Automatic configuration of complex IPsec-VPNs and implications to higher layer network management

Michael Rossberg[1] · Guenter Schaefer[1] · Kai Martius[2]

[1] Telematics/Computer Networks Research Group
Ilmenau University of Technology
{michael.rossberg | guenter.schaefer}@tu-ilmenau.de

[2] secunet Security Networks AG
kai.martius@secunet.com

Abstract

As the Internet emerges to be, not only the most important, but in many areas the only way of efficient communication, it becomes also vital for business and government institutions to securely exchange data via this medium. This led to the development of virtual private networks (VPNs). However, security in this aspect does not only refer to confidentiality, integrity, authentication, and access control, but also availability; a subgoal of increasing importance due to cheap and simple execution of denial-of-service (DoS) attacks.

In order to increase the DoS-resilience of VPNs, the topology of this overlay network must react flexible to circumvent affected network parts and to reintegrate systems, which become available after the DoS attack ended or have been moved to different address ranges. Therefore, we developed a fully distributed IPsec configuration mechanism, which is able to react to failures dynamically and is yet scalable, efficient, and secure.

Nonetheless, the usually required higher layer services do not work in a distributed way. Thus, a failure may still cause availability issues as services like Domain Name System (DNS) may become inaccessible, even though a network connection is still present.

This article introduces distributed VPN auto-configuration and goes into detail on distributed network services.

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, more flexible, and 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 and affordable communication and exchange of files.

Both scenarios can be supported by the creation of virtual private networks (VPNs) on top of the IP layer, e.g. by making use of the IPsec protocol suite. Every participant in such a VPN is given a certificate or password that enables him to securely communicate with others by pre-

senting a compatible certificate or the same password. Security in the sense of VPN primarily concerns confidential data transmission, but often also integrity protection and authentication.

Even though security is naturally handled very well by IPsec VPN, 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 IPsec 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 by the power of two 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. Thus, Secure OverLay for IPsec Discovery (SOLID) [RoSS10], was developed, which – in difference to other IPsec configuration mechanisms – does not rely on dedicated servers or hubs and simply uses the public Internet infrastructure.

SOLID is able to automatically configure complex IPsec VPNs, even in scenarios that require the configuration of nested networks and mobile IPsec gateways. For this purpose, it only requires valid certificates to autonomously establish VPNs, thus causing a bare minimum of manual intervention. It is inspired by established peer-to-peer principles and it structures the overall configuration problem into five subtasks: The bootstrapping of joining or restarting IPsec gateways, assignment of address ranges to these gateways, control and optimization of the VPN topology, discovery of private address ranges, and routing in the overlay. SOLID creates topologies that are very resilient towards single or correlated failures of IPsec gateways, and even towards denial-of-service (DoS) attacks.

The full distribution of all configuration tasks is the key to automatically achieving many of the fulfilled objectives, such as scalability, robustness and DoS-resistance, as the planning of central instances requires careful manual planning. Thus, in order to fully exploit SOLIDs capabilities, not only the configuration of the VPN protection mechanisms themselves must be distributed. It is also required to distribute supplementary higher layer services, such as the Domain Name Service (DNS), time synchronization and logging, so that they also work when parts of the VPN are unavailable due to mobility reasons or DoS attacks. By embedding structured and unstructured communication paradigms into the IPsec overlay itself, SOLID can also transparently provide some of these higher layer services.

The next section covers a brief overview on the objectives of VPN auto-configuration, followed by section 3 with a discussion of related work in reference to science and commercial development. Afterwards, a round-up view for an own approach for a self-configuring VPN – SOLID – and its availability properties are discussed. The fifth section goes into detail on three network services that are implemented within SOLID-VPN in a distributed way. Addi-

tionally, open issues regarding other network services are also covered. Finally, the article closes with a conclusion.

