3GPP TSG SA WG3 Security S3-000686
Meeting S3#16
Sophia Antipolis, 28-30 November, 2000

Source: Siemens AG

Title: SA negotiation protocol for the Z_A interface

Document for: Discussion and decision

Work item: Network domain security

Agenda item: tbd

Abstract

For the Release 5 core network key management architecture, two different methods are under discussion for negotiating IPsec or MAP security SAs over Z_A to protect security protocols over Z_C between different networks. In the Siemens contribtuion S3-z000021 to S3#a5bis, several disadvantages of the first method, which directly uses IKE (IETF RFC 2409) for negotiating core network SAs between two KAC entities, were identified. This contribution further elaborates on this. Several proposals how to advance the discussion about the SA negotiation protocol for the Z_A interface are made. Two examples are included that emphasize the requirement for flexible negotation of configuration parameters and policy.

1 Introduction

The internet key exchange protocol (IKE) allows to negotiate security associations for IPsec. Security associations (SAs) for other security protocols can be negotiated by defining a new domain of interpretation (DOI) for IKE. Within the 3GPP Rel'5 standards the two-tiered core network key management will offer third party SA negotiation, where the KACs use IKE over the Z_A interface for negotiating SAs that are required by network entities communicating securely over Z_C between different networks.

Two methods of how to use IKE for SA negotiation are currently under discussion:

- Within the first method IKE directly negotiates Z_C SAs for IPsec between the KACs, i.e. the SAs needed for IPSec over the Z_C interface are available as a result of a run of IKE over the Z_A interface between the KACs. MAP security SAs are negotiated by using IKE with the MAP security DOI (drafted in S3-z00018).
- The second method uses IKE to establish IPsec SAs used to protect communication over the Z_A interface between the KACs themselves. The use of IPsec over the Z_A interface subsequently protect a new protocol (to be defined by 3GPP) over the Z_A interface which is used to negotiate SAs for the Z_C interface .(see chapter 3).

At 3GPP S3#15bis we raised several technical issues regarding the first method in S3-z000021. The major problem is the limited support for exchanging configuration information within IKE, which appears to limit the flexibility to support dynamically adaptable network configurations through the two-tiered UMTS key management architecture.

By giving two examples, the second section of this contribution illustrates the need for exchanging configuration parameters between networks. The examples show that IKE is only sufficient to support simple scenarios, but does not in itself provide a mechanism to dynamically exchange more complex information during SA negotiation. In the third section we give an outline of the alternative SA negotiation mechanism. Chapter four continues the discussion of technical issues from S3-z000021.

2 Examples for configuration data exchange

In S3-z000021 we identified the minimal requirement that the KAC initiating an IKE exchange must be able to send at least one pair of IP addresses to the responding KAC. These are the IP address of the NE in the initiating network and the IP address of the NE in the responding KAC's network. This requirement shall be illustrated by the following example.

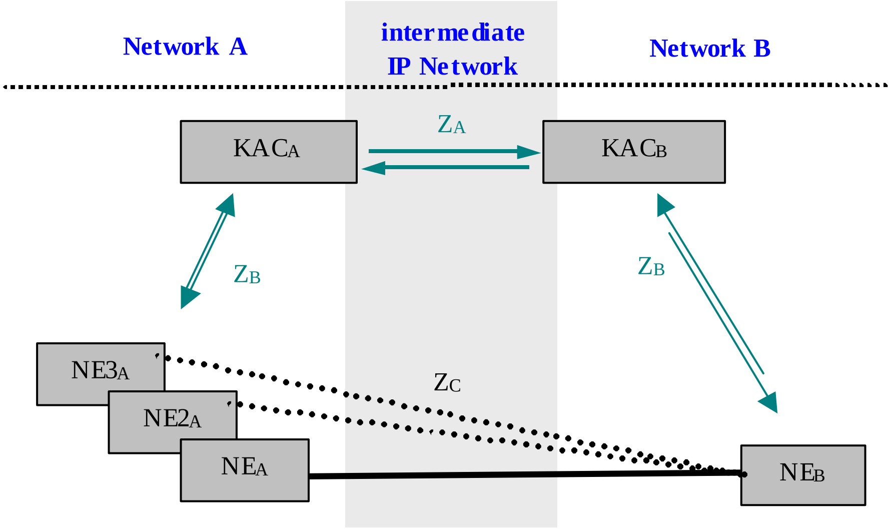


Figure 1: Key management for IPsec between two network entities

Assume NE_A in network A requires IPsec SAs with NE_B in network B. KAC_A therefore initiates the IKE SA negotiation with KAC_B. After the negotiation, KAC_A and KAC_B have agreed on a common pair of SAs. Now, without exchanging additional information during SA negotiation

