CGA integration into IPsec/IKEv2 authentication

Conference Paper · November 2013

DOI: 10.1145/2523514.2527097

5 authors, including:

Ahmad Alsadeh
Birzeit University
12 PUBLICATIONS **57** CITATIONS
SEE PROFILE

Marian Gawron
Universität Potsdam
10 PUBLICATIONS **5** CITATIONS
SEE PROFILE

CGA Integration into IPsec/IKEv2 Authentication

Ahmad Alsa'deh
Birzeit University
ahmad.sadeh@gmail.com

Christoph Meinel
Hasso-Plattner-Institut
meinel@hpi.uni-potsdam.de

Florian Westphal
Hasso-Plattner-Institut

Marian Gawron
Hasso-Plattner-Institut

Björn Groneberg
Hasso-Plattner-Institut

ABSTRACT

In IPv6 networks, two security mechanisms are available at the network-layer; SEcure Neighbor Discovery (SEND) and IP security (IPsec). Although both provide authentication, neither subsumes the other; both SEND and IPsec mechanisms should be deployed together to protect IPv6 networks. However, when a node uses both SEND and IPsec, the authentication has to be done twice, which increases the burden on the node and decreases its performance. In this paper, we propose an approach to enable them to work together under the mediation of an *Authentication Management Block*, where IPsec uses the public-private keys obtained by SEND rather than negotiating its own authentication credentials in order to save the time and facilitate the IPsec authentication deployment. We implement and evaluate our approach using *ipsec-tools* and *DoCoMo SEND* implementations. Our proof-of-concept experiment shows a considerable speedup of IPsec authentication time.

Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and Protection—*authentication*

Keywords

IPv6 security, Authentication mechanism, IKEv2, CGA

General Terms

Experimentation, Performance

1. INTRODUCTION

IP security (IPsec) is a suite of protocols that secure data communication at network-layer [12]. It provides packet-level authentication, data integrity, and data confidentiality. According to RFC 6434, IPsec should be supported by all IPv6 nodes [8]. However, IPsec is not appropriate for securing IPv6 Neighbor Discovery Protocol (NDP) [16], [17]. A chicken-and-egg problem [2] appears when trying to use Internet Key Exchange (IKE) [9] for NDP operations. Using automatic IKE requires that the nodes have addresses before. On the other hand, IPv6 nodes use NDP to configure their addresses automatically at the initial stage when they join a new subnet. Therefore, the security must be considered at the initial stage when the host gets its address. As a result, The Internet Engineering Task Force (IETF) decided to abandon IPsec for securing NDP and design the SEcure Neighbor Discovery (SEND) [3] which is an integral part to the NDP. However, SEND is a local link security approach and cannot protect the connections outside its scope.

Accordingly, in IPv6 environment both SEND and IPsec should be used together for protecting network-layer. SEND complements the weakness of NDP within the local network at the initial stage and IPsec is used to protect IP packets for Internet communication. Therefore, hosts are first authenticated with SEND and are then authenticated again with IPsec. The repeated authentications increase the burden on the host and decreases its performance, in particular on resource-constrained devices, such as mobile phones.

Our aim is to avoid this repeated authentication step in IPv6 environments by sharing the authentication information between SEND and IPsec. The shared authentication information is stored in authentication database. The Cryptographically Generated Address (CGA) [5] of the host, whose authentication is completed by SEND, is stored in this shared authentication database. IPsec checks the shared authentication database before it carries out the authentication step. If the authenticated credentials (public-private keys) obtained by SEND are stored in the shared database, IPsec does not negotiate its own authentication information.

Using CGA for IPsec authentication reduces the hurdles of IPsec authentication configuration mainly in the absence of a global trust authority. CGA is a standalone authentication which does not need a third party where the control remain at endpoint owner. Every node can generate it own public private key pairs. This approach automates the authentication process. So, no need for pre-deployed information that have been authenticated by trusted authorities that cause significant time and effort to setup security associations. Moreover, this approach is consistent with RFC3723 which forbids manual keying [1].

The rest of the paper is structured as follows. Section 2

SIN'13, November 26-28, 2013, Aksaray, Turkey.

provides the basic background about IPsec and SEND. Section 3 describes key related works. Section 4 describes our proposed scheme. Section 5 evaluates the performance and the security implication of our approach. We conclude our work in Section 6.