2 Objectives

From an end-user perspective a VPN should have the following properties:

- **Simplicity:** Users of a system do not wish to configure the VPN itself or a complex configuration mechanism manually, but want it to automatically work and adapt to its current environment.
- **Versatile network environment:** The VPN services are for example expected to work in global, unicast-only networks, shall configure VPN gateways and single nodes, handle internal private address ranges and cope with network address translation (NAT).
- **Transparency:** As existing protocols and applications are unlikely or at least expensive to be adapted for VPN awareness, configurations systems as well as network services must emulate accepted interfaces.
- **Scalability:** Large VPNs may consist of many hundred or even thousands of participants. A fact that does not only have to be considered for automatic configuration, but also by distributed services.
- **Security:** Like indicated in the introduction, the major functional goal of VPNs is the ensuring of data confidentiality, data integrity and authentication, as well as access control. All of these properties can be achieved by mandatory data protection, e.g. by IPsec or TLS. In comparison to manually deployed VPNs, it must be ensured that the configuration mechanism does not weaken the security. Even more difficult is the realization of availability, which can be dissected into the subgoals of:
    - **DoS resistance:** The ability to withstand sabotage by external as well as internal attackers requires VPNs to organize themselves and its services a fully distributed manner.
    - **DoS recovery:** Fractions of a VPN suffering from DoS attacks shall be able to be relocated to different addresses in the transport network, and seamlessly reintegrate into the VPN.
    - **Graceful Degradation:** Even if some components are compromised, the overall security of the rest of the VPN shall stay unaffected.

All in all, this article focuses on availability aspects of automatically deployed VPNs and the defense against internal attackers.

3 Related Work

Current topologies of VPNs can mostly be broken down into fully meshed site-to-site VPN on the one hand and “Hub-to-Spoke” architectures on the other hand (see Fig. 1). The major drawback of site-to-site VPNs is the limited scalability as $O(n^2)$ security associations must be configured and maintained. Due to this property the dynamic reintegration of mobile nodes or a fast DoS recovery is considered to be infeasible.

Hence, the more common topology implies the use of one or more static hubs and dynamic spoke nodes [RRKC04, Flur07, Bhaj08], which is also easier to configure automatically. However, the central coordinator is also a potential weakness in terms of scalability and avail-

ability. Furthermore, it is not possible to integrate VPN nodes without a direct connection to the hub, which is essential for attack resilient topologies. Graceful degradation is an additional problem, because hubs are able to decrypt all traffic that is passed through them.

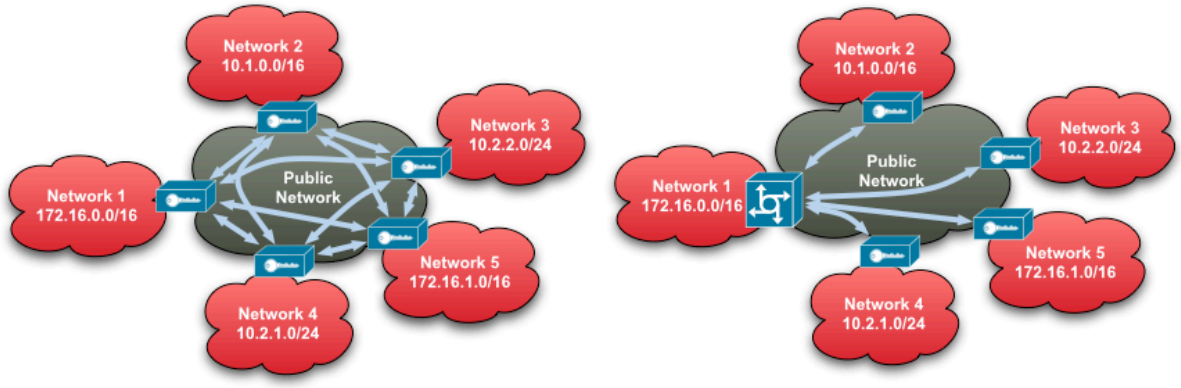

Fig. 1: Example topologies of a site-to-site VPN (left) and a hub-to-spoke VPN (right)

Other VPN topologies that can be configured automatically [Tran05, Aura05, Flur07, Bhaj08] depend on special services of the underlying transport network, e.g. public routable IP addresses within the VPN or globally available multicast. Thus, they only work in certain scenarios and are not suitable as a general replacement for manual VPN configuration. Furthermore, a widespread system showed severe security deficits [RoSc09].

