecture like seen in Figure 2. As the inclusion of a VPN gateway only involves the creation of a security association between the gateway and the central VPN concentrator, this architecture is often considered to cause the least management overhead.

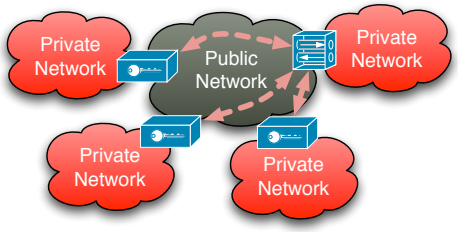

Figure 2: Hub and Spoke architecture to connect private networks

The other extreme case for *site-to-site* scenarios is a *full mesh* topology (see Figure 3), where each site is connected to each site individually. Thus, it involves the creation of $O(n^2)$ security associations. However, it offers usually better delay characteristics and avoids a single point of failure (SPoF).

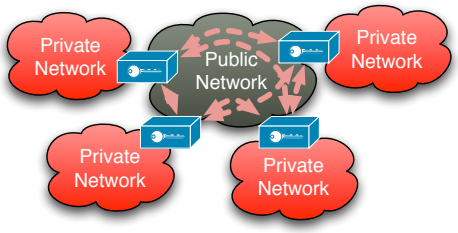

Figure 3: A fully meshed VPN

Between these two extremes *decentralized* architectures, like dynamic multipoint VPN [2], exist. In this case multiple hubs are interconnected by a mesh of security associations, and each dynamic gateway creates a single association to one of the hubs.

Further special cases may exist. For example, a company may decide to protect its research department from the rest of the intranet by another VPN gateway, and thus create a *nested scenario*. Failover objectives may require multiple VPN gateways to be responsible for a single trusted network.

2.1.2. Governmental & Military VPNs

Governments and military organizations mostly rely on VPNs with use cases that are similar to civilian networks. However, as they operate own networks [3] they are, for example, able to utilize multicast communication within their underlying transport networks.

Furthermore, for disaster relief communication and tactical networks (see Figure 4) [4, 5, 6] they may also rely on mobile ad-hoc networks (MANETs), which are highly dynamic

mesh networks created by soldiers or members of emergency services. Due to the nature of these networks, they must incorporate distributed auto-configuration mechanisms that need to be able to cope with high churn, bad radio conditions, and potentially compromised VPN devices.

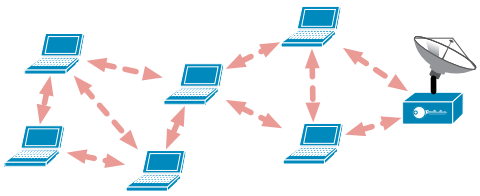

Figure 4: Tactical mesh network with satellite uplink

2.1.3. Private Individuals

A common use case for individuals is the utilization of VPN services to connect to home networks from the road, not differing from the commercial *remote access* scenarios [7] as already pictured in Figure 1.

A still growing trend is the use of VPNs for secure file sharing within communities. This may happen by using either conventional VPN techniques, i.e., tunneling traffic to a trusted provider [8, 9], or by utilizing VPN configuration mechanisms to autonomously establish peer-to-peer links between community members.

2.2. Cryptographic Protocols

While the main concepts of VPN auto-configuration mechanisms are usually rather independent from the used cryptographic protocol, the choice may have a severe influence on the overall properties of the resulting VPN. Therefore, a short overview of common security protocols is given in the following.

2.2.1. SSL/TLS/DTLS

One of the most successful security suites is TLS [10], which was originally based on Netscape's Secure Socket Layer (SSL) protocol. Being originally designed to provide end-to-end protection between applications on different hosts, TLS is implemented in user space libraries and integrates between main application logic and the Transmission Control Protocol (TCP) service access point. Nonetheless, programs like stunnel [11] provide means to encrypt arbitrary application traffic using SSL and TLS. While this is often done by an application specific port forwarding, some more advanced VPN tools such as CloudVPN and OpenVPN use so called TAP devices to capture generic traffic by user processes and transmit it securely over a TLS connection to other hosts. Due to this approach traffic may be transparently protected.

As TLS depends on TCP connections, it is not suited for all scenarios, e.g., retransmission mechanisms make it difficult to transport real-time traffic, and in Network address translation (NAT) situations it is not possible to perform "hole punching" with TCP. Therefore, Datagram Transport Layer Security (DTLS) was specified [12, 13], which transfers basic TLS functionality to the User Datagram Protocol (UDP). In order to cope with replays and dropped packets, DTLS implements sequence numbers and replay windows, and apart from the fact that it is implemented in user space, it is somewhat comparable to IPsec, which will be presented in the following.

2.2.2. IPsec suite

In contrast to TLS, IPsec [14, 15] is implemented at the network layer and designed to be quickly processed and forwarded not only in end systems, but also in intermediate security gateways. Hence, IPsec is predestinated to be used when multiple subnets need be connected in site-to-site scenarios. One of the most often cited disadvantages [16] – IPsec's complexity – becomes more and more negligible as

- with the introduction of Internet key exchange (IKE) version 2 [17] major simplifications with regards to IPsec key exchange were introduced,
- the implementations within operating systems and keying daemons have become mature and interoperable,
- even now after over ten years since initial standardization no practically relevant security vulnerabilities became known (in contrast to SSL, TLS [18], and many proprietary protocols),
- and the critic of complex deployment lead to many of the auto-configuration schemes presented in the following.

2.2.3. Other Cryptographic Schemes

Besides TLS and IPsec several other cryptographic protocols for VPN operation exist; perhaps best known is the Point-to-Point Tunneling Protocol (PPTP). Nonetheless, even though PPTP is still used in many infrastructures, it is considered obsolete and will not further be covered in this article as multiple severe security issues [19] have been identified.

Layer Two Tunneling Protocol (L2TP) [20] represents a protocol, which operates at data link layer and just like PPTP, it also allows to establish user authenticated tunnels over an untrustworthy network. In contrast to PPTP, L2TP requires a mandatory IPsec protection layer, when used over unsecured networks. Thus, often it is solely used for accounting purposes within VPNs (e.g. [21]).

Just like the previously mentioned stunnel, can the Secure SHell (SSH) protocol also create tunnels for higher layer applications by implementing a connection specific port forwarding [22]. Due to the use of TCP SSH is also not suited for the transport of delay critical data and usually not considered for realizing generic VPNs.

Many of the remaining cryptographic protocols that will be mentioned in the following (e.g. Wippien, WASTE, P2PVPN, N2N, and Hamachi[2]), raise serious doubts concerning the security of proprietary protocols. Without standardization efforts they will stay niche solutions for reasons of general support and security. They are presented for the sake of completeness and basic understanding, rather than to suggest them as a solution for actual VPNs.

3. Objectives & Challenges

Depending on the envisaged scenario a multitude of objectives must be fulfilled by an automatic configuration mechanism for VPN. We group them in functional, non-functional, and security objectives and present them in the following three subsections.

3.1. Functional Objectives

This first group of objectives focuses on particular tasks a configuration mechanism has to perform.

- **Simple configuration:** An automatic configuration approach should minimize human intervention. Ideally this reduces the administrative overhead to installing a certificate or setting a password.
- **Gateway functionality:** A comprehensive configuration approach must be able to not only include single hosts, such as nomadic nodes, in a VPN, but also interconnect smaller networks to a large VPN.
- **Private IP address ranges:** For reasons of easier management or security VPN often use private IP address ranges (such as 10.0.0.0/8) internally. VPN configuration mechanisms should allow the use of such addresses; thus, they need to include an internal routing mechanism.
- **Nested security associations:** It should be possible to create associations between indirect VPN devices, e.g., VPN devices that are only reachable through other VPN gateways for security reasons, for reasons of partial connectivity failures, or because communication takes place in a multi-hop MANET.
- **No special support from transport network:** The configuration mechanisms should make use of normal unicast traffic only, that is it should not rely on IPv6, multi-, any-, or broadcast support of the transport network.
- **NAT:** Especially in remote access scenarios, VPN devices join the network over NAT routers. This may imply address conflicts that have to be taken care of by VPN autoconfiguration mechanisms. Furthermore, it is desirable that connections between VPN devices can be negotiated, even though both are behind NAT routers.
- **Quality of service:** If the transport network offers quality of service (QoS), applications inside the VPN should be able to make use of them transparently. This does not only include the use of priority queues within VPN gateways, but also requires the handling of QoS specific protocols and protocol extensions, i.e., the Differentiated Services Code Point (DCSP) field in the original packet's IP header must be passed to network routers. To make use of integrated services even more sophisticated reservations mechanisms must be deployed as reservations within the VPN also have to be made in the transport network.
- **Multicast within VPN:** It should be possible for VPN applications to make use of IP multicast extensions transparently.