- KAC_B cannot determine that the SAs must be sent to NE_B. Therefore the IP address of NE_B must be sent from KAC_A to KAC_B.
- network B cannot determine for which NE in network A the SAs shall be used. If there are several entities in network A that are configured to send IPsec protected packets to NE_B ($NE2_A$, $NE3_A$,...), NE_B must have the IP address of the peer in network A belonging to the SAs. Therefore the IP address of NE_A must be sent from KAC_A to KAC_B (and then to NE_B) as well.

As a conclusion, during an IKE quick mode (phase 2) exchange, at least two IP addresses must be sent from KAC_A to KAC_B.

IKE quick mode supports two optional ID payloads for exchanging additional identities. Updating S3-z000021 which described the exchange of a single ID payload per peer within IKE quick mode as being supported, it seems to be possible as well that the initiating IKE peer uses both payloads to send two IP addresses. Therefore this simple example should be supported by IKE.

But, in a second example below we illustrate that the capability to exchange only two IP addresses is unlikely to be sufficiently to provide key management for the core network IP security architecture.

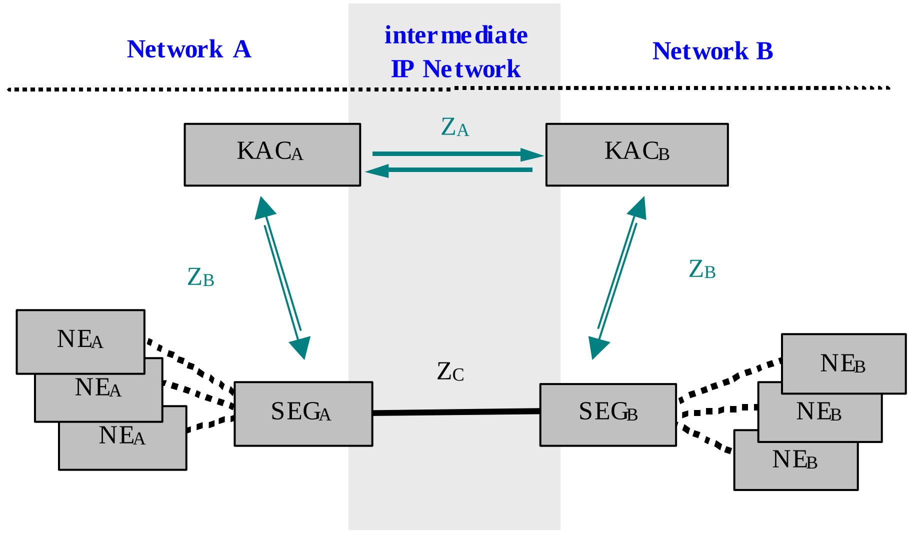


Figure 2: Key management for IPsec between two security gateways

Assume SEG_A in network A requires IPsec SAs with SEG_B in network B. KAC_A therefore initiates the IKE SA negotiation with KAC_B. In the above example we saw that the IP addresses of both SEG entities must be sent from KAC_A to KAC_B.

After SA negotiation, the SAs can only be used for a global IPsec tunnel (per port and protocol) between the SEGs through which all traffic is sent. When the SEG entities are supposed to support several IPsec tunnels each of them for communication between a specific NE_A/NE_B pair or sets of NEs in each network, KAC_A must send additional configuration parameters to distinguish between different tunnels, e.g. the NEs' IP addresses, to KAC_B.