None of the known systems and concepts is optimized on providing availability [RoSc10].

4 Secure OverLay for IPsec Discovery (SOLID)

In contrast to these sketched approaches our presented approach SOLID, creates a self-organizing overlay network, which includes functionality for the discovery of VPN gateways, routing and topology control. In order to construct a VPN overlay, SOLID creates initially only two security associations per gateway proactively, so that an ordered ring structure emerges. As all gateways are ordered by the internal IP address ranges of their private networks, the responsible destination gateway for each data packet can be determined by simply searching along the ring structure.

This search algorithm requires $O(n)$ overlay hops on average, but can be reduced to $O(\log n)$ by introducing cross-connections through the ring. This structure is similar to Chord [SMK+01] or I3 [SAS+02]. However, IPsec gateways cannot be ordered in the ring by random or hashed identifiers, as this would not allow for a variable subnet match. Thus, SOLID cannot rely on the uniform distribution of the looked up keys. Instead random samples are taken estimate the real distribution of inner IP addresses and later used to ensure a good placement of cross-connections over the address space.

Another problem of systems like Chord is their non-applicability to nested security gateways or transport networks in which not all participants can communicate directly with each others, e.g., due to security constraints, mobility, or potentially ongoing routing attacks. Hence, SOLID's topology control creates security associations between affected systems through the VPN itself. The required data exchange is routed through the paths, looked up during the discovery step and thus the ring structure is actually folded into the topology of the transport network like illustrated in Fig. 2.

In order to perform an actual forwarding of user data, a routing mechanism must be deployed to find paths that are as short as possible. However, security associations may change quickly,

and the usual routing algorithms depend on a distribution of link knowledge within the whole network, which is slow and must be performed proactively. Thus, in SOLID the forwarding of the actual data packets is initially performed over the security associations used during the discovery, which may be relatively long at first. If needed these paths are later on reactively optimized and within the typical topologies of substrate networks this mechanism leads to ways of ideal length with regards to the hop count. An important property is SOLID's guarantee for end-to-end security as routed data packets are delivered inside a nested IPsec association.