3.2. Non-functional Objectives

The presented mechanisms are expected to not solely configure VPN of the sketched topologies, but also to perform measurably well. Relevant criteria are:

- **Robustness:** The configuration system must react to failures within the VPN itself, but also in the transport network in a resilient way. This includes avoiding a SPoF due to a centralized algorithm, and ensuring a quick recovery time, if parts of it were restarted or a VPN was partitioned. Additionally, routing mechanisms within the VPN should allow for a recovery of partial failures of the transport network [23].
- **Scalability:** In order to handle the number of devices in VPNs of large organizations, an auto-configuration mechanism must be able to handle up to thousands of independent sites or remote workers.
- **Efficiency:** The efficiency of a configuration algorithm can be measured in two ways: it shall create on the one hand a low overhead in terms of sent and received messages to save bandwidth, and on the other hand it shall cause as few security associations to be created proactively as possible in order to save computing resources, which is especially critical if smart cards are used for asymmetric cryptographic operations.
- **Reconfiguration Speed:** Whenever the topology of the VPN changes, i.e., due to mobility of VPN devices or in reaction to network failures, the configuration mechanism must bring the VPN back to a stable operation as quickly as possible. The reconfiguration time primarily depends on the number of security associations to be created within the VPN.

3.3. Security Objectives

As VPNs usually transport sensitive data, and play a vital role in the internal communication of organizations, their overall security is crucial to their success.

- **Confidentiality:** Perhaps the most prominent objective of VPNs is the transparent encryption of packets, thus, ensuring the confidentiality of data. However, there are several other subgoals to be achieved:
  - **End-to-End protection:** All data that is passed indirectly between VPN devices, e.g., first sent to a hub, can be required to be protected by multiple layers of encryption. This ensures the confidentiality of data, even if parts of the VPN are compromised. Though, a direct consequence of this option is the impossibility to run central monitoring and intrusion detection systems.

  - **Perfect-Forward-Secrecy (PFS):** In many VPNs session keys need to be derived from a Diffie-Hellman key exchange [24], which guarantees the secrecy of the communication, even if the authenticating master secret is compromised later on.
  - **Covert-Channel Resistance:** VPN gateways are usually considered to be policy enforcement points that ensure all incoming and outgoing data is encrypted and sent to verifiable communication partners. In order to fulfill this challenge, communication channels of potentially compromised devices within the trusted network to the outside and vice versa must be prevented.
  - **Infrastructure Hiding:** External observers of VPN traffic should not be able to derive the structure of the networks behind VPN gateways or discover a large number of VPN devices as it would allow them to plan more sophisticated attacks.
  - **Anonymity:** In some (less common) VPN the confidentiality of the identity of participants shall be secured, thus, transmitted data needs to be anonymized. As anonymity requires enormous trade-offs with regards to efficiency, robustness, access control, and scalability and is not required in most scenarios that were identified in section 2.1, we will not focus on VPNs that provide such mechanisms.

- **Entity Authentication:** Within a VPN the different cryptographic endpoints must be able to securely identify communication partners. For IP networks this implies that the announced addresses ranges of a corresponding VPN gateway must be uniquely assigned and linked to the identity of the gateway.

- **Data Integrity and Authentication:** All data packets transported within VPNs must be processed in a way that allows for a detection of non-intentional as well as intentional modification. Furthermore, for VPNs devices it is required to be able to securely differentiate between packets that originate from other devices of the same VPN and those that do not. In many environments this objective even requires that devices must be able to determine from which VPN device the packet originated.

- **Access Control:** Even though VPNs protect the transmitted data by encryption, access control services have to guarantee that all VPN devices have a valid permission to participate, as the data might be sent or received by inappropriate devices otherwise. There are two subtasks to fulfill:

  - **Static Access Control:** VPN devices must only create connections to other devices belonging to the VPN. Furthermore, it can be demanded that the number and identities of VPN participants can be restricted, e.g., by defining a trusted certificate authority (CA).
  - **Dynamic Access Control:** If pair-wise session keys are used, static access control includes the checking of the validity of the peer's certificate when a session is established. However, there should be periodic reauthentications to verify the validity of the connection. If group keys are used within the VPN, these keys must be changed whenever a VPN device joins (Backward Access Control) or leaves (Forward Access Control) the group in order to prevent further access.

- **Availability:** To our best knowledge there is no reliable data regarding denial-of-service (DoS) attacks on VPNs as operators do not publish figures for security reasons and the common backscatter analysis [25] cannot detect DoS attacks on VPNs. However, DoS attacks are expected to be become increasingly important as DoS attacks are very cheap to realize and yet effective [26]. Thus, an automatically configured VPN shall have the following properties:

  - **DoS Resistance:** It shall withstand DoS attacks at least to a certain extent. Different mechanisms exist to ensure this property, e.g., IKE cookies [27], client puzzles [28], and a distributed management.
  - **Graceful Degradation:** In the case of a partial DoS attack or compromise, other not directly affected VPN devices shall continue to work. This property is sometimes also referred to as survivability [29].
  - **DoS Recovery:** If the VPN is already under a DoS attack, it should be possible to migrate affected parts of the network to other IP addresses or domain names to quickly recover from the attack.

- **Minimal amount of security relevant configuration:** Automatic configuration systems for VPN shall minimize manual intervention and thereby reduce the possibility of human errors, which could possibly lead to vulnerabilities.

4. Configuration Approaches

The following discussion of actual auto-configuration mechanisms is divided into three parts: In the first section, several configuration algorithms with a central coordinator are presented. The second part covers decentralized systems, which do not have to have a SPoF, even though certain designated networks components exist. At last, mechanisms for a fully distributed configuration are presented.

4.1. Centralized Systems

Even though allowing for the construction of rather simple configuration mechanisms, all of the centralized mechanisms share a potential SPoF and bottleneck; thus, reducing robustness, scalability, and availability properties. The central coordinator can easily be identified by traffic analysis as all VPN components are connected to it, inhibiting an effective infrastructure hiding. Furthermore, it is impossible to recover a VPN in the event of a DoS attack as all clients must know the coordinator's IP address in advance.

4.1.1. Cisco Easy VPN

Cisco offers an IPsec-solution for remote access as well as simple hub-to-spoke scenarios under the Easy VPN or EzVPN branding [30]. It allows remote clients as well as remote sites to connect to a central VPN concentrator, which may also distribute IP addresses to the clients. Further convenience is achieved by allowing clients to authenticate via IKE XAUTH [31], e.g., with passwords obtained from a RADIUS server.

From a functional point of view, Easy VPN can handle private IP address ranges, gateway functionality, and works on unicast networks. Connections by clients via NAT are possible in Easy VPN, although the concentrator certainly must have a public address, and even though it makes IPsec configuration simpler, there is still room for improvement. A support for nested associations, QoS, and multicast within the VPN is not given.

Robustness and scalability properties are limited by the central VPN concentrator. The same goes for the reconfiguration speed, as every client must individually reconnect to the central entity.

Due to its topology Cisco Easy VPN is the only system that allows for a centralized inspection of traffic, even if it is redirected between two VPN clients. However, this leads directly to the major drawback of having a central coordinator, and several smaller security scratches. First, not even an optional end-to-end encryption is offered, so that the VPN concentrator is always able to intercept all data in plain text.

Second, the concentrator can also modify or inject new data, so that data integrity and authentication cannot be guaranteed from end-to-end. Third, the central entity can easily be discovered, which prevents infrastructure hiding, and a compromise of the concentrator leads to a full degradation of the VPN security. Furthermore, XAUTH has been found vulnerable to man in the middle (MITM) attacks [32] and even though the vulnerability is known for over five years already, there are still attack programs actively maintained for it [33]. Major reasons for the non-deployment of fixes are the lack of clarification, e.g., several pages on Ciscos website [34, 35] explain how to configure Easy VPN insecurely, and the more complicated deployment of the secure protocol. For administrators Cisco does not give the ability for an easy upgrade.

4.1.2. Cisco Group Encrypted Transport VPN

Another automatic configuration approach is brought to market by Cisco under the name Group Encrypted Transport (GET) VPN [36, 30]. Being a conglomerate of techniques and standards, such as Group Domain of Interpretation (GDOI) [37], GET targets to protect extremely large site-to-site VPNs.

From a simplified point of view, GET VPN (cmp. Figure 5) operates as follows:

1. Connecting VPN gateways contact a predefined key server, which will perform an authentication and authorization utilizing the IKE protocol. Alternatively if the key server is not available, one of up to seven predefined backup key servers may perform this task.

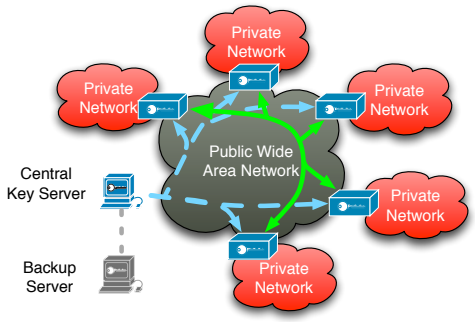

Figure 5: An example Cisco GET scenario

2. The key server or one of the backup servers will then send two symmetric group keys to the connecting VPN gateway. One is used to install a security association and one to protect further packets from the key server.
3. As all VPN gateways obtain the same keys they can exchange encrypted and integrity-protected data packets transparently.
4. In predefined intervals the key server distributes new data- and key-encryption keys to all VPN gateways, for reasons of scalability preferably utilizing multicast within the transport network.

