Noname manuscript No.
(will be inserted by the editor)

Distributed Automatic Configuration of Complex IPsec-Infrastructures

Michael Rossberg · Guenter Schaefer · Thorsten Strufe

the date of receipt and acceptance should be inserted later

Abstract The Internet Protocol Security Architecture IPsec is hard to deploy in large, nested, or dynamic scenarios. The major reason for this is the need for manual configuration of the cryptographic tunnels, which grows quadratically with the total amount of IPsec gateways. This way of configuration is error-prone, cost-intensive and rather static. When private addresses are used in the protected subnetworks, the problem becomes even worse as the routing cannot rely on public infrastructures.

In this article, we present a fully automated approach for the distributed configuration of IPsec domains. Utilizing peer-to-peer technology, our approach scales well with respect to the number of managed IPsec gateways, reacts robust to network failures, and supports the configuration of nested networks with private address spaces. We analyze the security requirements and further desirable properties of IPsec policy negotiation, and show that the distribution of security policy configuration does not impair security of transmitted user data in the resulting virtual private network (VPN). Results of a prototype implementation and simulation study reveal that the approach offers good characteristics for example with respect to quick reconfiguration of all gateways after a central power failure (robustness), or after insertion of new gateways (scalability and agility).

Keywords Computer Network Reliability, Computer Network Security, Robustness

1 Introduction

The Internet Protocol Security Architecture *IPsec*, defined by the Internet Engineering Task Force, is one of the most deployed protocol architectures to set up virtual private networks (VPNs). Generally, these VPNs have a similar setup (cmp. Fig. 1): There are different subnetworks each representing a company site or department. IPsec gateways connect these parts by directly tunneling through untrusted networks or over one or more other IPsec gateways in nested scenarios. The trusted networks in these VPN typically use private IP address ranges, and they may have multiple VPN gateways for reasons of load balancing and failure tolerance.

Managing, e.g., the internetworking of the police departments within a country, or the office sites of a large multinational company, creates a large, complex, and often fully-meshed VPN, generally protected using IPsec. Considering these scenarios, the task of configuring the VPN suffers from severe scalability problems: the number of security associations that commonly have to be configured and maintained manually, grows quadratic with the number of IPsec gateways. Thus, not only the administrative costs grow [1], but also the probability of human errors. Furthermore, a growing number of IPsec gateways in a VPN directly lead to a growth of the security policy database (SPD), which stores information on how security associations

Michael Rossberg · Guenter Schaefer
Technische Universität Ilmenau
Telematics and Computer Networks Group
Postfach 100565
98684 Ilmenau
Germany
E-mail: first.lastname[at]tu-ilmenau.de

Thorsten Strufe
Technische Universität Darmstadt
Peer-to-Peer Networking Group
Hochschulstraße 10
64289 Darmstadt
Germany
E-mail: strufe[at]cs.tu-darmstadt.de

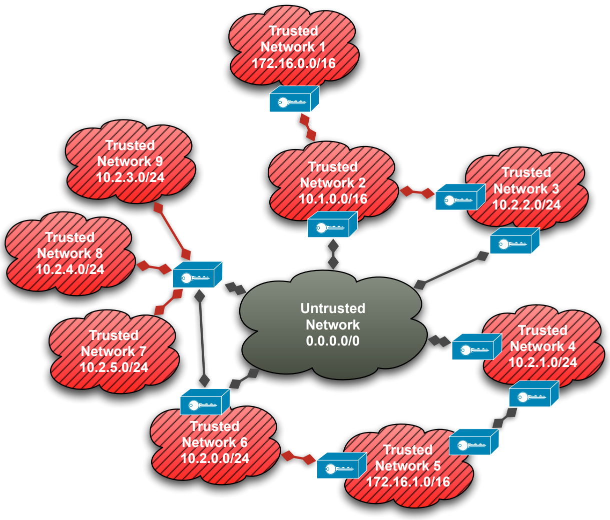
Trusted Network 1 172.16.0.0/16
Trusted Network 9 10.2.3.0/24
Trusted Network 2 10.1.0.0/16
Trusted Network 3 10.2.2.0/24
Trusted Network 8 10.2.4.0/24
Untrusted Network 0.0.0.0/0
Trusted Network 7 10.2.5.0/24
Trusted Network 4 10.2.1.0/24
Trusted Network 6 10.2.0.0/24
Trusted Network 5 172.16.1.0/16

Fig. 1 An IPsec infrastructure scenario

are established in each gateway. The additional resource demand must be compensated with increased memory and processing capabilities, especially when considering the simultaneous reestablishment of security associations after networking failures. This is not the only way in which the robustness of VPNs can be impaired by a manual configuration: In the case of partial failures of the transport network, IPsec in theory allows for a redirection of traffic through nested tunnels over other IPsec gateways of the VPN. However, such a reconfiguration cannot be performed in adequate time, if done manually. Furthermore, the dynamic nature of a VPN itself may prevent a timely manual reconfiguration. Office branches might change location, departments might be consolidated or restructured, and the providers that offer access to the Internet might be exchanged for cheaper or otherwise more convenient Internet service provider (ISP). All of these actions, as well as common dynamic IP address allocation practices of ISPs, lead to a change of the external IP address or address range of the concerned parts of the network. This problem gets more severe when considering mobile IPsec gateways that frequently change their geographic location and their connectivity. In consequence, joining or removing single IPsec gateways requires changes in the configuration of not only the affected subnetwork, but in the configuration of all other IPsec gateways as well, as they have to update the corresponding entries of their SPD. As a result, updates for dynamic, large, or topology-adapting VPNs are infeasible in a timely manner, as long as manual interaction is required.

Hence, an automatic approach to the configuration is required in order to allow for a deployment in large-scale and dynamic environments. To be suitable for the sketched general usage scenarios, the configuration mechanism needs to cope with nested security associations and private address spaces. This leads to the core problem that is to be solved: the dynamic mapping between the external IP address of an IPsec gateway and the internal address range of the trusted network it represents. This issue occurs whenever an IPsec gateway has to forward a packet to another IPsec gateway, since the packet contains an internal destination address and the gateway needs to be contacted by its visible external address.

Aiming at the scenario of a large scale automatic configuration on the Internet, the mechanism has to rely on existing and well available underlying technologies and services, only. Network multicast, e.g., cannot be considered available and the mechanism in consequence has to solely rely on unicast communication. The system must scale over the number of IPsec gateways and react robust to failures of IPsec gateways as well as of the transport network.

In this article, we make the following contributions:

– We analyze the security objectives and further requirements of IPsec policy configuration in complex scenarios, motivating the need for a fully distributed and automated solution.
– We present **S**ecure **O**ver**L**ay for **I**Psec **D**iscovery (SOLID), our approach to automatic configuration of IPsec-based VPN. 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, allocation 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.
– We detail the architecture and implementation of our SOLID prototype, which we use on our local IPsec gateways for automatic VPN configuration.
– While robustness and scalability are well studied properties of peer-to-peer systems, one major concern is whether security can be guaranteed in this context like in a common centralized system. Therefore, we analyze the security implications of SOLID and are able to show that the automatic configura-

tion does not impair the security of user data in the managed VPN.
- Finally, we present results of a simulation study, which reveals that SOLID scales well with the amount of IPsec gateways, and reacts robust to node failures.

The rest of the paper is organized as follows: section 2 describes background on the problem of VPN configuration and section 3 states the objectives for an automated configuration mechanism for IPsec policies. Related work is discussed in section 4, and in section 5 we explain our assumptions on the fundamental security approach, followed by a discussion of the core functions to be performed by an automatic configuration system. SOLID itself is described in section 6, followed by a basic description of a prototype in 7. Section 8 evaluates the system with respect the objectives by analysis and a simulation study. Section 9 concludes our work.

2 Background

A typical security infrastructure, like the one in Fig 1, is set in an environment consisting of three basic types of components: two or more *trustworthy "red" networks* (striped), one or more *untrusted "black" networks*, such as the Internet, and *IPsec gateways* to securely connect the trusted parts. Single computers with an IPsec association are special atomic cases of both an IPsec gateway and its related network.