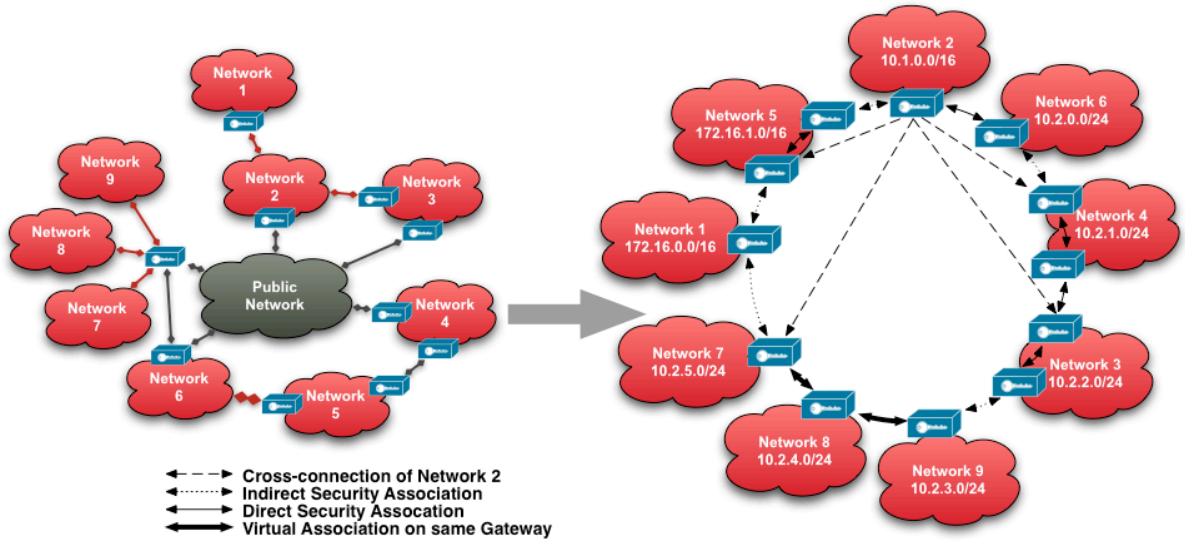

Fig. 2: Example of a transport network and the resulting overlay structure

5 Network Services

Its distributed structure and the possibility to create indirect security associations enable SOLID to react rather robust against DoS attacks as for example node failures affect only the directly hit parts of the VPN and selective link failures can be bypassed. However, problems arise when considering the support of services like DNS, NTP, or the distribution of certificate revocation lists as the devices in a VPN still depend on the availability of these decentralized or even central services. Thus, in order to take full advantage of a distributed and flexible VPN, these services also need to be implemented in a distributed way, while still considering the objective from section 3.

We started tackling the challenge of developing distributed services by implementing a set of entirely different services into SOLID's overlay network. In particular these were a time synchronization service, a name resolution system and a rudimentary network monitoring system. All three distributed systems have very different communication schemes: to perform time synchronization all systems must agree on a single time and frequency, name resolution requires the realization of a distributed database, and the monitoring facilities require a reverse multicast mechanism.

5.1 Time Synchronization

In order to provide distributed time synchronization services between VPN devices, SOLID uses a diffusion model in which each node constantly measures delays to its neighbors and exchanges timestamps [Gole10]. The required information is piggybacked in the dead peer detection mechanism, so that no additional message overhead occurs. If a VPN device

measures an offset to its neighbors, it will automatically adjust the value of its own clock and the corresponding frequency correction by a fraction of that offset.

This rather simple mechanism ensures a convergence of time and frequency of all devices within a VPN to a common arbitrary value, depending on initialization values and user communication patterns. Thus, the mechanism performs only an internal synchronization and depends on a different scheme to also allow for an external synchronization. As illustrated in Fig. 3, this is achieved by synchronizing a few VPN devices with external sources, such as GPS, or better by an authenticated source via modem or terrestrial signals.

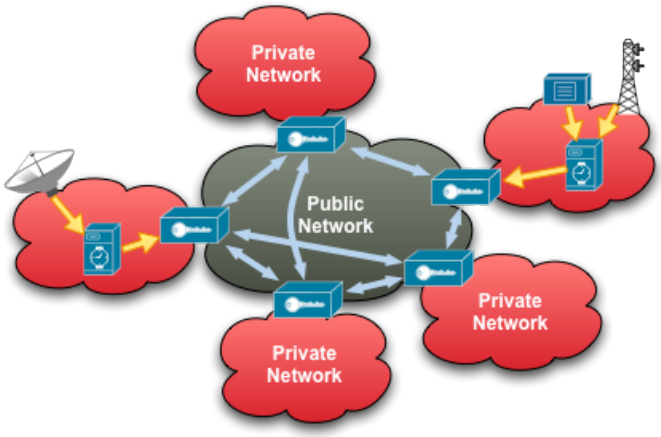

Fig. 3: Illustration of time synchronization infrastructure

In order to protect against external and internal attackers, all exchanged packets are protected by an end-to-end IPsec encapsulation. Furthermore, statistical tests ensure that associations with a high delay or jitter are not used for time synchronization, so that the convergence of the process cannot be threatened by external attackers.