A peculiarity of GET is the use of the so-called “tunnel-less encryption”, which uses a special IPsec operation mode, which is despite its name similar to the tunnel mode. However, in difference to tunnel mode the outer IP header is not generated by IPsec policies, but by simply carrying over large parts of the inner header, e.g., the IP addresses and the fields for QoS, to the unprotected outer part.

Result of this approach is a highly scalable system, perfectly designed to be transparent in current MPLS systems. This transparency allows for an integration of differentiated services and multicast, if the underlying transport network supports these IP extensions. However, it also prevents the use of NAT, nested associations, and private IP addresses, unless these are routed by the Internet service provider.

Despite the backup servers the approach must still be considered to be centralized, for the following reasons:

- Every VPN gateway connects to the same of key server, which has at most a constant number of replications.
- The backup key servers cannot be added or removed dynamically, as each server has to be configured in every VPN gateway.
- If there is a connection problem between the key servers, multiple key servers take over operation. This will eventually lead to deployment of inconsistent key material, and thus a network split.

All in all a failure of the central components may still lead to a major malfunction of the VPN.

The major drawback of the solution is the crippled security in comparison to manually configured VPNs [38]. Even though

some data confidentiality is achieved, internal attackers are able to decrypt any VPN traffic, in spite of the fact that they are neither sender nor receiver, due to the use of group keys. According to Cisco's specification PFS is not implemented, although GDOI actually specifies this. The topology of the infrastructure is easy to determine as all gateways only exchange IKE packets with the key servers, and due to copying the inner IP header to the outside, eavesdroppers can map traffic to client IP addresses. The copying also reduces covert-channel resistance significantly as IP headers are not forced to be encrypted. Thus, insiders behind a GET gateway can send arbitrary information via IP headers to external systems.

The integrity and authentication of all packets in the system does not only depend on all VPN gateways and key servers to be uncompromised, it also depends on the administrator who needs to explicitly activate a time-based replay protection mechanism. In comparison to the usual replay-window-based mechanism, this algorithm allows for a reinjection of packets limited only by the configured time frame. The result of such replay attacks primarily depends on the applications used within the VPN, but the re-injection of TCP ACK packets may cause a frequent retransmission of packets, for example. Thus, an attacker could use this to generate more data traffic within the VPN in order to support cryptographic attacks. Similar weaknesses are well known for wireless networks [39].

Furthermore, dynamic access control is not guaranteed as GET changes keys periodically and not if IPsec gateways are added or removed from the VPN. Thus, if the default rekey interval of 24 hours is chosen, a newly authenticated device may decrypt all VPN traffic of the last 12 hours on average. Even worse, gateways that are removed from the VPN could be able to decrypt or inject arbitrary future traffic as the rekeying is performed with symmetric keys, only, and the leaving device knows the key and may even still receive the multicast traffic.

The availability properties of GET are dominated by central key servers and every attacker knows that at most eight servers must be taken out in a GET VPN to prevent gateways from joining or installing new keys. Thus, if an attack is launched just before the expiration of the security associations, the whole VPN will be rendered unusable.

For security reasons VPNs usually change the traffic encryption key periodically. In GET, the central key server must initiate such a key change, even though it is not aware of the amount of traffic transmitted within the VPN. Thus, Cisco recommends [40] to estimate the amount of traffic sent on a regular basis, and schedule the rekeying message based on this. However, this estimation does not have to reflect the conditions under an attack, especially when considering reinjected packets, because of the faulty replay protection.

All in all, GET is a minimal invasive, privacy protecting mechanism to deploy IPsec in large infrastructures. Because of the crippled security properties, it is not a general replacement for manually configured VPNs.

4.1.3. Hamachi^2

Another system, which aims at VPN auto-configuration and has drawn much attention during the last years, is LogMeIn's Hamachi^2 [41]. The system uses a proprietary protocol, which is claimed to be designed according to principles given in IKE, to initiate site-to-site or remote access VPNs. However, site-to-site scenarios are currently not available for VPN gateways. Only the remote access appliance works with a single VPN gateway. The system is easy to install, and works well in NAT scenarios. An interesting property of Hamachi^2 is the use of the IP address space 5.0.0.0/8, which is currently unassigned by the Internet Assigned Numbers Authority (IANA). Every Hamachi^2 peer gets an internal, virtual IP address of this pool, in order to avoid collisions with common private IP address ranges like 10.0.0.0/8, if a client is behind a NAT router.

The company LogMeIn tries to make Hamachi^2 to appear as secure as possible, i.e., by stating that the cryptographic protocol is based on IKE and by load balancing over the Akamai network. Nonetheless, all configuration tasks are performed over the company's central services; so that all availability, confidentiality, and data access control questions depend on the company's ability to provide them. Furthermore, periodic rekeying properties are not mentioned in any of the documents.

4.2. Decentralized Systems

Central VPN configuration systems have been shown to offer easy administration and fulfill many functional objectives, but possible availability and scalability problems are undeniable. On the other side, distributed systems, usually without these problems, are often complex, rather insecure, or work only under special network conditions. An alternative are currently decentralized systems, with flexible distributed components and a coordinating infrastructure.

4.2.1. Key distribution via DNSSEC

A mechanism to automatically deploy certificates and keying material for VPN infrastructures is the use of the global Domain Name System (DNS) [42, 43], or preferably the more secure version DNSSEC [44, 45]. An opportunistic variant, where IPsec protection is only used if a DNS IPSECKEY record is available, is specified in [46]. The main idea behind all of these approaches is rather simple: whenever a secure connection needs to be established, a DNS lookup is performed and the authentication of the corresponding peer is performed on the returned data. Thus, the major advantage of the system is the ease of its deployment.

However, several problems arise already from a functional point of view, because VPN gateway-to-gateway functionality, nested security associations, private IP address ranges within trusted networks, and NAT compatibility are not addressed.

From a security perspective potential vulnerabilities of the authoritative DNS servers must be considered. This includes availability and robustness issues, as these servers form a logical SPoF, but also data authentication and integrity issues. Especially in standard infrastructures, DNS answers can sometimes be easily modified as recent discoveries have shown [47]. These data authentication and integrity issues were the major reason for the development of DNSSEC, but also here questions remain to be solved. Most prominent are the challenges to

find an acceptable organization to sign the root certificate and the single path of certification [48]. Furthermore, static as well as dynamic access control is problematic, because there is no restriction on who may obtain a valid certificate and a timely revocation of compromised keys is not addressed in the standard. Another issue may be the missing confidentiality of both DNS as well as DNSSEC, resulting insufficient infrastructure hiding as all IP addresses of VPN endpoints are disclosed in a public directory. This may imply, that attackers in possession of the DNS name of a single VPN device, are able to derive IP addresses of other VPN devices as well, at least if structured DNS names are used (e.g. `vpn01.myorganization.com`).

As noted in [49] even more issues may occur due to the combination of DNS and dynamic network environments. If mobile VPN devices are used, not only the DNS entries must be updated, but also the security policies, as a different policy may match in this case. The authors address this particular issue, with an Application Programmers Interface (API) for IPsec-aware programs.

4.2.2. Wippien

Wippien [50] is another system to establish decentralized VPN topologies. In contrast to the last approach the negotiation of connections is not performed via DNS servers, but over the Jabber infrastructure. To connect to one of the Jabber contacts, Wippien performs a simplistic key exchange via the Jabber servers. Later on, direct associations are created by a proprietary protocol, which is protected by Advanced Encryption Standard (AES).

In contrast to the DNS approach, Wippien is more flexible as Jabber allows for private address ranges and the software supports NAT hole punching. Nonetheless, gateway functionality is not available, security associations cannot be nested, and advanced properties like QoS-awareness or multicast support are not given. Despite being a decentralized system, the scalability is limited by the full meshed structure between friends and the robustness properties are bound by the Jabber servers, whose potential failures are critical to their clients.

With regards to security severe questions arise: we identified multiple severe security issues in less than one hour [51]. These included methods to perform man-in-the-middle attacks and the prediction of the used cryptographic keys. Mechanisms to attain PFS or periodic key changes are not foreseen. Integrity protection and authentication, if present, are not documented and not part of the open source components of Wippien.

4.2.3. Social VPN

In order to configure VPNs for private individuals the Social VPN system [52], utilizes a Facebook application to automatically exchange certificates with "friends" from the social network. These certificates are then used to configure a proprietary cryptographic key exchange, which has some similarities to IPsec. After this centralized processing the system works fully distributed, and a peer-to-peer network, based on Brunet [53], is used to discover peers and route encrypted IP packets. An interesting point is how address collisions are avoided: Social VPN assigns locally unique virtual IP addresses to each security association, which are then translated to the corresponding peer's addresses by NAT.

Social VPN is not able to configure gateways and can only create indirect security associations to circumvent untraversable NAT gateways. Apart from this, the functional and non-functional objectives are handled fairly well: it is simple to use, can handle private addressing and NAT, it scales over the number of participants, and can react somewhat robust against failures, even though the problem of network partitions stays unaddressed. It has also a very simple mechanism to distribute multicast packets by simply forwarding them to all connected peers.