On the one hand from a protocol point of view, IPsec gateways are interconnected using cryptographic tunnels to securely exchange arbitrary data. In a manual approach each of these tunnels is first configured by an administrator by specifying appropriate rules in the SPD, and then automatically established by an Internet key exchange (IKE) daemon. Thus, the time required to setup a VPN depends on two factors: First, if an IPsec gateway is added to the VPN for the first time or after an IP address change, the administrator needs time to update the configuration of the gateway and of all other gateways that shall be connected to it. Second, the IKE daemon requires time to establish the IPsec associations, with the time to process asymmetric signatures being the primary influencing factor. Especially in high security scenarios this poses to be a serious problem, as the processing of cryptographic functions generally are handled by smart cards in this case.

From a topological point of view on the other hand, the IPsec gateways may be connected in different ways: Every single IPsec gateway may be responsible for multiple separate trusted networks and may have one or more uplinks to untrusted network parts, thus creating complex structures as illustrated in Fig. 1. Nested networks, e.g., allow an additional protection of vulnerable departments within the same company building, multihop communication, or a better resilience against selective failures within the transport network. Furthermore, multiple connectivity of trusted networks may lead to loops within the graph of subnetworks.

Another particularity of VPNs is the common use of use of private, rather than public address spaces for the trusted networks, for the following reasons:

- The address allocation of public IP addresses for every device is bureaucratic and inflexible.
- The private allocation scheme presents a supplementary security barrier as packets from and to the Internet are normally not routable.
- As a consequence, it is not easily visible for outside attackers, how many devices are operating in a particular organizational unit.
- Routing updates for the trusted networks can be kept within the organization, leading to both: a better infrastructure hiding and the possibility to change the location of a network without the need to update internal IP addresses.

3 Objectives

According to this scenario, automatic configuration systems should fulfill the following objectives:
1. Minimal manual management: A main target of automatic configuration is efficiency in terms of human interaction. This leads to reduced administrative costs and to avoiding human errors.
2. Only unicast communication: In order to make use of the system via the Internet, an approach has to rely on existing and deployed communication services only and in consequence may not rely on multicast, anycast, or broadcast.
3. Support for nested trusted networks: Networks with nested IPsec gateways require a configuration system that includes functionality for complex routing. It has to facilitate finding and selecting efficient paths over multiple IPsec security associations between any pair of gateways of the VPN, as some gateway pairs are connected through forwarding gateways and hence unable to communicate directly.
4. Support for private address spaces: Inner trusted networks have to be addressable by private addresses (like 10.1.0.0/16 in Fig. 1). Hence, public routing mechanisms of the lower IP layers cannot be utilized for these ranges.
5. Security: An automatic configuration system must offer the same security as a comparable, manually man-

aged VPN, and thus fulfill confidentiality, integrity, accountability, controlled access, and availability.

6. Robustness: In order to be fit for commercial or governmental use, the system has to exhibit a high robustness, even in adverse conditions. High packet loss rates, concurrent startup of all managed devices after a correlated failure, temporary partitioning of the network, and DoS attacks have to be kept in mind as feasible incidents, which must not cause the system to stall or fail.

7. Scalability: As the networks managed by the mechanism might grow to thousands of IPsec gateways, the auto configuration has to scale over the total number of trusted networks and gateways.

8. Agility: In order to account for the mobility of users as an increasingly important characteristic and therefore allow IPsec gateways to quickly change their public IP addresses, the scheme has to cope with frequent insertion and removal of these agile IPsec gateways. Supporting agility additionally will support the requirement for high robustness (objective 3), too, as node failures are also handled quickly if agility is assured.

4 Related Work

Several other systems for the automatic IPsec configuration of VPNs exist today, which implement a wide range of different concepts. The Security Policy Protocol [2,3] and Tunnel Endpoint Discovery [4,5] use a similar approach: both depend on lower layer services to configure IPsec. Discovery of other trusted networks is performed by an IPsec gateway through sending an association request directly to the target device in another trusted network. This request is then intercepted by the destination IPsec gateway and a security association is initiated between the IPsec gateways. The main drawback of these approaches is the requirement for public, routable IP addresses for all involved trusted networks and their devices.

In [6] a proactive discovery using multicast announcements is suggested. While supporting private address spaces, scalability in the number of IPsec gateways is not given since the total amount of received announcements is of the order $\mathcal{O}(n^2)$ (n being the gateway count). Furthermore, it cannot be deployed in the Internet as it requires network layer multicast services. It additionally cannot configure nested networks.

A complex configuration infrastructure is part of Cisco's Dynamic Multipoint VPN (DMVPN) architecture [7,8]. It interconnects several static IPsec gateways, so called “hub routers", using static IPsec tunnels. Subsequently, gateways with dynamic addresses may connect to those “hub routers". Furthermore, on demand IPsec associations will be created between mobile gateways connected to the same hub router. Even though this approach allows the creation of more flexible infrastructures, it still depends on static IPsec gateways. This makes their availability critical for all other nodes. Nested security associations, connecting gateways using paths over multiple dynamic gateways, are impossible using this approach. Another drawback is the necessary manual interaction as in DMVPN a substantial amount of configuration data has to be maintained manually.

Cisco's most recent step towards an IPsec automatic configuration is the Group Encrypted Transport (GET) VPN [9]. It relies on a central key server that periodically distributes a shared symmetric key to the IPsec gateways of a VPN, which thereupon are able to mutually communicate. The distribution can either be done via multicast or individually via unicast. The first approach depends on the transport network to provide multicast services, which is not given for the Internet in general, and the latter does not scale over the number of gateways. On top of these drawbacks, GET VPN only supports private address ranges, if and only if, they are routed by the transport network. Apart from functional deficits, GET severely weakens the security [10] offered by standard IPsec solutions in many aspects. For example, the use of group keys for one-to-one communication leads to the loss of data confidentiality within the whole system in case of a compromise of a single gateway. Forward secrecy is not provided, and the anti-replay protection is based on time windows, thus enabling attackers to perform unlimited replays within a given time period.

The most recent approach is Social VPN [11], where SPD entries are created based on contact lists of Facebook accounts. Even though it uses peer-to-peer mechanisms to lookup IP addresses of security associations, it is meant to serve for end-to-end connectivity, only. Thus, it neither provides support for IPsec gateways, nor allows indirect connections. The social network aspect on top leads to a weak trust definition [12].

The deficiencies in the discussed protocols stimulated the development of the novel configuration system detailed in the next sections.

5 Automatically Configuring IPsec

In order to define the scenario for configuring IPsec VPNs, and to derive the requirements for automatic configuration systems, we describe the context, basic assumptions and core functions that have to be implemented.

5.1 Assumptions

Two basic assumptions regarding the roles and relations of IPsec gateways and trusted domains are taken in this article:

All IPsec gateways are equally trusted. It is assumed that all trusted components of the system belong to the same security level, i.e., they may forward data between each other. All participating red networks in consequence are trusted equally, there is no differentiation between the level of trust in different IPsec gateways that are participating in the same VPN, and hence there is no need to introduce different roles for access control between the IPsec gateways. Credentials and cryptographic algorithms are assumed to be stored and executed on trusted devices (smart cards), in accordance to common use in the targeted environment. Certificates and security credentials cannot be extracted from these devices and therefore there is no danger for them to be duplicated. The trusted IPsec gateways in consequence can be used in relatively untrusted environments, and all IPsec gateways therefore trusted equally. Please note that our trust assumption does not imply that an IPsec gateway can decrypt data that it relays for other gateways.

All IPsec gateways are cooperative. Since all IPsec gateways are of the same trust level, data packets can be relayed on any path through the IPsec VPN.

These assumptions pose a limitation to generality. Diverging environments are certainly conceivable, like, e.g., a VPN spanning devices of domains with different trust levels, which can not entirely be considered cooperative. Our current approach is based on these assumptions, and we sketch the necessary extensions to facilitate these circumstances in the future work section.