2. BACKGROUND

In this section, we introduce IPsec suite with the focus the authentication part of establishing security associations in Internet Key Exchange (IKE). This is followed with a brief description of SEND.

2.1 IPsec/IKEv2

The Internet Key Exchange (IKEv2) [9] is the protocol responsible for the establishment of IPsec security associations (SAs), achieving the mutual authentication and selecting cryptographic keys. It does an automatic configuration for the security protocols (Authentication Header (AH) [10] or Encapsulating Security Payload (ESP) [11]), manages the entries in the Security Association Database (SAD) and ensures the mutual authentication of the hosts while initializing the IPsec connection.

IKEv2 performs mutual authentication between initiator and responder by establishing security associations (SAs) in two phases. Phase 1 establishes SA that carries IKE messages between the peers. Phase 2 establishes other SAs to carry the protected IPsec traffic. The SAs that carries IKEv2 messages is denoted as *IKE_SA*, and the SAs for IPsec is denoted as *CHILD_SA*. All communications comprise pairs of a request and response messages (IKEv2 exchange). In IKEv2, Phase 1 has two steps. Step 1 *(IKE_SA_INIT)* negotiates security parameters to create *IKE_SA*, computes secret keys for IKE, and computes master secret for computing IPsec keys in Phase 2. The initiator sends its supported Diffie-Hellman key exchange and the *nonce* value to the responder in *IKE_SA_INIT* request. The responder includes its acceptable cryptographic algorithm, its own part of the Diffie-Hellman key and *nonce* and returns it back to the initiator in *IKE_SA_INIT* response. The attributes of the IKE SA can be exchanged during *IKE_SA_INIT* exchange, i.e., the message authentication algorithm, and encryption algorithm, and Diffie-Helman group information. After completing the *IKE_SA_INIT* exchange successfully, both peers can independently calculate the keying information that is used to authenticate the IKE peers, to authenticate and encrypt messages, and to derive keys that are established for child SAs.

Phase 1, step 2 *(IKE_AUTH)* completes activation of IKE SA and sets up an SA for Phase 2 (first child SA). The initiator sends an *IKE_AUTH* request that includes its identity and authentication information. The authentication information depends on the initiator's authentication mechanism that was defined in *IKE_SA_INIT* exchange. The most common IKE authentication methods are the pre-shared-key (PSK) and certificate-based (X.509 certificate). For PSK authentication, the communicating peer proves its identity through the knowledge of the PSK and with giving a correctly encrypted response. For certificate-based authentication, the initiator includes the certificate it used to create its signature in *IKE_AUTH* request. In the *IKE_AUTH* response, the responder includes its certificate which has to be signed by the responders trusted authority and prove that node has the private key which corresponds to the public key in the certificate. Besides that, the initiators send a list of their trust anchors. The responders send their signed certificate and a proof for the ownership of the private key for the certificate to prove their identities.

Phase 2 *(CREATE_CHILD_SA)* setups AH or ESP security associations (SAs). For IKEv2, the initiator sends a *CREATE_CHILD_SA* request that contains some proposals for parameters, such as session key. The responder answers with a *CREATE_CHILD_SA* response that contains the acceptable parameters. These parameters are stored in the SAD and will be used for ensuring confidentiality and integrity of the messages which are protected by IPsec. The attributes that can be negotiated in this phase include the protocol (AH or ESP), the authentication algorithm, encapsulation mode (tunnel or transport), encryption algorithm and Diffie-Helman group information.

However, IPsec is not applicable for securing NDP and the local link communications [4]. IPsec uses IKE that requires a valid IPv6 address, while this cannot be done during the initial phase of IPv6 stateless address autoconfiguration. Therefore, the SEcure Neighbor Discovery (SEND) has been designed as an extension to NDP to offer the authentication at the initial stage for generating a secure IPv6 address.