However, severe problems arise when security is concerned. Given the risk of identity theft [54], Facebook does not seem to be an adequate choice to exchange certificates after all. Furthermore, by using this way of distributing certificates, the graceful degradation is not possible. It seems that the authors of Social VPN became aware of this problem as current releases do not use Facebook anymore, but depend on Jabber servers just like Wippien. The security of the underlying peer-to-peer network stays unaddressed by the authors, and even though the end-to-end-protection is preserved, attackers can perform routing attacks to discover arbitrary peers and to assault the availability of chosen participants.

4.2.4. N2N

Network to network (N2N) [55] is another proprietary system to create VPNs autonomously by setting up connections, which the authors claim to be peer-to-peer, but in contrast to fully distributed systems, such as P2PVPN, it depends on one or more dedicated "supernodes" to work. These supernodes are basically servers that transfer data between other participants, so called "edge nodes" (an example topology is given in Figure 6). In order to provide more efficiency, the edge nodes create shortcuts between each other, when a direct connection can be established. Thus, the supernode will only relay data if edge nodes cannot communicate directly within the transport network (e.g. due to NAT), have not established a direct connection yet, or broadcast messages need to be delivered. This construct makes the routing and forwarding process rather easy: packets are either delivered directly to another edge node or a supernode, which will then forward the packet to the destination. For reasons of robustness, edge nodes can also connect to multiple supernodes by a simple trick: as N2N is a layer 2 VPN, multiple edge node processes can be started and connected by a virtual Ethernet bridge. This method allows also for a coupling of multiple N2N VPNs.

Comparing N2N to the functional and non-functional objectives, reveals that it is easy to configure, allows a use of private addresses, and that at least edge nodes can located behind NAT router. Due to the full-meshed structure of edge nodes and the use of broadcasts, N2N does not scale well, and when multiple supernodes are available, there is no way for the edge nodes to determine the best relay, so that the efficiency may be rather bad.

Even the website of N2N claims [56] that several security issues exist in the current version of N2N: Packets are encrypted

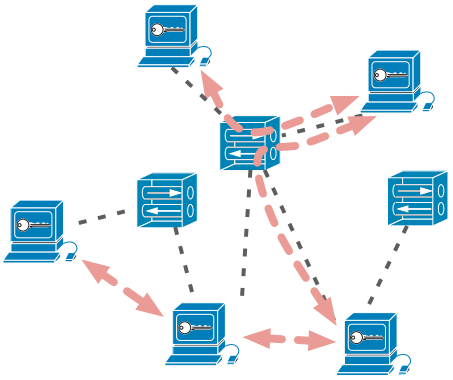

Figure 6: N2N topology with three supernodes and five edge nodes

by Twofish-CBC, but a message authentication code (MAC) for authentication and integrity protection is missing; the lack of nonces and sequence numbers allow for replay attacks, and used keys are not periodically changed to perform reasonable access control. The use of symmetric group keys inhibits entity authentication and a control of participating devices. Besides these documented issues, several others exist:

- The encryption algorithm does not seem to make use of initialization vectors (IVs), and thus eases traffic and cryptographic analysis.
- To secure the exchanged data all edge nodes of a VPN share a symmetric group key, making graceful degradation impossible.
- Overlay security is not addressed, so that spoofed addresses can cause severe problems.
- Supernodes do not authenticate edge nodes, and may be used by any client to relay data.
- As supernodes are peered over the Internet by the Spanning Tree Protocol, for scalability reasons only very few will exist, perhaps even forming a sparse topology. However, this will reduce the availability of the VPN.

Again, N2N is an example for difficulties resulting from the use of proprietary cryptographic protocols.

4.2.5. Dynamic Multipoint VPN (DMVPN)

DMVPN [57, 58, 2] is another combination of IP related protocols and best practices, which is merchandized by Cisco. When looking at DMVPN topologies, they seem to be very similar to the N2N topologies. Again a set of static nodes, so called "hubs", exist, but in contrast to N2N they are interconnected by IPsec tunnels, and perform an interior routing protocol between each others. Dynamic "spoke" nodes connect to hubs, which will then route data packets for them either to other spokes directly or via other hubs. In order to optimize this topology DMVPN introduces dynamic IPsec associations between communicating spokes that serve as shortcuts.

With regards to simple configuration, DMVPN reduces the amount of required configuration, but still substantially complex efforts are required – especially for scenarios with fault-tolerance, where hub interconnections need to be configured manually. Each hub and spoke may be a gateway for whole trusted network with its own private IP address range, and mostly for reasons of scalability hubs may be nested to create hierarchical structures. Spokes cannot be nested. DMVPN does not require any network features, other than unicast transport, and can create associations between NAT-spokes and public reachable hubs and spokes. Dynamic associations between spokes that are both behind NAT-gateways are not possible. The protection of multicast traffic in DMVPN topologies is simply resolved by routing these packets always over the responsible hub. Even though QoS is addressed, this is only done by possible queuing strategies, i.e., there are no considerations for reservation mechanisms. From a functional point of view, all objectives are addressed to at least some extend.

The non-functional objectives require a more differentiated discussion. Even though it is possible to use DMVPN to create redundant hub structures for the spokes, and thus create a robust VPN, each of the spokes must create a different virtual interface (including a unique IP address) for each of the hubs. Thus if the failure of 3 hubs must be tolerated, the configuration in each spoke grows due to Cisco's configuration format to nearly 400%. The scalability of DMVPN depends on several factors: the number of hubs that can be deployed, the amount of traffic that has to be forwarded over hubs, the mobility of spoke gateways, and the number of security associations that need to be created between spokes. A growing number of hubs may result for example in a higher latency until the routes of mobile gateways are correctly announced between hubs. In scenarios where each spoke transmits data to each other node, DMVPN will automatically create a fully meshed network, which may result in long IPsec setup times. Nonetheless, notably in comparison to other configuration approaches, DMVPN scales well for most scenarios, is rather efficient, and offers reconfiguration speeds that solely depend on routing protocol convergence.

The confidentiality of the transported data can be assured by unicast IPsec associations, which assures perfect-forward-secrecy, covert-channel resistance, data integrity, and authentication along the way. However, two confidentiality subgoals cannot be achieved completely: end-to-end protection is not given as hubs get in contact with plain-text data, and infrastructure hiding can be problematic as every spoke contacts its hub first and later on other spokes dynamically. Thus, for eavesdroppers it is easy to discover hubs, e.g., in preparation of further attacks, and to perform traffic analysis on spoke-to-spoke associations. Access control in DMVPN is performed by two measures: first a valid IPsec association must be created, and second a group authentication is performed via Next Hop Resolution Protocol (NHRP). Unfortunately, to perform this authentication symmetric group passwords are used, and neither a mechanism for forward access control, nor for an automatic periodic refresh of the passwords is given. Furthermore, the IP address ranges that are announced by spokes are only derived from a spoke configuration. This weakens DMVPN in compar-

ison to manually deployed IPsec VPNs, where hubs can verify the announced IP address ranges with their own configuration. Thus, the compromise of a spoke can be considered critical for access control and availability. Besides this problem of graceful degradation, it is the administrator's task in DMVPN to create a decentralized hub structure to be sufficiently protected against DoS attacks. A recovery during a DoS attack is not possible, because spokes only contact preconfigured hubs. In comparison to manual IPsec the required configuration is reduced.

4.3. Distributed Systems

VPN configuration systems without a central instance have often better scalability, robustness, and availability properties than their counterparts. Furthermore, in theory it is usually more difficult for an attacker to find good targets, because of better infrastructure hiding properties. However, these properties are usually traded in for fewer functional services or special requirements on the infrastructure.

4.3.1. Opportunistic Encryption

The most simple form of distributed VPN configuration is to setup each end system to use encryption if possible, i.e., to try to establish an SSL connection or IPsec association for each data packet first, and transmit the packet without protection if the setup failed. Better-Than-Nothing Security (BTNS) is a rather recently standardized approach that implements unauthenticated encryption for IPsec [59], but without the possibility of falling back to unprotected transmission.

Utilizing this method in each end system is rather simple, scalable, efficient, and resilient against DoS attacks. Nonetheless, the functionality is rather limited as is the security: Private address ranges, VPN gateways, nested security associations, NAT-to-NAT associations, confidentiality against active attackers, integrity protection and authentication, as well as access control are not possible with opportunistic encryption.

4.3.2. Cryptographically Generated Addresses

Similarly to the last presented approach, Cryptographically Generated Addresses (CGA) [60, 61, 62] do not aim at providing a security policy for security gateways. However, in contrast to BTNS, they increase the protection against active attackers by simplifying certification and making it applicable for each end-device to automatically configure and verify security policies. To use CGA each participant generates a self-signed public-key and registers a network address which corresponds to a cryptographic hash-value of the public key, e.g., in IPv6 networks the last 60 bits of the address are used to represent the fingerprint. In order to prevent attackers from generating certificates with the same fingerprint, client-puzzle-like mechanisms [28] are used to increase the computational complexity of certificate generation. Nonetheless, even with optimizations like in [63], the offered security against MITM attacks is efficiently limited by the number of bits that can be used for the fingerprint.

From a functional perspective CGA offers only very limited advances; only the objective "simple configuration" is fulfilled and only in IPv6 networks. In IPv4 networks the approach cannot be applied, because of the shorter addresses. Due to the simplistic design, non-functional objectives, such as efficiency and scalability, are handled very well. However, the offered security is rather low. In comparison to opportunistic encryption, a certain degree of entity authentication is achieved, but a DoS recovery becomes more complicated as a new certificate with a new crypto-puzzle has to be computed.