5.2 Core Functions

A fully automatic IPsec configuration system for VPNs needs to perform the following five subtasks:

Bootstrapping: Whenever an IPsec gateway joins an untrusted network, it needs to contact some other gateway, which takes part in the VPN that the new gateway aims to join. It establishes a security association to the identified gateway, which then can be used to coordinate any further tasks.

Address Allocation: After the initial bootstrapping process, the joining IPsec gateways need to allocate address spaces that they subsequently can assign to devices in the trusted networks they manage. While these addresses may be private and not globally routable, it has to be assured that they are unique within the VPN.

Topology Control: In order for any pair of IPsec gateways to be able to spontaneously communicate and to avoid unnecessary delays, a network of IPsec tunnels has to be maintained permanently in the VPN. Considering a full mesh overlay network for these purposes leads to a high, and most probably unnecessary messaging overhead and thus is not feasible for reasons of the scalability of the approach. In consequence, each gateway proactively has to establish a subset of security associations. This subset has to be selected such that introduced delays are minimal, in order to assure a timely communication between any pair of gateways. The topology control manages the proactive setup of these tunnels between selected IPsec gateways.

In contrast to bootstrapping and address allocation, topology control is a continuous task as the topology needs to be permanently adapted in relation to the changes of the VPN and the traversed networks.

Discovery: In case a packet needs to be forwarded by an IPsec gateway that does not have a matching SPD entry for the destination subnetwork, a discovery service is needed. This discovery has to implement a systematic search for IPsec gateways, based on arbitrary IP addresses belonging to the range of the subnetworks they manage.

Routing: In a nested environment, the discovery service has to be extended: it is not sufficient to only resolve the identity of a gateway, but in addition, means to determine a path through the overlay between the concerned gateways are needed.

Some requirements to these functions arise in order to meet the objectives in section 3: In order to ensure availability, all of these subtasks have to be solved in an entirely distributed way as a centralized approach creates a single point of failure and potential performance bottleneck.

6 Secure OverLay for IPsec Discovery

VPNs are by their nature special instances of overlay networks, which perform the service of confidential data transfer. With respect to this fact, we design and implement a system for the automatic configuration of IPsec VPN leveraging well understood concepts of peer-to-peer systems. In order to implement a self-managing overlay, a logical ring over the complete namespace of managed private IP addresses is created, as has been proposed similarly by other systems [13], too. The security gateways are connected to their neighbors in the ring, based on the address ranges of the trusted networks they represent. Using this overlay, a lookup is implemented which allows for a dependable discovery of an IPsec gateway that is responsible for certain, trusted

IP addresses. In consequence it is possible to establish associations that can be used to securely communicate between two arbitrary gateways without manual interaction.

The major reason for creating a virtual ring lays within the cost of proactively establishing security associations. Every security gateway only has to establish two security associations to be connected in this structure and to be discovered by others. Thus, the method allows for a fast integration of new security gateways.

Especially in complex networks with nested or multihomed subnets, using the evolving ring structure to relay all traffic poses high messaging overhead and is not efficient: With all gateways being ordered by their managed IP address ranges on the ring, routing packets from a source to a destination network along the ring induces an overhead on the path length, which potentially is super linear in the number of participating gateways, as some connections between two gateways can be indirect and thus traversing a number of additional gateways.

Consider the path from the trusted network 2 to trusted network 7 in Fig. 1. If sent along the shortest path in the ring structure it is necessary for the packet to traverse network 5 and network 1 (see also Fig. 2) . For reasons of the physical connectivity this results in a path via the networks 2, 6, 5, 6, 2, 1, and again 2, to finally reach network 7 – even though a direct connection between the gateways to TN 7 and TN 2 over the untrusted network was possible. In order to decrease this overhead, the network of IPsec associations will be adapted to reach higher efficiency.

Despite seemingly introducing complexity at first, the required manual management overhead to add a new IPsec gateway of SOLID is reduced to a bare minimum. SOLID only requires the administrator to generate a public/private key pair, retrieve the corresponding certificate by the certificate authority (CA) and write both to a new smart card.

Only if the untrusted side requires parameters such as Domain Name System (DNS) entries for Internet bootstrapping, DSL passwords, or WLAN passwords, these additionally have to be configured which, however, can be done using the same card. Thus, upon deployment IPsec gateways do not require any manual configuration to participate in the VPN, besides the appropriate smart card constraining its public/private key pair.

The system is designed according to the five core functions stated in the previous section 5.2. While for the tasks of *bootstrapping* and *address allocation* existing approaches are selected, the *topology control*, *discovery* and *routing* are developed, designing novel configuration algorithms.

6.1 Bootstrapping

Different schemes have been proposed for *bootstrapping* a decentralized system [14,15]. SOLID implements a combination of existing techniques: In style of an escalation plan, each IPsec gateways use the following methods successively:

Local Cache: If an IPsec gateway returns to a VPN after a temporary disconnection, it uses a local cache to connect to IP addresses that were used by previously known gateways.

Connect to Internet Gateway: In nested scenarios the assigned default Internet gateway by nature is part of the VPN as well and the joining gateway can establish a security association to it.

Directory Service: In networks that support unicast exclusively, directory services, such as DNS and Lightweight Directory Access Protocol (LDAP), are the only way to perform a systematic bootstrapping. Replication and group-wise encryption assure the required security objectives in this case.

Any-/Multi-/Broadcast: In networks that support extended addressing techniques, such as LANs, MANs and MANETs, any-, multi- or broadcasting is used as a last resort.

6.2 Address allocation

Distributed *address allocation* approaches, used in mobile ad-hoc networks, can also be adapted to work within the given IPsec scenario. The binary split mechanism [16,17] was chosen to be used with SOLID as it works entirely distributed and does not rely on duplicate address detection. Hence, it is scalable and robust against network partitions. In order to prevent over-claiming, which might happen for reasons of mistakes or malicious behavior, certificate chains as address attestations are distributed together with the allocated address ranges to the IPsec gateways. The root certificate needed for these attestations is distributed in the initial offline configuration process.

An alternative for this approach is the possibility to statically assign IP address ranges to each SOLID gateway. As the IP addresses are not required to have any particular structure, the ranges can simply be chosen sequentially and embedded within the certificate of the IPsec gateway. This static allocation also reduces the exposure to attacks, if the flexibility of dynamic address allocation is not required.

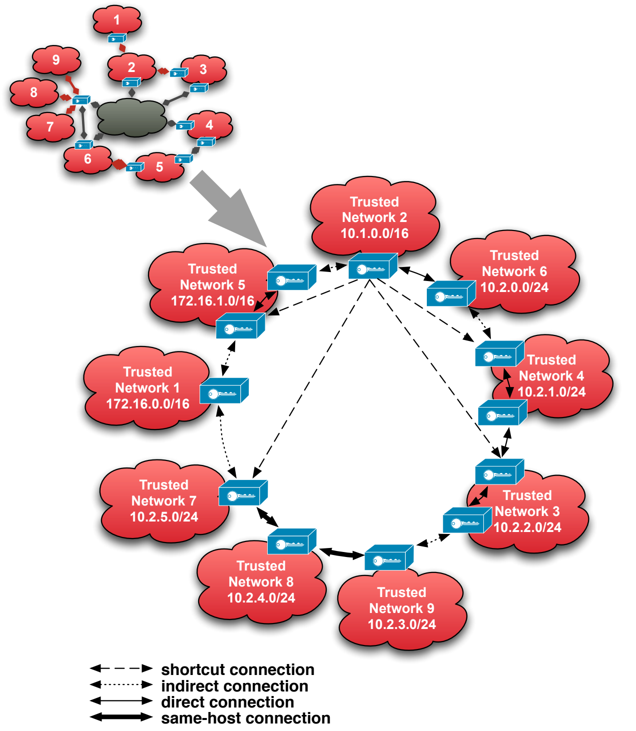
1
9
2
3
8
4
7
6
5
Trusted Network 2 10.1.0.0/16
Trusted Network 5 172.16.1.0/16
Trusted Network 6 10.2.0.0/24
Trusted Network 4 10.2.1.0/24
Trusted Network 1 172.16.0.0/16
Trusted Network 7 10.2.5.0/24
Trusted Network 3 10.2.2.0/24
Trusted Network 8 10.2.4.0/24
Trusted Network 9 10.2.3.0/24
shortcut connection
indirect connection
direct connection
same-host connection

Fig. 2 Corresponding overlay structure of the IPsec infrastructure scenario

6.3 Topology Control

In an initial step, the *topology control* part of the SOLID algorithm creates a structured overlay network with all known IPsec gateways as nodes. In order to perform this task, all trusted network addresses are interpreted as values in an algebraic ring structure. Within this structure, each network has a preceding network as well as a succeeding one. Ordered in a two dimensional space they form a ring. The IPsec gateways proactively construct tunnels to their direct predecessors and successors on the ring, which are used to securely transmit all management as well as payload traffic. Any two IPsec gateways that may not communicate directly to each other through the underlay, i.e., they are separated by other security gateways or subject to a partial network failure, create a nested IPsec association that is forwarded by other gateways through the VPN. The task of establishing forwarding paths over other gateways is performed by SOLID's routing functionality.