2.2 SEcure Neighbor Discovery (SEND)

SEND has been developed to secure Neighbor Discovery Protocol (NDP). SEND is a set of enhancements to the NDP messages which mainly depends on Cryptographically Generated Addresses (CGA) [5] and X.509 certificates [15]. SEND participates in authentication at the initial stage for generating a secure IPv6 address. CGA for IPv6 enables the nodes to generate their own secure address and verify the ones from others without relying on any global trust authority, i.e., the node's capability to prove it has the address it claims to have. In CGA, the interface identifier of IPv6 address is created from truncated SHA-1 hash of the address owner's public key and other auxiliary parameters. The node sends its CGA parameters and a signed message from the CGA. The receiver recomputed the hash of the sender CGA parameters according to CGA standard [5], compares it with interface identifier of the source address, and verify the signature using the *Public Key*. Thus, the receiver knows that this address is bonded to this public key.

Therefore, each SEND and IPsec perform a separate authentication for the same node in IPv6 environments. The duplicated authentication on the same host increases the processing cost and consumes the computing device energy. Therefore, the duplicate authentication might be redundant by sharing the authentication information between SEND and IPsec. Several approaches have been proposed to achieve this goal as we show in the following section.

3. RELATED WORK

The most related approach to ours has been presented by Kim et al. [13], where they propose a cooperation authentication of IPsec and SEND. Our approach is in fact an enhanced version of their approach because their approach has some limitations; They store only CGAs in a reposi-

tory and IPsec skips the authentication for the existing IP address in this repository. Skipping the authentication negotiation in IPsec makes it vulnerable to man in the middle attacks. Also, the collaboration between SEND and IPsec, in their approach, is not always feasible for all network environments. The collaboration system assumes that both SEND and IPsec are running simultaneously on particular nodes located at the same subnet in order to benefit from the SEND authentication information and skip the IPsec authentication procedure. The reason is that the host authentication information cannot be shared with the other host in a different network because the SEND operates inside the local network only. Moreover, they used asynchronous communication for the interaction between IPsec and SEND. This makes it necessary for the IPsec to wait for one second after each request to make sure that the repository process had enough time to handle the request and answer accordingly. Because of this necessary waiting, the processing time for the authentication step of the IKE protocol is always at least one second, which is considerably longer than the time needed for the regular authentication step.

There are other several approaches whose goal is to integrate CGA and IKE. For instance, Castelluccia et al. propose an opportunistic encryption scheme based on cryptographic identifier (Crypto-Based Identifiers) for establishing IPsec tunnels between specific security gateways [6]. An IETF draft [14] follows this work and proposes using IKE with IPv6 CGA. However, this draft expired on January 9, 2008. Combes et al. follow the design choice of this IETF draft [14] to use CGA as alternative security credentials with IKEv2 [7]. They included the CGA parameters in the IKEv2 payload and consider it as a new certificate type.

4. SEND AND IPSEC COMBINED AUTHENTICATION METHOD

The idea of our approach is to use the CGA as an authentication mechanism for IPsec. The CGAs of all communication nodes which are authenticated by SEND represent the mutual authentication information between SEND and IPsec. If a host uses CGA authentication, then *IKE_AUTH* exchange in IKE of IPsec can use CGA public key. This combined authentication approach uses a shared database between SEND and IPsec for exchanging the authentication information.

Fig. 1 shows the general architecture of this approach and the interaction between the different components. The main components are the *SEND Block*, the *IPsec Block*, and the *Authentication Management Block*. The *SEND Block* stands for the SEND implementation whereas the *IPsec Block* represents the IPsec protocol implementation. The *Authentication Management Block* manages the interaction between *SEND Block* and *IPsec Block*.

The *SEND Block* does the CGA generation (IP address authentication), handles the packets related to ND messages, and adds the authenticated CGAs and parameters to the *IP Database*. The *SEND Block* verifies the source CGA of an arriving SEND packet and adds the validated address to the database if it is not already stored there.