4.3.3. Cisco Tunnel Endpoint Discovery

The Tunnel Endpoint Discovery (TED) protocol [30, 57, 64] is part of another IPsec configuration system that is developed and marketed by Cisco. Even though some standardization efforts were taken [65], the only known implementation is proprietary.

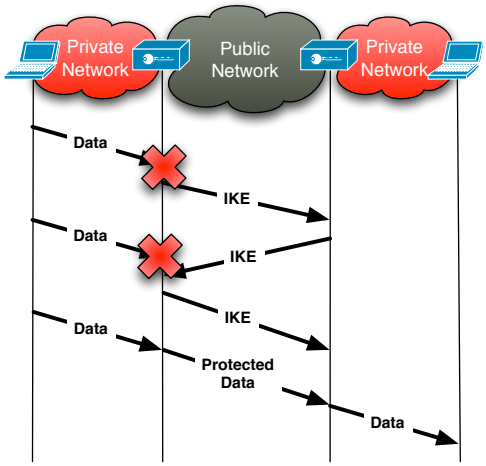

Figure 7: A typical establishment of IPsec associations in TED

The configuration of IPsec associations in TED works as follows (cmp. Figure 7):

1. Data packets without a valid security association are dropped.
2. However, in contrast to standard IPsec, an IKE packet is generated and sent to the destination address of the triggering data packet.
3. The IKE packet is intercepted by the corresponding IPsec gateway, and a response is sent.
4. After the establishment of a valid IPsec association subsequent retransmissions of the trigger packet will be handled like in standard IPsec.

The major drawback of this highly automated approach is the requirement for public IP addresses within the trusted networks as the routing of the IKE message must be performed by the transport network. Nested security associations, multicast, and NAT cannot be handled with TED, either. Furthermore, the configuration can be still a bit complex, and TED cannot react as robust to routing failures within the transport network as VPNs that provide an own routing functionality can.

The security properties are all-in-all handled well. The only drawback in comparison to statically assigned IPsec associa-

tions is the possibility of covert channels. That is, if the complete flexibility of TED is used, administrators cannot restrict the IP address range for which IKE packets are generated. Thus, client computers may communicate to arbitrary external hosts by triggering these packets at certain intervals. In case of DoS attacks, all client computers would require to obtain new IP addresses in order to recover from the attack, which is considered to be infeasible.

4.3.4. Security Policy Protocol

The Security Policy Protocol (SPP) [66, 67, 68] is an orphaned approach of the IETF, following principles similar to TED. Clients discover so called IPsec policy enforcement devices by a traceroute-like functionality, which in turn create an IPsec tunnel for client packets. Even though the approach offers more flexibility than TED as security policies are mediated and not assumed to be of the same administrative domain, its major properties in terms of functionality and security remain identical to TED.

4.3.5. Proactive Multicast IPSEC Discovery Protocol

An entirely different approach is suggested with Proactive Multicast IPSEC Discovery Protocol (PMIDP) [69]. Here, IPsec gateways regularly announce the private address ranges of their trusted networks by sending encrypted multicast packets within the untrusted network. The multicast routing functionality of this network ensures that the announcement is delivered to all gateways of the VPN, which in turn establish security associations on demand.

From a functional point of view, the system requires multicast support within the untrusted network, i.e., it is not possible to be used over the Internet. Whereas NAT and nested security associations are neither handled, PMIDP is one of the few approaches that can handle multicast within the VPN itself. The robustness properties of PMIDP are dominated by the rate at which announcements are sent, which influences both: the used network bandwidth and the rate at which networks can recover after a network failure. It must be kept in mind that each IPsec gateway only transmits one announcement at a time, but it receives the announcement of all others, possibly limiting the scalability of the approach, e.g., if some gateways are only reachable over low bandwidth wireless connections. Furthermore, the robustness of the approach could be improved by overlay routing mechanisms.

From a security point of view, several security challenges especially with regards to availability arise:

- Since all security gateways must be able to transmit data to the multicast group, the transport network must ensure that external attackers do not misuse the multicast forwarding for DoS attacks (e.g. by rate control). However, a method for securing the multicast transportation itself is not foreseen.

- The problem of replayed announcements is not dealt with, i.e., attackers could announce subnetworks even though they moved to a different position in the transport network.

- Announcements are supposed to be "encrypted" to support infrastructure hiding, even though the subject of distributing and renewing the required group keys stays undiscussed.

- The announced IP address ranges are not attested by a certificate. Thus, compromised gateways can announce arbitrary networks and attract their traffic.

The restriction to multicast-capable transport networks and the remaining security issues limit the field of applications of PMIDP significantly.

4.3.6. WASTE

Even though becoming famous for political rather than technical reasons [70], the so-called WASTE software was one of the first self-routing VPN systems that emerged for private individuals. Users of WASTE associate with friends using public-key negotiated connections, and within this partial mesh network they may transfer files or communicate via private chats [71]. In comparison to other mechanisms it does not allow for a forwarding of arbitrary traffic, but only specific services that are specified in the WASTE protocol. All of them are based on broadcast requests, which are flooded over all links of the VPN, and unicast replies that are routed by a backward learning algorithm.

WASTE is primarily for end-users, and thus does not support a VPN gateway mode or the protection of arbitrary data. Nonetheless, it is rather simple to use and does not require any special network support other than a public IP-address. Due to its routing algorithm, the scalability and efficiency of WASTE VPNs are rather limited.

The same goes for security: While confidentiality, authentication, access control, and integrity protection can protect to some extend against external attackers, internal attackers are not considered and the used Propagating Cipher-Block Chaining (PCBC) encryption mode allows for a reordering of cipher-text packets without a cryptographic avalanche effect [72]. Within the VPN itself no further security measures are taken, and every participant must rely on the integrity of all others: graceful degradation and end-to-end security are not given. In contrast to other approaches WASTE does offer some limited anonymity as VPN participants do not obtain the IP addresses of other devices than those that they are directly peered with. Peers that are under DoS attack may simply reconnect with a different IP address to recover from the attack.

4.3.7. P2PVPN

P2PVPN [73] is an open source project, which has not gained much notice from the scientific world. Nonetheless, it has some interesting properties: It uses its own protocol to create a partially meshed overlay network, in which locally generated packets are captured by a user process. These packets are sent over encrypted TCP connections to other hosts along routing paths that are determined by simple link-state routing. While this forwarding process is not very surprising, the bootstrapping offers an initial support for secure distributed operation. P2PVPN

connects to a BitTorrent server to find other peers, thus it can even work in scenarios, where all clients only have dynamic IP addresses. Even though P2PVPN does not support it, more sophisticated implementations of this approach can also work with all clients behind NAT gateways, and therefore make the deployment of VPNs for private individuals rather simple.

From a functional point of view P2PVPN offers many features: it is simple to configure, can handle nested security associations (if added manually) as well as private IP address ranges, and it works in generic IPv4 networks. Drawbacks are the missing gateway functionality and limited NAT capabilities. As the system creates unstructured overlays with as many connections as possible, P2PVPN is suited only for small VPNs for scalability and efficiency reasons.

The security properties of P2PVPN (as of version 0.7) are rather questionable: Even though an encryption between hosts is protected by AES in Cipher-block chaining (CBC) mode, there is no rekeying mechanism and none of the defined subgoals are fulfilled. In particular there is no end-to-end protection, no perfect forward secrecy, no covert channel resistance and infrastructure hiding possible. P2PVPN does not use any integrity protection and no authentication. It rather relies on the fact that an attacker cannot generate packets with valid UDP and TCP checksums after decryption, which cannot be considered to offer an acceptable level of security. In order to realize access control, P2PVPN distinguishes between two types of users: The first kind knows the private key associated to the VPN. This key is used to sign access tickets for the second type of users, those with temporary access capabilities. During this ticket generation the signer also generates the key pair of the invitee, thus, the signer can later on impersonate the invitee. Access control properties are furthermore weakened as it is not possible to remove users who know the private network key. Users with temporarily limited access are only removed by expiration of the ticket. Graceful degradation cannot be assured in any way.

4.3.8. tinc

The so-called *tinc* system [74] is similar to P2PVPN, and uses a proprietary protocol to create a full-mesh forwarding topology between manually connected VPN gateways. However, in contrast to P2PVPN it requires a substantial amount of configuration as every node's public key, internal IP address range, and public IP address must be configured in any other host that a connection shall be established to. Furthermore in contrast to P2PVPN, tinc can create connections between two systems that are both behind NAT gateways.

The manually created topology may limit the robustness of VPNs, even though the system itself can cope with network partitions and mergers, as well as partial routing problems in the transport network. The full-mesh forwarding structure also limits scalability and efficiency. From a security perspective end-to-end protection as well as PFS stay unaddressed, and even though rekeying mechanisms exist, re-authentications are not performed, nor is it possible to revoke authorizations easily. The resistance of tinc against DoS attacks primarily depends on manually deployed VPN topology, which would also be required to be updated in order to recover from DoS attacks. These configuration considerations, and the fact that tinc requires configuration changes, e.g. the MAC length, to reach the same security level as IPsec, make the security relevant configuration relatively large.