The protection mechanisms against internal attackers are more complex. First of all, statistics are performed on the reasonability of the exchanged time data, i.e., other nodes must show a consistent behavior over time and if more than half of the neighbors are in a stable state, strongly deviating values are not taken into account. Hence, the influence of a potential internal attacker can be strongly limited. Second, a node only synchronizes with nodes that are marked for proactive creation by the topology control algorithm, ensuring that attackers cannot widen their influence by connecting to more nodes. The influence of any potential attacker is thus bound to a logarithmic number of peers.

5.2 DNS Name Resolution

A second, perhaps even more important mechanism copes with the problem of name resolution. If SOLID is configured to perform this task, every VPN client or VPN gateway will get one or more name ranges it is responsible for, e.g., `*.accounting.vpn` [Schu10]. These ranges must be attested by a certificate authority to prevent internal attacks. Clients within the private networks may then either be statically appointed with a name from a set or register one dynamically utilizing the Dynamic Host Configuration Protocol (DHCP).

All VPN devices will aggregate the names they are responsible for, sign sets of DNS records, and publish them in a distributed hash table (DHT), which is embedded within the VPN overlay. Replication mechanisms and on demand re-registrations ensure that the required information is available, when considering node and link failures.

Using the name service is transparent to end-systems: every VPN gateway is registered via DHCP to be a local DNS server, and queries are automatically handed to the SOLID daemon, which will either reply with a cached value, or by querying the DHT and verifying the answers.

5.3 VPN Monitoring

A further problem of distributed topologies is the more complicated monitoring and problem solution. In the context of VPN this includes, for example, the surveillance of the processing and network load of the VPN devices as well as the stability of security associations. Instead of creating a centralized monitoring facility, SOLID can report its status to one or more probes by utilizing an independent reverse multicast tree to each one. Administrators may then connect to these probes and obtain a live overview over the whole VPN.

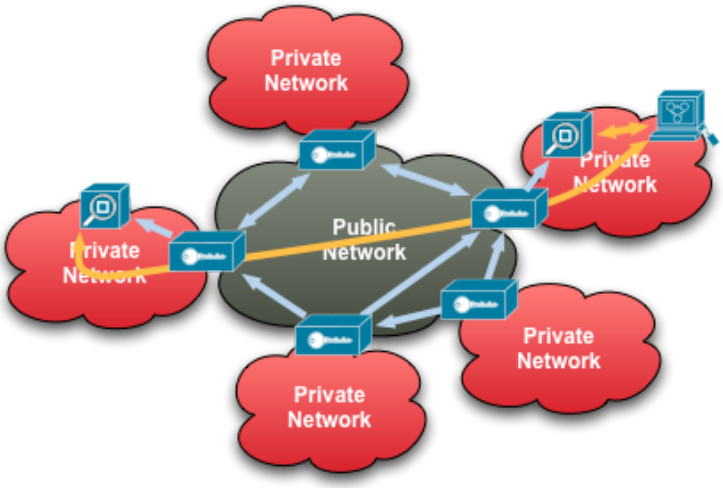

Fig. 4: Creation of two reverse multicast trees for live monitoring

Again, all messages are protected by IPsec and contain a timestamp so that external attackers are assumed to be able to delay messages for a short time or suppress them at most. As there can be multiple independent distribution trees, chances of a successful suppression can be further lowered. The possible influence of internal attackers is also bound to the messages that are passed through them, and as control information and data is protected by asymmetric signatures within the trees, data authentication and integrity is ensured. Thus, compared to external attackers, the only advantage of internal ones is possibility to drop packets more selectively. Furthermore, is the connection between probes and administrators is protected by Transport Layer Security (TLS).

5.4 Other Services

Just like the presented examples for the distributed realization of network services within VPNs, other mechanisms are needed to distribute software and certificate data, perform logging, and to provide means for distributing access control lists (ACLs), which configure client-side firewalls within VPN gateways, or revoking potentially compromised certificates (e.g. by CRLs).