The *IPsec Block* secures the other IP packets rather than ND

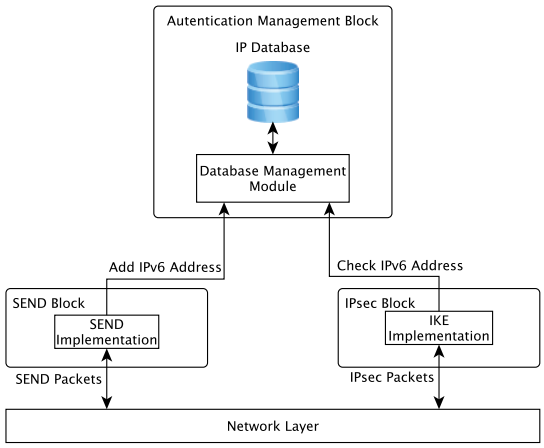

Figure 1: General architecture for SEND and IPsec collaboration.

packets. *IPsec Block* sends a check signal to the *Database Management Module* to check if the source IP address can be found in the *IP Database*. If yes, *IPsec Block* skips the normal authentication during *IKE_AUTH* of the IKE key exchange and uses the CGA public keys. The *IPsec Block* processes all IPsec packets and verifies if the connection partner can be authenticated via its IP by iterating over the stored IP addresses to determine if the provided IP address has been authenticated by *SEND Block*.

The *Authentication Management Block* carries out the function of SEND and IPsec packets analysis and authentication management. Once a CGA is successfully generated, the *Authentication Management Block* stores the authentication information in the *IP Database*. This database contains the CGAs and parameters of the nodes whose authentication is completed by SEND. The *Authentication Management Block* answers the check signal of *IPsec Block* whether the address already exists or not.

The communication between *SEND Block* and *IPsec Block* is implemented in a way such that both blocks use the provided interface to access the database. Fig. 2 and 3 show a typical interaction of those blocks with the *Authentication Management Block*.

To achieve the combination between SEND and IPsec, we modified IKEv2 payload to use SEND keys. However, our modifications do not change the existing IPsec architecture. In IKEv2, the Identification (ID) payload contains the peer's identity to be authenticated with the authentication (AUTH) payload. CGA is set within the ID payload under the IKEv2 identity type ID_IPV6_ADDR. The certificate payload (CERT) provides a means to transport certificates or other authentication related information via IKE. In case of using SEND credentials for IPsec, the CERT payload is used to transmit the CGA parameters. The Certificate Request (CERTREQ) payload determines what certificates the initiator is willing to use. For CGA, we need new type of CERTREQ payload. The Authentication payload (AUTH) contains data used for authentication purposes. For CGA, it contains a digital sig-

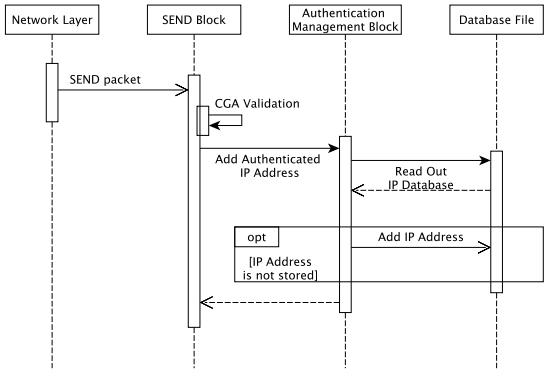

Figure 2: Sequence diagram of for handling the received ND message in a combined SEND and IPsec authentication.

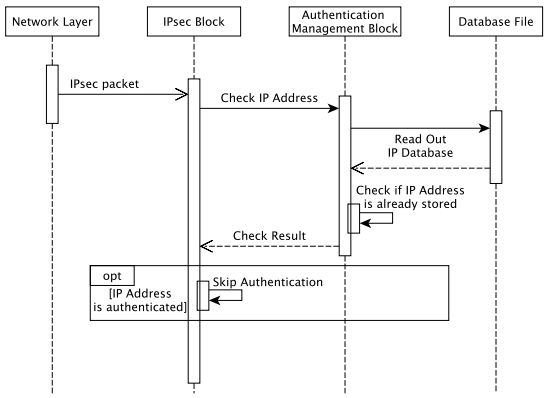

Figure 3: Sequence diagram for handling the received IPsec packet in a combined SEND and IPsec authentication.

nature of the message computed using the corresponding public key in CGA parameters.

5. EXPERIMENTS AND EVALUATION

In this section, we present the evaluation of our approach with respect to the deployment considerations, performance and security implications.