4.3.9. CloudVPN

The presented approach *CloudVPN* [75] creates unstructured overlay networks, similar to tinc, but without automatically creating a full-meshed forwarding structure. On top of the created SSL connections, nodes announce themselves by broadcast messages, and paths will be preferred according to their delay. Two peculiarities of CloudVPN are the possibility to use multipath routing with a weighting of paths according to their delays and the option to forward link layer data.

Despite being an auto-configuration approach CloudVPN requires still a substantial amount of settings, mostly because a static topology must be set up. Furthermore, there is no real gateway mode as every client needs to run a "gate"-program which forwards data via a TCP connection to the actual gateway. NAT can be traversed, but there are no mechanisms to create a direct connection between two NAT hosts. From a non-functional point of view, robustness is handled well as a VPN may be partitioned and merged back together, routing problems in the transport network may only be circumvented, if and only if in the manually created topology enough alternatives exist. The scalability and efficiency properties are dominated by the regular broadcasts. Again, CloudVPN does not offer end-to-end-protecting means, and does not authenticate the announced addresses, which becomes a problem when compromised VPN devices are considered. Periodic re-authentications are not provided, and a full DoS recovery is only possible when the topology is manually adapted.

4.4. Summary

The table in Figure 8 summarizes the presented approaches by matching them against the most relevant criteria. In particular the simplicity of configuration is rated in three grades (+,Ø,-), the number of supported VPN gateways (n,1,0), the ability to use private address ranges (+,-), the supported levels of nesting (n,1,0), multicast requirements (m,u) and grade of NAT traversal (+,Ø,-) are resumed. Whereas, in relation to NAT-traversal "-" means no traversal is possible, "Ø" means that devices behind NAT may connect to those that are not, and "+" stands for direct connections regardless of NAT gateways.

From the non-functional objectives robustness, scalability and efficiency are rated. Only agility was left out as it would have required measurements in a common user scenario, which currently does not exist.

The major block in the table concerns the security objectives: Again, the end-to-end security was rated in three steps, and reflects whether an intermediate system can gain access to the transmitted data (- = any system, Ø = compromised gateways, + = no system). The next column simply reflects the presence of PFS, followed by the convert-channel resistance for the approaches that have VPN gateway support. The infrastructure hiding column rates the difficulty to find exposed nodes,

			centralized			decentralized					distributed								
	Approach		Easy VPN	Group Encrypted Transport (GET)	Hamachi[2]	Key distribution via DNSSEC	Wippien	Social VPN	N2N	DMVPN	Opportunistic Encryption	Cryptographically Generated Addresses	Tunnel Endpoint Discovery	Security Policy Protocol	Proactive Multicast IPSEC Discovery Protocol	WASTE	P2PVPN	tinc	CloudVPN
General Properties	Protocol Layer of VPN		3	3	4	*3*	4	4	4	3	3	3	3	3	3	7	4	4	4
	Protocol Layer of Forwarded Data		3	3	3	3	3	3	2	3	3	3	3	3	3	4	3	2/3	2/3
Functional Objectives	Simple Config.		Ø	Ø	+	+	+	+	Ø	-	+	+	Ø	Ø	Ø	+	+	-	Ø
	Gateway Function		n	n	1	0	0	0	n	n	0	0	n	n	n	0	0	n	0
	Private Addresses		+	-	+	-	+		+	+	-	-	-	-	+	+	+	+	+
	Nesting		-	-	-	-	-	1	1	1/n	-	-	-	-	-	-	n	n	n
	Uni-/Multicast		u	u	u/m	u	u	u	u	u	u	u	u	u	m	u	u	u	u
	NAT Traversal		Ø	-	+	-	+	+	Ø	Ø	-	-	-	-	-	-	Ø	+	Ø
Non-functional Objectives	Robustness		-	-	-	-	-	+	+	Ø	Ø	Ø	Ø	Ø	Ø	Ø	+	Ø	Ø
	Scalability		-	+	Ø	+	Ø	+	-	Ø	+	+	+	+	Ø	-	-	-	-
	Efficiency		+	+	Ø	+	Ø	+	+	+	+	+	+	+	-	-	-	-	-
Security	Confidentiality	E2E-Protection	-	-	+	+	+	+	+	-	-	Ø	+	+	+	-	-	-	-
		PFS	+	-	+	+	-	+	-	+	+	+	+	+	+	-	-	-	+
		Covert-Channel Resistance	+	-	+	NA	NA	NA	+	+	NA	NA	Ø	Ø	+	NA	NA	+	+
		Infrastructure Hiding	-	-	-	-	-	+	-	Ø	NA	NA	Ø	Ø	-	+	Ø	+	+
	Entity Authentication		+	-	?	+	-	+	-	Ø	-	Ø	+	+	Ø	Ø	-	+	Ø
	Data Integrity/ Authentication		Ø	-	?	+	?	+	-	Ø	+	+	+	+	+	-	-	Ø	Ø
	Static Access Control		+	+	+	-	+	+	Ø	+	-	-	+	+	+	Ø	Ø	+	+
	Dynamic Access Control		+	-	-	-	-	+	-	Ø	-	-	+	+	-	+	-	-	-
	Availability	DoS-Resistance	-	-	Ø	-	-	Ø	-	Ø	+	+	+	+	-	Ø	Ø	Ø	Ø
		Graceful Degradation	-	-	-	+	+	+	-	-	+	+	+	+	-	-	-	Ø	-
		DoS-Recovery	-	-	-	-	-	-	-	-	+	Ø	-	-	-	+	+	-	Ø

Figure 8: Comparison between the different configuration approaches

i.e., central or bootstrapping servers, and can only be applied for the approaches where a special infrastructure is present, of course. For the rest of the security objectives grades are given for all approaches, again. That is entity authentication reflects, whether devices can be cryptographically distinguished, and IP addresses can be matched to the identity of a device without doubt. Static and dynamic access control are also scored, like defined in the objectives, i.e.: is it possible to allow access control by a group key (results in "Ø") or individual certificates ("+")? Last, availability properties are discussed, including the approaches' ability to withstand DoS attacks, to cope with internal attackers, and to recover from a DoS attack, e.g., by moving affected parts VPN to different IP address ranges. We refrained from adding another "security relevant configuration" column, as this would be redundant to "simple configuration" in this specific table.

5. Open Issues

As the last section revealed, several issues remain to be approached by developers as well as the research community. The following most important tasks were identified:

- **Secure bootstrapping:** When a device is first inserted into a VPN, it must somehow contact at least a single first other node, e.g., a hub, a server, an inviting participant. As VPNs are generally prone to DoS attacks, this first contact must be accomplished in a way that only authorized devices may find cryptographic endpoints. For example, simply choosing `vpn.intel.com` or `vpn.apple.com` is not considered well hidden. An interesting approach that might be worth a closer look is the possibility to embed bootstrapping information into public peer-to-peer networks, like in proposed by P2PVPN or [76]. This would allow to even bootstrap a VPN where all peers are behind

NAT routers, e.g., after a hub failure. However, especially in such an environment like a peer-to-peer network care must be taken to mask the presence of VPN devices. Thus, not only the identifiers used in the peer-to-peer network, must be changed periodically to avoid a traceability, but also as identifiers must rely on shared secrets a revocation mechanism must be designed.

- **Secure address management and allocation:** Even though some of the presented VPN configuration solutions allocate virtual addresses for single VPN devices, no solution can allocate private addresses for whole subnetworks. But even worse: nowadays address allocation is still a major security risk in auto-configured VPNs. As there is no solution which will embed the announced address ranges into certificates, compromised VPN devices may circumvent network access control, and thus graceful degradation cannot be guaranteed. A promising starting point might be the binary split algorithm [77], which has to be secured by certificate chains.

- **Dynamic topology control:** While many current VPN auto-configuration approaches construct fully routed overlay networks, their topology control is mostly rather simplistic. Usual systems still either create full-mesh or hub-to-spoke topologies, which represent two extremes: The first does not scale to hundreds of participants, because of the high number of security associations in each device, and the second is limited by hub capacities. DMVPN is a first solution with more flexibility, but still hub structures are not automatically determined. More flexible topology control algorithms could for example automatically detect feasible relay nodes, create backup paths, or construct security associations based on usage statistics or QoS parameters.

- **DoS resilience:** Current VPNs neither offer a way to migrate parts that are affected by DoS attacks, nor can they automatically protect the identity of weak nodes by relaying through more protected ones (similar techniques are suggested in [78, 79]). As there is no simple security service, which can guarantee availability, it is important to implement DoS resilience into all parts of the auto-configuration mechanism, e.g., do not allow externals to find the identities of vulnerable nodes by bootstrapping, create a robust topology, and use multi-path routing to find suitable backup paths. Regarding the robustness of a VPN topology, mathematical models should be developed to provide insights on the worst-case influence of optimal DoS attacks. Using such models it might then be possible to derive generic construction criteria for DoS resilient VPN.

- **Multicast Routing:** The scalable distribution of multicast traffic via generic overlay networks has been discussed widely in the research community [80]; the VPN overlay mechanisms do not offer such services, yet. Creating a secure and yet sufficiently easy to use multicast service, would give VPNs an additional value, e.g., for video conferencing or software distribution.