Fig. 2 shows an overlay ring structure that corresponds to the example infrastructure given in Fig. 1. The solid lines between IPsec gateways show proactively created IPsec tunnels. Dotted lines indicate IPsec tunnels that are routed through intermediate systems for reasons of a nested setup. Thick lines show virtual associations between IPsec gateway processes on the same device due to the connection of multiple secured subnetworks through a single IPsec gateway.

In addition to the ring associations, shortcuts are set up to connect different parts of the ring, as illustrated by the dashed lines in Fig. 2 (for reasons of visible clarity only the shortcuts selected by the gateway to the trusted network 2 are shown). They are chosen such that they bisect the address space with every overlay hop, which guarantees that the longest of all shortest paths between any two gateways is within logarithmic order in the number of participating IPsec gateways.

Unlike conventional distributed hash tables (DHTs), in which a resource is stored at the node with the ID that is closest to the resources' key measured by a selected distance metric, SOLID does not use the DHT structure to store any resources. The ring is merely used to order the IPsec gateways along a structure that allows for a deterministic routing to any destination, sorted by the addresses of their respective trusted networks. Furthermore, the key information being the IP address prefix of the inner devices cannot be hashed before gateway insertion and lookup as in the discovery process searches have to be performed considering subnetwork masks of variable lengths. Such a search cannot be performed on hashed values without a substantial additional communication overhead as this would mean to store values or create connections for every possible client in the network.

Since address allocation algorithms in most cases will not create an even spread of the allocated address ranges in the address space, the IPsec gateways consequently are not distributed evenly over the address space. Hence, the system loses the load balancing, inherent to conventional DHT.

As a result, the shortcuts cannot be selected straightforward, based on the overall address space like in comparable peer-to-peer approaches. In order to still achieve a high speed up, the space of used addresses has to be bisected. This can be achieved by estimations that are based on observed packets. Smaller discrepancy from the optimal selection of shortcut associations, does not pose a drawback as the lookup traffic is expected to be rather low in comparison to the amount of transmitted payload.

In order to give an exemplary message sequence of the protocol, consider that an IPsec gateway with the network ID 1 joins the example VPN and for reasons of the allocated address space, it shall be placed between the IPsec gateways with ID 5 and 7 :

1. During bootstrapping the joining IPsec gateway 1 creates a security association to the IPsec gateway with ID 2.

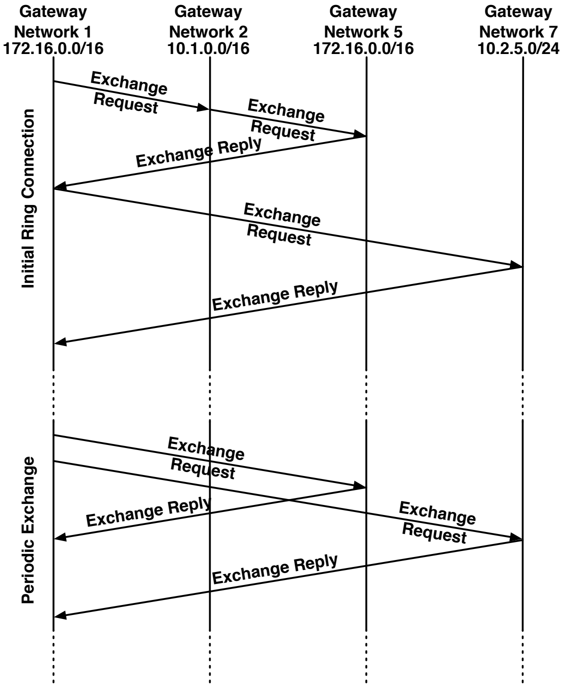

Fig. 3 Association and periodic exchange

2. IPsec gateway 1 sends an exchange request to IPsec gateway 2, which it got to know during the bootstrapping process.
3. The request is forwarded over the ring structure to gateway 5 , which will be the succeeding neighbor of gateway 1 in the ring.
4. Gateway 5 sends a reply via ring structure and gateway 2 to gateway 1, which contains both the identities of gateway 5 and additionally of gateway 7 , being the previous predecessor of gateway 5 and hence the new predecessor of gateway 1.
5. Subsequently, a security association to gateway 7 is created, which thereupon represents the new predecessor.
6. If possible, the communication paths between 1 and 5, as well as 1 and 7, will be optimized by SOLID's routing algorithm to an optimal path.
7. At the following reauthentication cycle, the security association between gateway 5 and 7 may be dropped, unless it is further needed as a short cut or due to high direct communication load between the respective subnetworks.

Fig. 3 shows the sequence of all exchanged messages, only leaving out the IPsec encapsulation, which, for reasons of clarity, is not explicitly shown.

The described exchange request is repeated periodically between ring neighbors to detect node failures and to assure a consistent structure of the ring in this case.

All SPD entries that are created this way are kept in the database. They are removed only when the corresponding IPsec gateways do not react to the periodic exchange requests or if the association is of no further use to the system, e.g. due to topology changes. In this case, they are dropped after a grace period and before computational intensive IKE reauthentications occur.

6.4 Discovery

A key component for the system is the *discovery* service as it is invoked every time a packet needs to be securely transmitted between two gateways, which previously have not established a security association.

If a packet arrives at an IPsec gateway, that does not hold a matching rule in its SPD, it is cached first. In order to locate the IPsec gateway responsible for the respective IP address, the IPsec gateway subsequently sends a discovery request message over the existing associations of the VPN. Implementing a greedy routing, each IPsec gateway forwards this message to the gateway that, with respect to the algebraic ring structure, is closest to the destination address of the original datagram. The ring structure assures that the packet will eventually be delivered to the correct network and the shortcut associations reduce lookup times. The respective IPsec gateway will answer to the discovery request with its outer IP address. The source gateway in consequence can encrypt and forward the original datagram to the destination gateway.

Like the proactively created SPD entries, the associations for data transport are kept active for at least a grace period and are removed, if they are not used for a longer period.

6.5 Routing

In a nested environment, a *routing* has to be implemented by the configuration system as in this case not only the identity of an IPsec gateway has to be resolved, but also a path to it. SOLID's routing always selects valid paths. However, these initially traverse several other gateways on the ring structure and in consequence are longer than necessary.

For example, in the scenario of Fig. 3 the first reply of IPsec gateway 3 would be relayed through IPsec tunnels over IPsec gateway 2 twice.

In order to account for the fact that shortcuts exist for almost all paths, they subsequently are optimized until the shortest possible path is found. The hop count is used as primary metric as the cryptographic operations on intermediate IPsec gateways severely influence delay and available bandwidth.

These comparably long paths hence have to be optimized, which is accomplished in two ways: First, if the outer address of another IPsec gateway is a public IP address, it is likely to be globally routable and a direct network layer communication can be established. Second, if a SOLID node detects that incoming traffic on a network interface is forwarded to another node via the same interface, chances are that all three IPsec gateways are part of the same subnetwork. The forwarding SOLID node can now inform the others of the condition and a direct communication is established.