And while a feasibility of service applications with very different distribution patterns has been illustrated in this article, especially the ACL and CRL functionality does not only require an integrity protected and authenticated delivery, but also a certain guarantee that all affected VPN devices are informed. Hence, the focus of our future research activities will con-

centrate on the scalable and secure creation of node-disjoint delivery paths through the VPN in order to tolerate a certain fraction of compromised nodes.

6 Conclusion

While the automatic deployment of VPNs has been discussed in science as well as network industry for already a decade, many of the availability issues cannot be resolved without fully distributed approaches like SOLID. However, the discussion in this article shows that a distributed configuration can only be a first step, because many higher layer network services are currently not fully accustomed to flexible network changes. Hence, the mechanisms of distributed alternatives to three common, diversely structured services were presented.

As also outlined, the decentralization of CRL and ACL transmission requires a scalable system to distribute them over node-disjoint paths in order to achieve a tolerance against internal attackers. Further research will also concentrate on more resilience against external DoS attacks, by automatically creating stable topologies. The creation of distributed network services allows also for the ad-hoc creation of mobile VPN, i.e., in disaster scenarios, which will be also in focus of our further research.

References

[SMK+01] Stoica, Ion ; Morris, Robert; Karger, David ; Kaashoek, M. F.; Balakrishnan, Hari: Chord: A scalable peer-to-peer lookup service for internet applications. In: ACM SIGCOMM Computer Communication Review 31 (2001), Nr. 4, S. 149–160

[SAS+02] Stoica, Ion ; Adkins, Daniel; Shenker, Scott ; Surana, Sonesh; Zhuang, Shelley: Internet Indirection Infrastructure. In: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM), 2002, S. 73–86

[RRKC04] Raghunath, Satish; Ramakrishnan, K. K.; Kalyanaraman, Shivkumar; Chase, Chris; Measurement based characterization and provisioning of IP VPNs, ACM SIGCOMM, 2004.

[Tran05] Tran, Trung: Proactive Multicast-Based IPSEC Discovery Protocol and Multicast Extension, IEEE MILCOM, 2005.

[Aura05] Aura, Tuomas: Cryptographically Generated Addresses (CGA), IETF RFC 3972, 2005.

[Flur07] Fluhrer, Scott: SYSTEM AND METHOD FOR PROTECTED SPOKE TO SPOKE COMMUNICATION USING AN UNPROTECTED COMPUTER NETWORK, United States Patent US 2007/0271451 A1, 2007.

[Bhaj08] Bhaiji, Yusuf: Network Security Technologies and Solutions, Cisco Press, 2008.

[RoSc09] Rossberg, Michael; Schaefer, Guenter: Ciscos Group Encrypted Transport VPN – Eine kritische Analyse, D-A-CH security, 2009.

[Gole09] Golembewski, René: Live Visualisierung virtueller privater IPsec Netzwerke, Student Research Project, Ilmenau University of Technology, April 2009.

[Schu10] Schüttler, Florian: Sichere dezentrale Namensauflösung in IPsec-Infrastrukturen, Bachelor Thesis, Ilmenau University of Technology, January 2010.

[Gole10] Golembewski, René: Sichere, verteilte Zeitsynchronisation in virtuellen privaten Netzwerken, Diploma Thesis, Ilmenau University of Technology, March 2010.

[RoSS10] Rossberg, Michael; Schaefer, Guenter; Strufe, Thorsten: Distributed Automatic Configuration of Complex IPsec-Infrastructures, To appear: Journal of Network and Systems Management, Septermber 2010.

[RoSc10] Rossberg, Michael; Schaefer, Guenter: A Survey on Automatic Configuration of Virtual Private Networks, Submitted to: Computer Networks, 2010.

Index

Virtual Private Networks (VPNs), Self-Configuration, Availability, DoS-Resistance