- **End-to-End protection in nested scenarios:** Current VPN configuration approaches cannot cope with multiple layers of encryption, like tunnel-in-tunnel scenarios in IPsec. Thus, in indirect communication scenarios intermediate nodes can eavesdrop on transported traffic fairly easily, which may become a problem when parts of the VPN are compromised. Auto-configuration approaches should be able to configure multiple encryption layers to protect data of different sensitivity levels.

- **Logging and monitoring:** Finally, when VPN become more flexible by dynamic topology control and routing, it becomes also more complex to detect and analyze problems. Thus, decentralized logging and monitoring facilities must be developed, allowing administrators for example to discover areas in their VPNs where packets are lost, where cryptographic handshakes fail, or where VPN devices have connectivity problems. Additionally to reverse path forwarding mechanisms for rather static scenarios, it might be possible to modify approaches developed for delay tolerant networks to ensure a feedback in highly mobile or tactical VPN.

All in all, there is still a lot of room for auto-configuration approaches to become more secure, flexible, and simple to maintain.

6. Conclusion

The different VPN deployment scenarios generate a large number of different functional objectives. The resulting network topologies may be complex, contain non-broadcast point-to-point connections and NAT gateways. Furthermore, the VPN should not restrict the use of multicast or QoS mechanisms. On the one hand large networks lead to high scalability requirements, while on the other hand nomadic or even mobile VPN devices require good network agility. Finally the heterogeneous security considerations make it difficult to judge whether a VPN is secure, and despite the number of existing VPN configuration approaches and over a decade of research, none of them seems satisfactory for all user groups.

Besides functional and non-functional deficits, many approaches offer a weaker security in comparison to manually configured VPN. Especially the large number of auto-configuration approaches that use own insecure cryptographic protocols is somewhat worrying. Two reasons for the deficits are probably the very different deployment scenarios of VPN users and the different objectives that developers and scientists had in mind, when designing their methods. This article summarized and categorized both: scenarios and objectives for a better understanding of relevant issues and design criteria of manual and automatic VPN configuration.

The break down of auto-configuration approaches showed three major groups: centralized, decentralized, and distributed,

approaches. Each of these groups offers with more or less secure systems, but with a clear trend towards scalable overlay networks in the last years. Still VPN auto-configuration must become more flexible and simple to become attractive. And while auto-configuration could make VPNs more secure by reducing the chances of human error, a lot of work needs to be done to really display this advantage.

References

[1] S. Raghunath, K. K. Ramakrishnan, S. Kalyanaraman, C. Chase, Measurement based characterization and provisioning of IP VPNs, in: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, 2004, pp. 342–355.

[2] S. Fluhrer, SYSTEM AND METHOD FOR PROTECTED SPOKE TO SPOKE COMMUNICATION USING AN UNPROTECTED COMPUTER NETWORK, United States Patent US 2007/0271451 A1 (2007).

[3] J. Reinhardt, BWI Informationstechnik GmbH, 2.400 Bundeswehr-Mitarbeiter kommen jetzt zur BWI, Statement to the Press (German) (2007).
URL http://www.bwi-it.de/presse_koeln_20-04-07.html

[4] A. Ghosh, R. Talpade, M. Elaoud, M. Bereschinsky, Securing ad-hoc networks using IPsec, in: Proceedings of the IEEE Military Communications Conference (MILCOM), 2005, pp. 2948–2953.

[5] L. A. DaSilva, S. F. Midkiff, J. S. Park, G. C. Hadjichristofi, N. J. Davis, K. S. Phanse, T. Lin, Network mobility and protocol interoperability in ad hoc networks, IEEE Communications Magazine 42 (11) (2004) 88–96.

[6] S. Hanigk, M. Kretzschmar, F. Eyermann, A Distributed Routing Architecture for Secure Communication over Highly Dynamic Radio Networks, in: Communication Systems and Networks and Workshops, 2009.

[7] VPN (The Easy Way) V24+ (2009).
URL http://www.dd-wrt.com/wiki/index.php/VPN_%28the_easy_way\%29_v24\%2B

[8] D. Kravets, Wired.com, Pirate Bay Launches VPN Service (2009).
URL http://www.wired.com/threatlevel/2009/06/ipredator

[9] Relakks – Trygghetsbolaget i Lund AB, Relakks – Questions and Answers (2009).
URL https://www.relakks.com/faq/qna/

[10] T. Dierks, E. Rescorla, The Transport Layer Security (TLS) Protocol, IETF Request for Comments 5246 (Proposed standard) (2008).
URL http://www.ietf.org/rfc/rfc5246.txt

[11] M. Bauer, Paranoid penguin: Rehabilitating Clear-Text Network Applications with Stunnel, Linux Journal 2004.

[12] E. Rescorla, N. Modadugu, The Design and Implementation of Datagram TLS, in: ISOC Network and Distributed System Security (NDSS) Symposium, 2004.

[13] E. Rescorla, N. Modadugu, Datagram Transport Layer Security, IETF Request for Comments 4347 (Proposed standard) (2006).
URL http://www.ietf.org/rfc/rfc4347.txt

[14] S. Kent, K. Seo, Security Architecture for the Internet Protocol, IETF Request for Comments 4301 (Proposed standard) (2005).
URL http://www.ietf.org/rfc/rfc4301.txt

[15] S. Frankel, Demystifying the IPsec Puzzle, Artech House Publishers, 2001.

[16] N. Ferguson, B. Schneier, A Cryptographic Evaluation of IPsec, Tech. rep., Counterpane Internet Security, Inc. (2000).
URL http://www.schneier.com/paper-ipsec.pdf

[17] C. Kaufman, Internet Key Exchange (IKEv2) Protocol, IETF Request for Comments 4306 (Proposed standard) (2005).
URL http://www.ietf.org/rfc/rfc4306.txt

[18] E. Rescorla, M. Ray, S. Dispensa, N. Oskov, Transport Layer Security (TLS) Renegotiation Indication Extension, Internet-Draft (2010).
URL http://tools.ietf.org/id/draft-ietf-tls-renegotiation-03.txt

[19] E. Levy, The Crumbling Tunnel, Phrack 53.
URL http://www.phrack.com/issues.html?issue=53&id=12&mode=txt

[20] W. M. Townsley, A. J. Valencia, A. Rubens, G. S. Pall, G. Zorn, B. Palter, Layer Two Tunneling Protocol "'L2TP"', IETF Request for Comments 2661 (Proposed standard) (1999).
URL http://www.ietf.org/rfc/2661.txt

[21] gematik – Gesellschaft für Telematikanwendungen in der Gesundheitskarte mbH, Konnektorspezifikation Version 3.0.0 (in German) (September 2009).
URL http://www.gematik.de/cms/media/dokumente/release_4_0_0/DezentraleKomponenten.zip

[22] G. Schaefer, Security in Fixed and Wireless Networks: An Introduction to securing data communications, John Wiley & Sons, 2004.

[23] D. Andersen, H. Balakrishnan, F. Kaashoek, R. Morris, Resilient Overlay Networks, in: SOSP '01: Proceedings of the eighteenth ACM symposium on Operating systems principles, 2001, pp. 131–145.

[24] W. Diffie, M. E. Hellman, New Directions in Cryptography, IEEE Transactions on Information Theory 22 (6) (1976) 644–654.

[25] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, S. Savage, Inferring Internet Denial-of-Service Activity, ACM Transactions on Computer Systems 24 (2) (2006) 115–139.

[26] M. Lesk, The New Front Line: Estonia under Cyberassault, IEEE Security & Privacy 5 (4) (2007) 76–79.

[27] R. Perlman, C. Kaufman, Key Exchange in IPSec: Analysis of IKE, IEEE Internet Computing 4 (6) (2000) 50–56.

[28] T. Aura, P. Nikander, J. Leiwo, DOS-Resistant Authentication with Client Puzzles, in: Security Protocols, Springer Berlin / Heidelberg, 2001, pp. 170–177.

[29] S. Patton, B. Smith, D. Doss, W. Yurcik, A Layered Framework Strategy for Deploying High Assurance VPNs, in: fifth IEEE International Symposim on High Assurance Systems Engineering, 2000, pp. 199–202.

[30] Y. Bhaiji, Network Security Technologies and Solutions, 1st Edition, Cisco Press, 2008, Ch. Part III: Data Privacy.