This routing will generally find an optimal path between any pair of given IPsec gateways. Only in environments, which are characterized by the existence of cycles with a large diameter, the optimization can lead to local minima. In this case packets are forwarded along a path through a set of trusted networks, even though a shorter path through a different set exists. Such suboptimal paths can only be avoided by introducing a local routing protocol. However, this routing protocol cannot be deployed without manual interaction as routing policy decisions, i.e., which path shall be used for forwarding which type of data, can only be done with constituted metrics.

To secure routing the information on the discovered routing paths it is digitally signed, following ideas similar to S-BGP [18]. For most other messages, e.g. shortening, it is sufficient to use nonces and IPsec certificates with embedded valid address ranges, i.e., those from the address allocation process.

7 Prototype

In order to evaluate the SOLID approach, a prototype based on Linux and Charon, strongSwan's IKEv2 daemon, was created. Fig. 4 shows the basic architecture of the prototype as well as the numerous interfaces to different applications, libraries, and the kernel.

A basic message flow of a discovery works in the following way:

1. SOLID receives a packet without corresponding IPsec association via a tun device.
2. The system uses the discovery (as described in 6.4) and searches for the correct IPsec gateway. In this step, the intermediate SOLID gateways create a forwarding path when the search messages are forwarded along the overlay structure.
3. Once the gateways have successfully performed a mutual discovery, a security association along the forwarding path is created via Charon's stroke interface.
4. As soon as the association is established, XFRM will inform SOLID of the event, so that firewall rules can be adopted and the cached packet is reinjected using libnet.

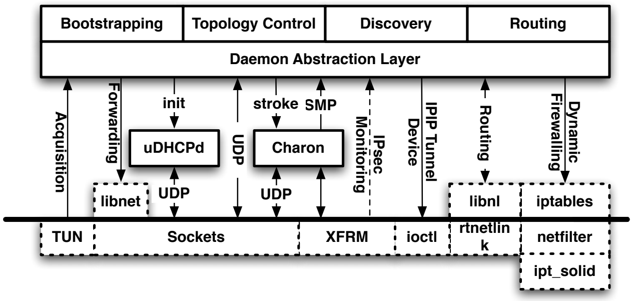

Fig. 4 SOLID prototype architecture and interfaces

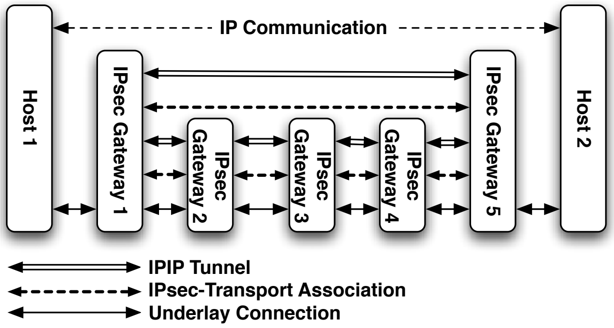

Fig. 5 SOLID's protocol stack

All other interfaces are used for initialization of supplementary subtasks, such as providing a DHCP (Dynamic Host Configuration Protocol) server to client computers.

A peculiarity of the prototype is the use of IPIP tunnels within IPsec transport associations to emulate the behavior of IPsec tunnel associations. This mode of operation (sometimes called IIPtran [19]) allows not only for a better routing integration, but also the creation of nested associations on a single host, which poses currently a problem for many IKE daemons. Fig. 5 shows the resulting protocol architecture for a small sample setup. IPsec gateways are directly connected by transport associations, all routed packets are encapsulated by IPIP. An inner tunnel is created to forward payload packets for nested networks.

As an alternative the architecture of the prototype allows for an easy integration of the OMNeT++ simulator. For this purpose, the Daemon Abstraction Layer only needs to be replaced by a lightweight simulation library, while the rest of the code stays unmodified. This concept is used in the following section to evaluate the most important properties of SOLID.

8 Evaluation

We evaluate SOLID in three different steps. In order to assess the qualitative objectives of the approach, we present an analytical discussion in the first subsection. Subsequently, we perform an analytic evaluation of the security objectives. Testing if SOLID meets the performance objectives requires a simulation study that is covered in the last part of the evaluation section.

8.1 General Discussion

SOLID meets some of the objectives for an automatic configuration system for IPsec VPN by design, as shown in the following.

Minimal manual management: In conjunction with a bootstrapping and address allocation system, SOLID is able to configure IPsec infrastructures in an almost entirely autonomous fashion. The only manual configuration required at each security gateway is to supply it with a certificate that serves as an attestation for the identity and the permission to participate in the VPN.

Only unicast communication: The SOLID system uses only unicast packets and does not rely on any special system, allowing a deployment in any common networking environment. Broadcast or possibly even multicast may additionally easily be introduced as an extension to the bootstrapping, but are not necessary for SOLID to operate.

Support for nested trusted networks: The routing functionality assures the connectivity in the case of nested security associations and therefore allows for a nesting of trusted networks.

Support for private address spaces: Introducing the discovery service and the mapping of address ranges to the overlay structure allows for the use of solely private address ranges. In consequence, SOLID by default does not use public routed IP addresses, despite its capability of managing these as well.

Robustness: The decentralized structure of the configuration system supports a robust reaction in extreme situations as there is no single point of failure. For the purpose of entirely exploiting this advantage, additional naming and session management services like DNS, SIP, etc., may be replaced by decentralized solutions like [20, 21, 22]. However, in this work we focus on the automatic configuration of the underlying VPN. The quantitative aspects of robustness are further studied in the subsequent section 8.3.

Scalability: The absence of a central structure is a precondition for scalability over the total count of IPsec gateways, too. Detailed quantitative aspects are also discussed in the following section. Nonetheless, from an administrative point of view the system scales very well as it no longer requires a quadratic amount of security policies to be configured manually.

Agility: In order to support mobile IPsec gateways the configuration system needs to cope with a highly dynamic behavior of some IPsec gateways and very frequent changes in the topology. The idea of a structured overlay network is to keep knowledge in the network structure itself and to avoid storing the outer identities of IPsec gateways in all other gateways. Therefore, a change of a single identity does not involve the usual routing database update in all IPsec gateways, but the initial setup of a small constant amount of security associations, i.e., the moving gateway must only create two IPsec associations to its ring neighbors to be fully reachable again. In this way, the costs of a network transition are minimized. A quantitative evaluation follows in section 8.3.

8.2 Security Discussion

Implementing a system with implications on a security infrastructure in a decentralized fashion might seem paradox at first. However, it is the only way to maximize the availability, as any central entity represents a single point of failure and in consequence is an attractive target for DoS attacks. The introduced configuration system does not generate such specific targets, therefore it is solely dependent on a working transport connection between communicating entities. However, distributing the security configuration comes at the risk of partial disclosure when one or a subset of security gateways is compromised. In consequence, special care has to be taken to protect the remaining security objectives and to guarantee the security of the whole system at a level of a comparable, centralized, system.

The following security evaluation is structured into four parts: After defining the assumptions and an attacker model, SOLID's security is analyzed with respect to these increasingly powerful attackers.

8.2.1 Assumptions and Attacker Model

SOLID is designed to be run in a transport network that provides an Internet-like underlying communication infrastructure. This means that it does not rely on any security services being offered by the underlying layers, but only on pure node connectivity.

Within the environment, we consider realistic attackers with different capabilities:

- **External attackers:** On the physical layer, attackers might be able to eavesdrop and replay messages in a LAN environment, in case of wireless communication or if the attacker controls a relevant Internet router they are even capable of dropping, delaying, modifying, and inserting forged messages.
  Furthermore, they can send arbitrary traffic to one or more IP addresses, thus congesting links, overloading IPsec gateways, or possibly even routers on the path to the targeted destinations.
- **Compromised devices in trusted networks:** Attackers who are in control of a device within a trusted network are considered to be able to eavesdrop or modify all data sent from or received by devices within this particular network. Furthermore, they may send arbitrary data through the SOLID gateway and thereby initiate IPsec associations to transmit the data.
- **Compromised SOLID gateways:** If attackers control one or more SOLID gateways or have gotten access to one or more private keys, they are able to intercept traffic of other gateways in the event of indirect communication. The compromised gateways may also deviate from the SOLID protocol and thus perform routing attacks.