Since scenarios like this are not unlikely, the exchange of only two IP addresses during IKE quick mode does not seem to be sufficient to provide IPsec SAs for the Z_C interface within the two-tiered architecture. The IP network structure that must be supported by the SA negotiation protocol will probably be even more complex than described in the two above examples.

To avoid the limitations of IKE, it would be possible to agree on a list of configuration profiles between two network operators by any out-of-band mechanism. A specific profile out of this list could then be selected during IKE SA negotiation by transmitting a pointer to this profile within quick mode. Although this seems to be a feasible approach, it would only allow for static configuration profiles. A change in the configuration of a single network would require changes in all related profiles of this network in all other networks using these profiles as well. Furthermore, it is not clear whether this would imply a certain misuse of IKE (see S3-z000021 for a more detailed discussion).

As a conclusion of this section, 3GPP requires an SA negotiation protocol for the Z_A interface that offers sufficient flexibility for exchanging complex configuration information in the Rel'5 network domain.

Before a final decision for a specific Z_A protocol can be made, the requirements for such a protocol must be clearly identified.

This especially means that the expected complexity of the supported scenarios should be known. If it is not known then the mechanism must be extensible to be able to accommodate unforeseen changes. The complexity translates into requirements on the support of the according parameters in the protocol exchanges. Parameters required to be supported include:

- Configuration parameters and policy information that must be exchangeable during SA negotiation over Z_A like IP addresses, ports, entity names, etc.
- Additional parameters, for example parameters required for the SA distribution mechanism of the core network key management architecture. An example could be a flag which indicates that new SAs must immediately replace the current ones if a key is compromised.

3 Alternative approach for SA negotiation over Z_A

We propose to further investigate the second method (as listed in chapter 1) for SA negotiation over Z_A. Major advantages of this approach are:

- Off-the-shelf IKE implementations can be used within this model.
- All 3GPP specific parts of the SA negotiation procedure are defined within 3GPP and can easily be extended in future releases.

The alternative approach is outlined as follows.

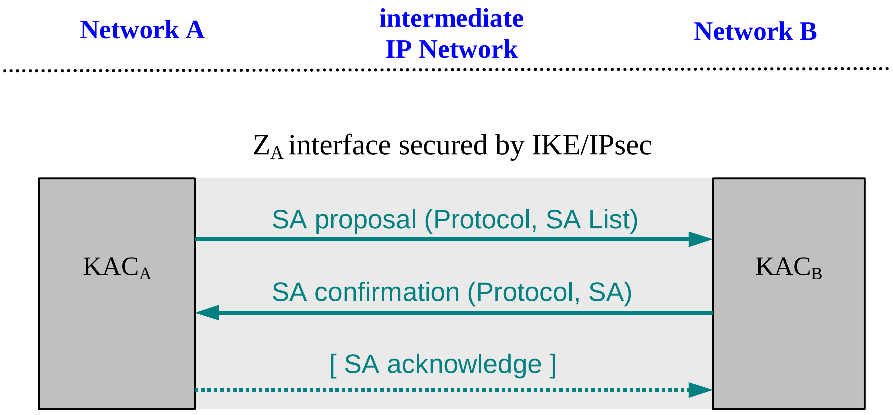
- IKE negotiates IPsec SAs between the KACs for use by the KACs. With these SAs a channel secured by IPsec is established between the KACs, which is integrity protected, encrypted and offers replay protection.
- This secure channel must be established only once (except key refresh) and is subsequently used to secure many runs ofan SA negotiation protocol which is still to be defined.

Such an SA negotiation protocol could consist of an exchange of two (or possibly three) messages:

1) The initiating KAC sends an *SA proposal* message.

2) The responding KAC selects one of the proposed SAs of the *SA proposal* message and sends an according *SA confirmation* message to the initiating KAC.

3) It is for further study whether an *SA acknowledge* message sent by the initiating KAC to the responding KAC is required.


Network A
intermediate IP Network
Network B
Z_A interface secured by IKE/IPsec
KAC_A
KAC_B
SA proposal (Protocol, SA List)
SA confirmation (Protocol, SA)
[SA acknowledge]

The parameters required in the *SA proposal* and *SA confirmation* messages need to be defined, as well as the SA proposal format for IPsec and MAP security. Another issue is the key agreement method which will be largely determined by the question whether joint key control is an issue or not.