[31] S. Beaulieu, R. Pereira, Extended Authentication within IKE (XAUTH), Expired Internet-Draft (2001).
URL http://tools.ietf.org/html/draft-ietf-ipsec-isakmp-xauth-06

[32] Cisco Systems Inc., Cisco IPsec VPN Implementation Group Password Usage Vulnerability, Cisco Security Notice (2006).
URL http://www.cisco.com/warp/public/707/cisco-sn-20040415-grppass.pdf

[33] D. Roethlisberger, FakeIKEd (2010).
URL http://www.roe.ch/FakeIKEd

[34] Cisco Systems Inc., Cisco Easy VPN Client for the Cisco 1700 Series Routers, White Paper (March 2010).
URL http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/1700/1700cnts/ezvpn.pdf

[35] Cisco Systems Inc., Scenario: IPsec Remote-Access VPN Configuration, White Paper (March 2010).
URL http://www.cisco.com/en/US/docs/security/asa/asa80/getting_started/asa5505/quick/guide/rem_acc.pdf

[36] Cisco Systems Inc., Cisco Group Encrypted Transport VPN, Cisco Feature Guide (2007).
URL http://www.cisco.com/en/US/docs/ios/12_4t/12_4t11/htgetvpn.pdf

[37] M. Baugher, T. Hardjono, H. Harney, B. Weis, The Group Domain of Interpretation, IETF Request for Comments 3547 (Proposed standard) (2003).
URL http://www.ietf.org/rfc/rfc3547.txt

[38] M. Rossberg, G. Schaefer, Ciscos Group Encrypted Transport VPN – A sceptical analysis, in: Proceedings of D-A-CH security, German, 2009, pp. 351–360.

[39] E. Tews, R.-P. Weinmann, A. Pyshkin, Breaking 104 Bit WEP in Less Than 60 Seconds, in: Information Security Applications, Springer Berlin / Heidelberg, 2008, pp. 188–202.

[40] Cisco Systems Inc., Group Encrypted Transport VPN Security Analysis, White Paper (2008).
URL http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6635/ps7180/white_paper_c11-471053.pdf

[41] LogMeIn, LogMeIn Hamachi^2 Security: An Overview (2009).
URL https://secure.logmein.com/documentation/hamachi/Hamachi_Security_White_Paper.pdf

[42] M. C. Richardson, A Method for Storing IPsec Keying Material in DNS, IETF Request for Comments 4025 (Proposed standard) (2005).

URL http://www.ietf.org/rfc/rfc4025.txt
[43] S. Josefsson, Storing Certificates in the Domain Name System (DNS), IETF Request for Comments 4398 (Proposed standard) (2006).
URL http://www.ietf.org/rfc/rfc4398.txt
[44] V. Le, H. Guyennet, IPSec and DNSSEC to Support GRID Application Security, in: IEEE International Symposium on Cluster Computing and the Grid, 2002, pp. 458–459.
[45] P. J. M. Merino, A. García-Martínez, M. M. Organero, C. D. Kloos, Enabling Practical IPsec Authentication for the Internet, in: On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, 2006, pp. 392–403.
[46] M. C. Richardson, D. H. Redelmeier, Opportunistic Encryption using the Internet Key Exchange (IKE), IETF Request for Comments 4322 (Proposed standard) (2005).
URL http://www.ietf.org/rfc/rfc4322.txt
[47] D. Kaminsky, Catching up with Kaminsky, Network Security 2008 (9) (2008) 4–7.
[48] M. Burmester, Y. Desmedt, Is hierarchical public-key certification the next target for hackers?, Communications of the ACM 47 (8) (2004) 68–74.
[49] J. Trostle, B. Gossman, Techniques for improving the security and manageability of IPsec policy, International Journal of Information Security 4 (3) (2005) 209–226.
[50] K. Petric, Wippien - Open source p2p VPN software (2010).
URL http://www.wippien.com/
[51] M. Rossberg, Multiple Security Issues in Wippien, Tech. rep., Ilmenau University of Technology (2010).
URL http://wcms1.rz.tu-ilmenau.de/fakia/fileadmin/template/startIA/telematik/Mitarbeiter/rossberg/wippien-advisory.txt
[52] R. Figueiredo, P. O. Boykin, P. S. Juste, D. Wolinsky, Social VPNs: Integrating Overlay and Social Networks for Seamless P2P Networking, in: 17th IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE/COPS), 2008, pp. 93–98.
[53] A. Ganguly, A. Agrawal, P. O. Boykin, R. Figueiredo, IP over P2P: enabling self-configuring virtual IP networks for grid computing, in: 20th International Parallel and Distributed Processing Symposium (IPDPS), 2006.
[54] L. Bilge, T. Strufe, D. Balzarotti, E. Kirda, All Your Contacts Are Belong to Us: Automated Identity Theft Attacks on Social Networks, in: Proceedings of the 18th International World Wide Web Conference, 2009, pp. 551–560.
[55] L. Deri, R. Andrews, N2N: A Layer Two Peer-to-Peer VPN, in: Resilient Networks and Services, Springer Berlin / Heidelberg, 2008, pp. 53–64.
[56] L. Deri, R. Andrews, n2n: a Layer Two Peer-to-Peer VPN (2010).
URL http://www.ntop.org/n2n/
[57] V. Bollapragada, M. Khalid, S. Wainner, IPSec VPN Design, Cisco Press, 2005.
[58] Cisco Systems, Inc., Dynamic Multipoint VPN (DMVPN) (2006).
URL http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t13/ftgreips.pdf
[59] N. Williams, M. C. Richardson, Better-Than-Nothing Security: An Unauthenticated Mode of IPsec, IETF Request for Comments 5386 (Proposed standard) (2008).
URL http://www.ietf.org/rfc/rfc5386.txt
[60] J. Laganier, G. Montenegro, A. Kukec, Using IKE with IPv6 Cryptographically Generated Addresses, Expired Internet-Draft (2007).
URL http://tools.ietf.org/html/draft-laganier-ike-ipv6-cga-02
[61] T. Aura, Cryptographically Generated Addresses (CGA), IETF Request for Comments 3972 (Proposed standard) (2005).
URL http://www.ietf.org/rfc/rfc3972.txt
[62] T. Aura, Cryptographically Generated Addresses (CGA), in: Information Security, Springer Berlin / Heidelberg, 2003, pp. 29–43.
[63] J. W. Bos, O. Özen, J.-P. Hubaux, Analysis and Optimization of Cryptographically Generated Addresses, in: Information Security, Springer Berlin / Heidelberg, 2009, pp. 17–32.
[64] S. Fluhrer, DETERMINING SECURE ENDPOINTS OF TUNNELS IN A NETWORK THAT USES INTERNET SECURITY PROTOCOL, United States Patent US 2007/7207063 B1 (2007).
[65] S. Fluhrer, Tunnel Endpoint Discovery, Expired Internet-Draft (2000).
URL http://tools.ietf.org/html/draft-fluhrer-ted-00
[66] P. Srisuresh, L. A. Sanchez, Policy Framework for IP Security, Expired Internet-Draft (1999).
URL http://tools.ietf.org/html/draft-ietf-ipsec-policy-framework-00
[67] M. Baltatu, A. Lioy, D. Lombardo, D. Mazzocchi, Towards a policy system for IPsec: issues and an experimental implementation, in: Proceedings of 9th IEEE International Conference on Networks (ICON), 2001, pp. 146–151.
[68] L. A. Sanchez, M. N. Condell, Security Policy Protocol, Expired Internet-Draft (2002).
URL http://tools.ietf.org/html/draft-ietf-ipsp-spp-01
[69] T. Tran, Proactive Multicast-Based IPSEC Discovery Protocol and Multicast Extension, in: Proceedings of the IEEE Military Communications Conference (MILCOM), 2006.
[70] J. Hu, AOL pulls Nullsoft file-sharing software, CNET News (2003).
URL http://news.cnet.com/2100-1032_3-1011585.html
[71] M. Ek, F. Hultin, J. Lindblom, WASTE Peer-to-Peer Protocol, Tech. rep., Luleå tekniska universitet (2005).
URL http://sourceforge.net/projects/j-waste/files/WASTE\%20Documentation/WASTE\%20Documentation\%201.1/waste_documentation-1.1.pdf/download
[72] C. J. Mitchell, Cryptanalysis of Two Variants of PCBC Mode When Used for Message Integrity, in: Information Security and Privacy, Springer Berlin / Heidelberg, 2003, pp. 560–571.
[73] W. Ginolas, P2PVPN – Documentation (2010).
URL http://p2pvpn.org/documentation.html
[74] G. Sliepen, The difficulties of a peer-to-peer VPN on the hostile Internet, Slides of Talk at FOSDEM (2010).
URL http://www.tinc-vpn.org/presentations/fosdem-2010/tinc_fosdem2010_slides.pdf
[75] M. Kratochvil, CloudVPN how it works (2009).
URL http://e-x-a.org/stuff/cloudvpn-poster.jpg
[76] J. Dinger, O. P. Waldhorst, Decentralized Bootstrapping of P2P Systems: A Practical View, in: Proceedings of NETWORKING 2009, 2009, pp. 703–715.
[77] A. Misra, S. Das, A. Mcauley, S. K. Das, Autoconfiguration, Registration and Mobility Management for Pervasive Computing, IEEE Personal Communications Systems Magazine (8) (2001) 24–31.
[78] A. D. Keromytis, V. Misra, D. Rubenstein, SOS: An Architecture for Mitigating DDoS Attacks, IEEE Journal on Selected Areas in Communications 22 (2004) 176–188.
[79] M. Brinkmeier, M. Rossberg, G. Schaefer, Towards a Denial-of-Service Resilient Design of Complex IPsec Overlays, in: Proceedings of International Conference on Communications (ICC), 2009.
[80] Y. Liu, Y. Guo, C. Liang, A survey on peer-to-peer video streaming systems, Peer-to-Peer Networking and Applications 1 (2008) 18–28.