For all of our security considerations, we account the validity of security schemes that conventionally are understood as secure. In particular, the attacker cannot easily solve hard problems or break one-way functions, i.e., invert a hash or decrypt ciphertext without possession of the respective key.

8.2.2 External attackers

Considering external attackers, SOLID does not introduce any possibilities for additional attacks. All traffic, configuration datagrams and payload, is subject to IPsec end-to-end encapsulation and hence SOLID VPNs are as secure as any other properly configured IPsec VPN. The system in conclusion is secure against outsider attacks with regards to confidentiality, accountability, and data integrity. Controlled access is guaranteed by the IPsec authentication in conjunction with a CA that provides certificates only to trusted partners.

With regards to infrastructure hiding, externals may gain information about the internal address structure by observing which associations are proactively setup. However, the risk is fairly manageable as SOLID does not require addresses to have a particular structure, and thus the observations do not lead to any vital disclosure.

8.2.3 Compromised devices in trusted networks

Considering an attacker that has managed to get in a position of controlling hosts, or other end devices inside the VPN, the possible attack opportunities are much more complex, as traffic local to the VPN in this case can be observed and modified. This fact is not particular to SOLID's configuration, but a natural limit of the influence of VPN systems. The only possibility to secure all network devices is a mandatory IPsec protection of all datagrams within the trusted network itself, which would require an additional client configuration. Apart from eavesdropping and modifying local network traffic, an attacker can use a device in a trusted network to launch a simple exhaustion attack to achieve a DoS of the local IPsec gateway: Transmitting a large number of packets to different, random subnetworks, will lead to a setup of an arbitrary amount of new IPsec associations. The required asymmetric cryptographic operations pose high load and in consequence may exhaust the computing- or smart card resources of the IPsec gateway.

A side effect of this DoS of one of the security gateways is a slight influence on the overall VPN topology: The topology control optimizes the VPN topology for efficiency and in consequence aims at creating shortcut associations, as described in section 6.3. The event of a failing IPsec gateway in consequence may lead to the creation of new shortcuts in the VPN. In this case, in order to maintain the bisectioning, the topology control additionally may decide to upgrade a suitable existing security association to a long-term VPN connection. Hence, an attacker by achieving a DoS of the local security gateway may influence the VPN topology to some extent, and may be able to cause the creation of additional security associations at the same time. Due to the scalability of SOLID, this attack does not have any significant impact on other IPsec gateways, and other than disconnecting the local subnetwork from the VPN, it has no influence on the security at all. If the clients within the trusted network are authenticated by IPsec themselves, it is also possible to simply control and limit the rate for each client. Thus, the impact of such an attack is reduced significantly.

8.2.4 Compromised IPsec gateways

An even more sophisticated attacker might be able to control an IPsec gateway or acquire a valid IPsec gateway certificate. In this case, it can perform all the attacks described in the last section as well as some limited attacks regarding the data that it forwards. In any case, the attacker cannot eavesdrop or modify payload data without detection as even for indirect communication all data is encrypted and integrity protected by an end-to-end IPsec tunnel (as shown in Fig. 5).

However, due to the fact that some VPN payload is relayed along the SOLID overlay, this position enables the attacker to monitor the amount of transmitted data between two other IPsec gateways, and to perform attacks on the availability, such as gray hole or black hole attacks [23] by forwarding only selected packets, or even none at all. This is a well known vulnerability in any multi-hop environment and the only currently known circumvention is redundant data transmission over disjoint paths, which is very cost intensive [24] and not always possible.

However, as outlined before in contrast to routing attacks in normal IP networks, the confidentiality, integrity, and data authentication of every packet is guaranteed by IPsec at all times as the end-to-end payload of the communicating subnetworks is always protected by the tunneled IPsec association between their respective IPsec gateways. Hence, the compromise of an intermediate gateway along the path does not differ from a compromise of transport network routers, which are usually less secured than IPsec gateways. Thus, SOLID does not weaken the confidentiality, integrity, and data authentication in comparison to manually configured IPsec VPNs.

With regards to the configuration protocol, all management messages of SOLID are digitally signed in order to prevent IPsec gateway impersonation, as this would have enabled an attacker to forge announcements, and in consequence to the attacker being able to falsely advertise routes that are shorter than the actually available routes. An attacker in consequence could have attracted traffic to a gateway under his control. However, due to the protection only a limited amount of further attacks are possible in this scenario. By proactively connecting to other IPsec gateways of the domain, the rogue IPsec gateway can perform a sinkhole attack and thus increase its importance, as more other gateways might use it as an intermediate system for forwarding search requests.

An attacker that is controlling multiple IPsec gateways at different points in the VPN may be able to target IPsec gateways that are connected through a chain of nested connections by mounting a wormhole-like [25] attack: By creating a tunnel between each other and thus creating a less nested path, the attacker could persuade the targeted IPsec gateway to route its traffic through the controlled IPsec gateways. This technique may lead again to a sinkhole attack and further possibly to optimized traffic monitoring or DoS attacks.

The illustrated attacks require a very sophisticated attacker and still yield a very limited effect. A further protection against sinkhole attacks is possible, for example by periodically measuring link delays to detect wormholes or using cover traffic to avoid observations of communication flows. However, such techniques involve significant cost in terms of network bandwidth and routing overhead. They were therefore not considered for SOLID.

8.2.5 Discussion

All in all attackers can only affect availability and monitor traffic flows. Coordinated DoS attacks may disconnect a gateway from the VPN more quickly as SOLID creates less associations proactively, but in difference to manually configured VPNs SOLID will automatically recover from this situation. In case of compromised IPsec gateways a number of different sink, gray, or black hole attacks are possible, but they do not pose a significant threat, as they are hard to control, require a very sophisticated attacker, and yield no effects that may not be achieved by compromising routers in the lesser secured transport network.

8.3 Quantitative Analysis

SOLID being a complete automatic configuration system for IPsec VPN is too complex for comprehensive formal modeling. Additionally, even though SOLID is implemented in a prototype, it would have required too much effort to perform adequate measurements in order to properly model key characteristics with significant influence on the performance, like, e.g., delay distributions and realistic user behavior in a small lab environment. In order to still perform a quantitative evaluation of the robustness to correlated failure and startup of nodes, the scalability, and of the agility of SOLID, discrete simulations, based on the simulation framework OMNeT++ and its associated TCP/IP-library INET, were conducted. While the simulator itself is based on the prototype and very detailed, the simulated user model only needs to focus on the security operations for the following two reasons: The first reason is that SOLID is a pure configuration mechanism and, comparing to payload traffic in VPN, generates a negligible amount of overhead. It thus does not generate load significant enough to cause network congestion. Secondly, SOLID is designed for high security scenarios, such as the configuration of the VPN of national police forces, or governmental offices. The security operations, like, e.g., the calculation of IKE signatures, are typically performed by smart cards in these environments. While executing the security operations would not cause any significant delay when using personal computers, or other complex devices, it is a very time intensive task for smart cards, which, in consequence, are the main bottleneck with respect to the performance of SOLID.

Due to the large signature delay and the low required bandwidth, the simulation abstracts from the details of the networking infrastructure. Only the communicating application processes are implemented on top of the TCP/IP models of INET, and the infrastructure with its different types of networks is modeled according to the environment as described in section 2. Robustness, scalability, and agility thus can still be evaluated without loss of generality. All results still depend on the networking characteristics and can be interpreted as a good estimate on the expected behavior of the system in real world scenarios.