In order to implement the collaboration between SEND and IPsec, we adapt both the SEND and the IPsec implementations accordingly. We use the open source *DoCoMo SEND*[1] and *ipsec-tools*[2] software to test the necessary extension of the collaboration module between SEND and IPsec. In DoCoMo SEND, we adapt the method of handling the incoming ND messages by changing the method *handle_incoming()* in the file *proto.c*. This change adds the respective code necessary to add the verified CGA to the *IP Database*. For *ipsec-tools*, we add the communication code, which is necessary to check if the received CGA is already stored in the *IP Database*. We also do the modification code responsible for skipping the original authentication if possible to the method *oakley* validate *auth()* in the IPsec daemon *racoon* source file *oakley.c*.

The communication between the three described blocks is done in a synchronous way. This synchronous communication enables the *IPsec Block* to directly decide if the CGA authentication can be used, without waiting for a reply from another process. Our implementation keeps the *IP Database* duplicate-free, due to the insert mechanism used.

5.1 Performance Evaluation

During our experiments, we use a testbed which consists of two machines and one router, which are all located in the same subnet. Whereas the router advertises the prefix used for global IP addresses, the other two host machines use the respective implementations to create an encrypted IPsec connection between hosts, according to our modified approach. All three machines in this scenario runn FreeBSD 9.0[3] as operating system, *DoCoMo SEND*[4] version 0.3 and *ipsec-tools* version 0.8.0.

Table 1 shows the measurement results of the authentication time of the regular IPsec/IKEv2 implementation and our modified implementation. The measurements are taken for two authentication methods: the pre-shared key authentication (PSK) and the certificate-based authentication using X.509 certificates. The measurement results are shown in milliseconds (ms). Our implementation performs 1.5 to 1.7 times faster than the regular IPsec/IKEv2 authentication.

Table 1: IPsec authentication time measurement results in milliseconds (ms).

Implementation	Authentication Methods	
CGA IKEv2	0.80 (CGA)	0.28 (CGA)
Regular IKEv2	1.38 (X.509)	0.43 (PSK)

5.2 Security Implications

We believe that using CGA for authentication of IPsec does not lead to any recognizable security threats. If a malicious node spoofs a valid CGA and tries to use it for communicating with other nodes, it will not succeed, because the malicious node needs to sign the outgoing packets with the private key corresponding to spoofed address. On the other side, the receiver node will verify the CGA and the signature and discards the packet if the verification fails. Moreover, the spoofing attack is not possible if IPsec uses the pre-shared-key authentication, because the malicious node is unable to decrypt the initial packets where the session key is created. Since CGA does not depend on any trust authority, a malicious node can generate its own CGA and start communicating with other nodes. Proof of the authority is out of the scope of CGA. At least using CGA prevents the theft of other nodes' addresses. Therefore, we do not consider it a drawback of CGA or our proposal. On the contrary it can be an advantage of our approach, because

[1]http://www.aestheticscientist.com/082406/lab_opensource.html
[2]http://ipsec-tools.sourceforge.net
[3]http://www.freebsd.org/where.html
[4]http://people.freebsd.org/~ehaupt/distilator/net-mgmt/send/

the entire communication process is secured for all steps. CGA is used to generate a secure IPv6 address at the initial stage and this secure address is used for establishing secure IPsec connection.

5.3 Deployment Considerations

In practice, there are three different scenarios where SEND and IPsec can be used simultaneously: When the communicating hosts are on the same subnet, when the hosts communicate with each other through a security gateway, and when the communicating hosts belong to different networks. Our approach can be used for all these scenarios. If SEND authentication is successful, a CGA assigned to the host can be stored in the authentication database with its corresponding CGA parameters. Then, *IPsec Block* can check *IP Database* and use these authentication information for its own authentication.

In IPv6 networks, the IP address can be frequently changed in order to protect the user's privacy or to achieve the movement of the host. The changeable addresses may affect the IPsec connection. If one host changes its address, the established IPsec connection will fail. Accordingly, the collaboration system should be able to maintain IPsec connection for changeable addresses. There are two possible solutions for this problem. First, when a host changes its address, it sends the existing session information and security key information concerning the newly assigned address to the other host. Since the node that changed its IP address is the only one that could know the common secret it could prove its identity by this. Therefore, there is a need to create a mechanism where a new connection could be established with an already existing common secret which has to be saved and which could be associated with a previous interaction. Second solution is to continue using the existing IP address for the already established connections for which it is not possible to use the new address without interrupting the current application. However, the old address should be treated as a deprecated address and should not be used for new connections and should be deleted once the already established connections are finished. Using the deprecated address does not require sending any secrets or re-authenticating because the connection is not aborted and the other side is still able to communicate to the former address. Therefore, we recommend the use of the second approach because there is lower management cost and the risk of interrupting the running applications is also lower.