The definition of such a protocol is expected to be feasible for Rel'5.

4 An alternative to be ruled out

The IPsec base protocols do not allow an entity to process incoming IPsec packets without a valid SA. If an SADB lookup using the SPI, destination address and IPsec protocol of the outer IP header does not return a valid SA or SA bundle, the packet must be dropped (see RFC 2401, section 5.2.1).

For the example in figure 1 one could imagine a solution that allows to exchange less or no configuration parameters over Z_A. With receiving the first IPsec packet from NE_A, NE_B could for example send the packet's SPI and IPsec protocol to KAC_B to get the corresponding SAs. Hence, the approach would be the following:

NE_A requests SAs with NE_B from KAC_A, which initiates IKE SA negotiation with KAC_B. This KAC now does not know the destination of the negotiated SA in network B and temporarily stores the SAs including the SPI. KAC_A distributes the SAs to NE_A, which in turn starts to send IPsec packets to NE_B. When receiving the first IPsec packet NE_B sends the SPI and IPsec protocol identifier to KAC_B which returns the already negotiated SAs. These SAs are added to the SADB of NE_B and are in place for communication.

A solution like this could probably be realized by a “shared” implementation, where the SADB is able to start an SA retrieval operation over Z_B, whenever valid SAs for incoming IPsec packets are not available. The KAC could return valid SAs or indicate that the packet must be dropped.

Although possible, this solution would introduce several new security problems:

- It is not clear how the parameters extracted from incoming IP headers, at least the SPI, can uniquely identify an SA stored in the KAC. For example, SPI and IPsec protocol are not sufficient to distinguish between different hosts or ports. When other parameters like the source IP address of the incoming IP packet are sent by NE_B to the KAC as additional identifier, again, these must sent over Z_A during SA negotiation. Otherwise the KAC could not identify SAs keyed with these additional parameters.
- If the KAC_B does not exactly know how to apply a negotiated SA, in principle every NE in network B can request this SA. The KAC_B cannot verify if the requesting NE is allowed to get a specific SA, including secret IPsec keys. Of course, NEs may not be assumed to attack their own KAC if they function correctly, but the possible points of attack are multiplied in this case. Also, a sending NE should have assurance that only the intended receiving NE can get hold of the SA.
- This approach shares a problem with the pure pull approach for the SA distribution protocol over Z_B: the NE cannot process the first protected packet before it received the SA from the KAC. Denial-of-service attacks are more likely within this approach: An attacker could easily send a large number of IP packets with different SPIs (and spoofed IP addresses if necessary) to a NE, which in turn must cache a possibly large amount of IP packets while initiating a large number of expensive SA retrieval operations over Z_B. The attack is likely to seriously affect the performance of the KAC_B and the NE_B. It seems difficult to protect NEs (including SEGs) from this kind of attack.

5 Conclusion

This contribution proposes the following:

- Before a final decision for one of the two discussed SA negotiation mechanisms can be made within 3GPP TSG SA3, the coomplete requirements for negotiating Z_C-SAs over the Z_A interface shall be identified. In particular, it is required to identify the parameters that must be exchanged between the KACs to agree on SAs, and to bind them to the correct policy. This requirement will result from an analysis of network configurations which need to be supported and the effects of dynamical changes of these network configurations.
- To define an SA negotiation mechanism for the Z_A interface that offers sufficient flexibility for exchanging configuration information in the Rel’5 network domain. The mechanism should be easily extensible by 3GPP.
- If it turns out that standard IKE procedures are not sufficient to satisfy the 3GPP requirements then 3GPP should define the mechanism needed in addition to the standard IKE procedures. No modifications of the standard IKE procedures are intended.
- The same protocol for negotating security associations over Z_A shall be used for all UTMS core network security protocols, including MAPSec as well as IPsec. The content of the negotiated SAs will differ, of course.

In addition to the above proposals and as result of the first method’s technical problems discussed in S3-z000021 and in this contribution, we propose to use the second method as SA3 working assumption. The feasibility of either method yet remains to be demonstrated because the protocol for the second method has not yet been specified.