8.3.1 System Parameters

The security operations, processed on smart cards, are expected to cause delays that are an order of magnitude higher than any other operation in the system. In order to back this expectation, we measured the intervals of the IKE signature generation between two prototypes as we expect them to be used in a realistic environment. As devices we chose two embedded PCs, each equipped with an AMD Geode LX800 and operated with a strongSwan/Linux system. The experiment covered computing IKE signatures with different key lengths of the common ECDSA and RSA algorithms. Additionally, we measured RSA authentication with 1024 and 2048 Bit using two Aladdin eToken PRO 64k devices, which are essentially smart cards with USB connectors.

Figure 6 shows the average processing time, with 99% confidence intervals of 32 measurements. While for all, but the 4096-bit RSA, the software based algorithms terminated in under 0.4 seconds, the smart card authentication took at least 1.5 seconds.

Thus, we conduct that the generation of IKE signatures dominates the configuration of IPsec networks. In particular, these delays have a significant influence on the required time for the insertion of IPsec gateways, when many associations have to be established simultaneously.

8.3.2 Agility

Following the previous experiment, the IKE negotiation interval is expected to cause significant delays during the convergence, as multiple security associations have to be created when a new IPsec gateway is inserted into the system.

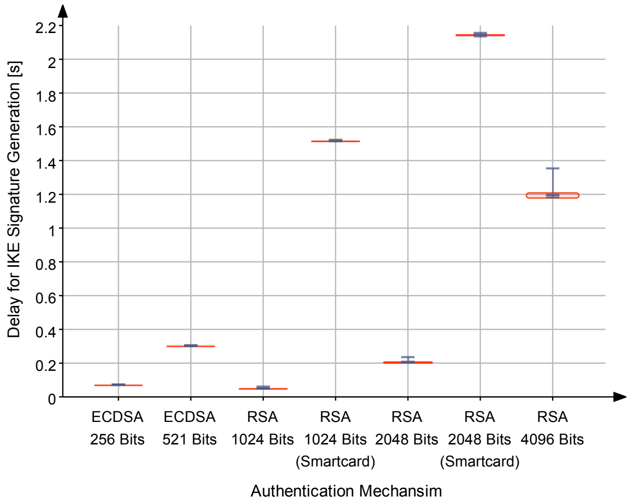

Fig. 6 Required time for IKE signature generation with 99% confidence (red) and min-max (whiskers)

Hence, in a first simulation experiment the IKE delay was varied and the time until an IPsec gateways was successfully inserted into a SOLID ring was measured. The VPN size was chosen to be an average of assumed sizes and consisted of 250 gateways, each representing a unique trusted network. All gateways were directly connected to a wide area network with propagation delay of 20 milliseconds between any two IPsec gateways, which is a reasonable for wired connections within a country. Periodic exchange requests were sent with a mean interval of 5 seconds.

Fig. 7 shows the results of 1,000 simulation runs with a confidence interval of 99%. The assumption of a congestion free network and delays of 20ms between IPsec gateways, results in a very quick convergence time of less than 1 second in absence of IKE signature delays. The rest of the delays follow a clearly linear trend: The IKE negotiation interval has an important influence on the convergence interval, and the time to convergence is proportionally dependent on this value.

SOLID needs 12 seconds to converge, when a single gateway is inserted and realistic delays for the IKE negotiation are assumed. This value might seem very high at first. However, every IPsec gateway will experience such a high delay on its own initial join to the VPN only. Subsequent join operations, due to mobility handovers or other integration events, can be handled quicker, as the respective neighboring IPsec gateways are known already.

Moreover, enhancements of the IKE, like e.g. MOBIKE [26], could help and circumvent a computation

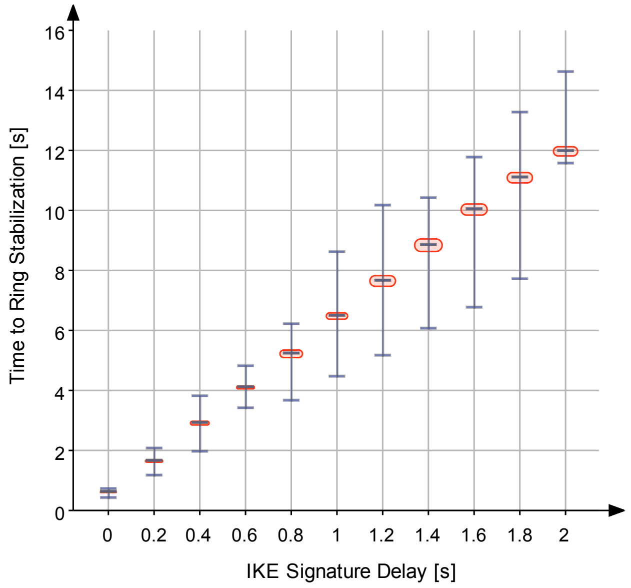

Fig. 7 Convergence time at gateway insertion, with 99% confidence (red) and min-max (whiskers)

intensive renegotiation in case of subnetwork relocation (changing public IP addresses), thus further reducing this initial delay.

For the rest of the experiments we simulated an IKE delay of 1.0 seconds, but the measured convergence delays scale linearly with differing negotiation delays.

8.3.3 Scalability

The time SOLID takes to converge when a new IPsec gateways is inserted is also an important factor with respect to its scalability. The system can only be considered scalable, if it is subject to at most a linear increase with regards to the number of configured networks, in the worst case. For reasons of SOLID's design, being based on a structured ring overlay with additional shortcut links, the time to convergence is expected to even be of a logarithmical order, only.

To evaluate this assumption another simulation experiment was conducted and the time to convergence after the insertion of a single new IPsec gateway into stable VPNs of growing size was measured.

The results of 1,000 simulation runs are given in Fig. 8, on a logarithmic scale. They back the expectation that SOLID is scalable, as the increase of the required interval can be fit by root-like function. Hence, the interval is within the assumed order of $\mathcal{O}(\log n)$ with a constant $c < 1$.

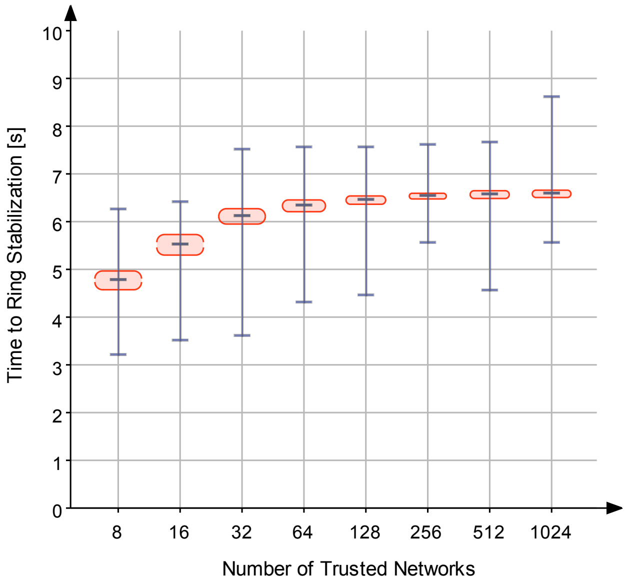

Fig. 8 Convergence after insertion of a single gateway, with 99% confidence (red) and min-max (whiskers)

8.3.4 Robustness

As noted in section 3, a key performance index of robustness is the required startup time of the system after a complete breakdown, such as a crash of an Internet exchange point, a restart after a critical software update, or a power failure. This event can be interpreted as a worst case correlated failure and thus gives the upper bound of the restoration time for such events. A simulative evaluation was conducted to evaluate the hypotheses that SOLID converges under the condition of a concurrent restart of all IPsec gateways and to identify major influence factors on the required convergence interval. The most significant factor in the scenario is the number of trusted networks. With rising numbers of participating IPsec gateways, the time to convergence for the VPN setup can be expected to show a logarithmic increase, as a logarithmic number of security associations must be setup.

Hence, in a simulation experiment the number of networks was varied and the time until all IPsec gateways had established stable associations to all neighbors was measured. Every IPsec gateway was started randomly within the first 10 seconds, and all other parameters were kept like in the last experiment.