6. CONCLUSION

This paper proposed a collaboration authentication method between SEND and IPsec in an IPv6 environment. The collaboration system includes three main blocks; *SEND Block*, *IPsec Block*, and *Database Authentication Block*. The *SEND Block* authenticates the hosts' address and stores the authentication information corresponding to the hosts in the *IP database*. The *IPsec Block* checks whether the IP address is present in the *IP Database* or not. If the authentication information is valid, IPsec can use these authentication credentials inside of exchange its own. The *IP Database* can be updated when the host generates a new address. The new address should be used for all connections when it possible. The evaluation results show a significant performance advantage of using this approach compared to the conventional system where SEND and IPsec deployed separately. We additionally discuss the feasibility of using this system in different network environments and discuss the security implications involved.

7. REFERENCES

[1] B. Aboba, J. Tseng, J. Walker, V. Rangan, and F. Travostino. Securing block storage protocols over ip. RFC 3723, April 2004.

[2] J. Arkko. Effects of icmpv6 on ike, March 2003. Expired.

[3] J. Arkko, J. Kempf, B. Zill, and P. Nikander. Secure neighbor discovery (send). RFC3971, March 2005. Updated by: 6494, 6495.

[4] J. Arkko and P. Nikander. Limitations of ipsec policy mechanisms. In *Proceedings of the 11th international conference on Security Protocols*, pages 241–251, Berlin, Heidelberg, 2005. Springer-Verlag.

[5] T. Aura. Cryptographically generated addresses (cga). RFC3972, March 2005. Updated by: 4581, 4982.

[6] C. Castelluccia, G. Montenegro, J. Laganier, and C. Neumann. Hindering eavesdropping via ipv6 opportunistic encryption. In P. S. et al., editor, *The European Symposium on Research in Computer Security (ESORICS 2004)*, volume 3193, pages 309–321. Springer-Verlag, 2004.

[7] J.-M. Combes, A. Wailly, and M. Laurent. Cga as alternative security credentials with ikev2: implementation and analysis. In *SAR-SSI 2012, 7th Conference on Network Architectures and Information Systems Security*, Cabourg, France, May 2012 2012.

[8] E. Jankiewicz, J. Loughney, and T. Narten. Ipv6 node requirements. RFC 6434, December 2011.

[9] C. Kaufman, P. Hoffman, Y. Nir, and P. Eronen. Internet key exchange protocol version 2 (ikev2). RFC5996, September 2010. Updated by: 5998.

[10] S. Kent. Ip authentication header. RFC 4302, December 2005.

[11] S. Kent. Ip encapsulating security payload (esp). RFC 4303, December 2005.

[12] S. Kent and K. Seo. Security architecture for the internet protocol. RFC 4301, December 2005. Updated by: 6040.

[13] T. Kim, I. Kim, Z. Zhen, J. H. Kim, G. Gyeong, and Y. I. Eom. A cooperative authentication of ipsec and send mechanisms in ipv6 environments. In *Proceedings of the 2008 International Conference on Advanced Language Processing and Web Information Technology*, ALPIT '08, pages 418–423, Washington, DC, USA, 2008. IEEE Computer Society.

[14] J. Laganier, G. Montenegro, and A. Kukec. Using ike with ipv6 cryptographically generated addresses. draft-laganier-ike-ipv6-cga-02, July 2007.

[15] C. Lynn, S. Kent, and K. Seo. X.509 extensions for ip addresses and as identifiers. RFC3779, June 2004. Updated by: 5998.

[16] W. A. S. Thomas Narten, Erik Nordmark and H. Soliman. Neighbor discovery for ip version 6 (ipv6). RFC 4861, September 2007. Updated by: 5942.

[17] S. Thomson, T. Narten, and T. Jinmei. Ipv6 stateless address autoconfiguration. RFC 4862, September 2007.