The chart in Fig. 9 shows the average results of 1,000 simulation runs with a confidence interval of 99%. In contrast to the expectations, the increase in time is not logarithmic, but linear in the number of networks. Additionally, the min-max-interval of the measured conver-

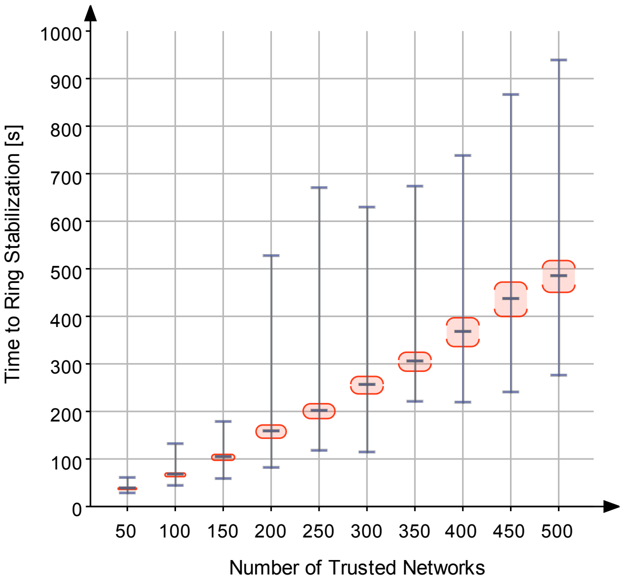

Fig. 9 Convergence after simultaneous start with 99% confidence (red) and min-max (whiskers)

gence suddenly drastically increases when the number of 200 simulated networks is reached.

This effect is caused by inconsistencies in the ring structure that occur due to the concurrent startup of all IPsec gateways, which are fixed by the periodic ring exchanges subsequently. During the process of restructuring, an unnecessary large amount of IPsec security associations is created in order to converge to the appropriate structure. When 200 or more networks are configured, this temporarily causes the queues of signature operations at the smart cards to grow to a size, in which some gateways due to the resulting delays start to observe timeouts. In consequence, the security gateways start retrying to establish associations, which results in the higher variance in the measured time.

SOLID still securely recovers from this extreme situation, and successfully restores the correct ring structure. With knowledge about the number of expected security gateways, the required time additionally can be reduced by adapting the update frequency on demand.

9 Conclusion and Future Work

Within this article a novel approach for automatic IPsec configuration has been presented. In difference to other IPsec configuration mechanisms, SOLID does not rely on a dedicated infrastructure and simply uses the public Internet infrastructure. The use of an overlay discovery and routing approach enables the system to meet extensive security, robustness, and scalability requirements. SOLID requires no manual configuration, except for a unique certificate for each IPsec gateway, and thus reduces the cost of IPsec deployment. SOLID additionally allows for the application of IPsec mechanisms even under quickly changing networking conditions that are to be expected when the VPN configuration has to account for not only static, but dynamic participants as well.

However, some issues for SOLID remain to be improved in further studies. A task that has not been addressed in the required depth is the secure and yet robust allocation of network addresses. We are currently developing an address allocation that allows for extension and reduction of address ranges, when new devices are added to, or removed from an IPsec gateway.

Furthermore, we expect to improve the performance of the discovery by adding location awareness to the overlay. In order to speed up the reintegration of security gateways, we plan to measure handover times in the presence of MOBIKE and routing strategies that reduce overhead on mobile gateways, like preferably using wired links to forward lookup traffic.

Another direction of research incorporates the creation of dynamic IPsec infrastructures under the focus of DoS resilience. The general methods presented in [27] can be adapted to the SOLID system in order to create more robust networks.

References

1. Z. Fu, S. F. Wu, Automatic generation of IPSec/VPN Security Policies In an Intra-Domain Environment, in: Proceedings of the 12th internation workshop on Distributed System Operation and Management (DSOM), 2001, pp. 279–290.
2. 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.
3. L. A. Sanchez, M. N. Condell, Security Policy Protocol, Expired Internet-Draft (2002).
   URL `http://tools.ietf.org/html/draft-ietf-ipsp-spp-01`
4. S. Fluhrer, Tunnel Endpoint Discovery, Expired Internet-Draft (2000).
   URL `http://tools.ietf.org/html/draft-fluhrer-ted-00`
5. S. Fluhrer, DETERMINING SECURE ENDPOINTS OF TUNNELS IN A NETWORK THAT USES INTERNET SECURITY PROTOCOL, United States Patent US 2007/7207063 B1 (2007).
6. T. Tran, Proactive Multicast-Based IPSEC Discovery Protocol and Multicast Extension, in: Proceedings of the IEEE Military Communications Conference (MILCOM, 2006.
7. 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`

8. S. Fluhrer, SYSTEM AND METHOD FOR PROTECTED SPOKE TO SPOKE COMMUNICATION USING AN UNPROTECTED COMPUTER NETWORK, United States Patent US 2007/0271451 A1 (2007).
9. Y. Bhaiji, Network Security Technologies and Solutions, 1st Edition, Cisco Press, 2008, Ch. Part III: Data Privacy.
10. M. Rossberg, G. Schaefer, Ciscos Group Encrypted Transport VPN – A sceptical analysis, in: Proceedings of D-A-CH security, German, 2009, pp. 351–360.
11. 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.
12. 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.
13. I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, H. Balakrishnan, Chord: A scalable peer-to-peer lookup service for internet applications, ACM SIGCOMM Computer Communication Review 31 (4) (2001) 149–160.
14. C. Cramer, K. Kutzner, T. Fuhrmann, Bootstrapping locality-aware P2P networks, in: Proceedings of 12th IEEE International Conference on Networks (ICON), 2004, pp. 357–361.
15. M. Knoll, A. Wacker, G. Schiele, T. Weis, Decentralized bootstrapping in pervasive applications, in: Proceedings of 5th IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom), 2007, pp. 589–592.
16. Z. Hu, B. Li, ZAL: Zero-Maintenance Address Allocation in Mobile Wireless Ad Hoc Networks, in: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS), 2005, pp. 103–112.
17. A. J. Mcauley, K. Manousakis, Self-configuring networks, in: Proceedings of the IEEE Military Communications Conference (MILCOM), 2000, pp. 315–319.
18. S. Kent, C. Lynn, K. Seo, Secure Border Gateway Protocol (S-BGP), IEEE Journal on Selected Areas in Communications 18 (4) (2000) 582–592.
19. J. Touch, L. Eggert, Y.-S. Wang, Use of IPsec Transport Mode for Dynamic Routing, IETF Request for Comments 3884 (Proposed standard) (2004).
URL `http://www.ietf.org/rfc/rfc3884.txt`
20. C. Cachin, A. Samar, Secure distributed DNS, in: Proceedings of Dependable Systems and Networks (DSN), 2004, pp. 423–432.
21. R. Gupta, A. Gavrilescu, J. L. Miller, G. A. Wheeler, Peer-to-peer name resolution protocol (PNRP) security infrastructure and method, United States Patent 7,051,102 (2001).
22. H. Schmidt, T. Guenkova-Luy, F. J. Hauck, A Decentral Architecture for SIP-based Multimedia Networks, in: KiVS, Informatik aktuell, Springer Press, 2007, pp. 63–74.
23. Y.-C. Hu, A. Perrig, D. B. Johnson, Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks, Wireless Networks 11 (1-2) (2005) 21–38.
24. A. Barbir, S. Murphy, Y. Yang, Generic Threats to Routing Protocols, IETF Request for Comments 4593 (Proposed standard) (2006).
URL `http://www.ietf.org/rfc/rfc4593.txt`
25. Y.-C. Hu, A. Perrig, D. B. Johnson, Packet leashes: a defense against wormhole attacks in wireless networks, in: Proceeedings of the 22nd Conference of the IEEE Computer and Communications (INFOCOM), Vol. 3, 2003, pp. 1976–1986.
26. P. Eronen, IKEv2 Mobility and Multihoming Protocol (MOBIKE), IETF Request for Comments 4555 (Proposed standard) (2006).
URL `http://www.ietf.org/rfc/rfc4555.txt`
27. 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.