Network Working Group J. Laganier
Internet-Draft DoCoMo Euro-Labs
Expires: January 9, 2008 G. Montenegro
Microsoft Corporation
A. Kukec
University of Zagreb
July 8, 2007

Using IKE with IPv6 Cryptographically Generated Addresses
draft-laganier-ike-ipv6-cga-02

Status of this Memo

This Internet-Draft will expire on January 9, 2008.

Copyright Notice

Abstract

This document describes IKEv2 peer authentication via IPv6 Cryptographically Generated Addresses (CGA). This techincque have been proposed to solve several security issues in the absence of any centralized trusted security infrastructure and without any pre-arrangements, to provide IPSec self-evident authentication mode between IPv6 nodes or security gateways.

Table of Contents

1. Introduction and Problem Statement

This document describes how to use IKEv2 with IPv6 Cryptographically Generated Address (CGA). The use of CGA provides authentication for IKEv2 based on the address-proof-of-ownership. It requires only slight modification of IKEv2 protocol and provides self-evident authentication between previously unknown IPv6 nodes and/or gateways.

CGAs have been proposed to solve several security issues in the absence of any centralized trusted security infrastructure and without any pre-arrangements, for example, securing Binding Updates in Mobile IPv6, securing Neighbor Discovery for IPv6, securing IPv6 multihoming protocol - SHIM6, and securing Group Membership in Multicast and Anycast communications.

Until now, the deployment of IPSec has been severely limited because it constrains IP nodes to have either a pre-shared key or a common trusted root (e.g., a PKI infrastructure). Thus, the lack of a global, Internet-wide, trusted infrastructure has precluded a straightforward application of IPsec between any two previously unknown nodes. CGA provides an alternative for Security Association establishment that does not require using a centralized infrastructure or prearrangements. This approach is similar to [4].

From the point of view of implementation effort, the fact that this approach only requires the addition of stand-alone CGA validation routines into existing IKEv2 daemons (e.g. racoon2, pluto, ikev2, etc) is another considerable advantage.

This note is organized as follows: we will first describe related work and some usage scenarios for a CGA-enabled peer authentication within an IKEv2 exchange, then we will enumerate requirements for those IKEv2 payloads that need it, and finally, describe precisely how to perform the IKE_AUTH exchange. Accordingly, this note presents an overview of how to use the Internet Key Exchange version 2 protocol [1] while one or both peers authenticate themselves via CGA proof-of-ownership. This document details the slight modifications needed. Additionally, it aims at capturing the current thinking about how to achieve proof-of-ownership in IKEv2 via CGA in a standard manner, thus preventing subsequent conflicting definitions. Currently, it has still some open aspects/issues.

2. Terminology

Cryptographically Generated Address (CGA): An adress which is generated in accordance with [2]. Any crypto address used must be generated as a CGA.

CGA enabled Node: An IPv6 node that has one CGA configured as its IPv6 address and posseses related CGA Parameters structure, as well as the private key. For such nodes we use terms initiator and responder. CGA enabled Nodes are present both in the transport scenario and tunnel scenario.

CGA enabled Security Gateway: An IPv6 enabled IPsec security gateway which applies tunnel mode CGA Security Policy (CGA SP), either in the net to net scenario or in the endpoint to net scenario. It posseses CGA and related CGA Paramateres data structure. We use mark GW_i for the initiator's CGA enabled Security Gateway and GW_r for the responder's CGA enabled Security Gateway.

Self-evident authentication mode: It denotes the use of IPSec/IKEv2 between previously unknown CGA enabled nodes and/or CGA enabled gateways. The IKEv2 Security Association establishment in case of self-evident authentication mode of IPSec does not require any pre-arrangements nor trusted security infrastructure.

CGA Security Policy (CGA SP): A CGA Security Policy is a Security Policy that specifies that traffic towards any outbound destination (i.e., ::0/0) SHOULD be protected by IPsec, either transport mode or tunnel mode (transport mode for securing end-to-end traffic between two nodes and tunnel mode for securing traffic between two CGA enabled Security Gateways, while in transit in the public internet).

3. Node Configuration and Requirements

Each CGA enabled Node and CGA enabled Security Gateway MUST prove address ownership of its CGA. CGA has to be generated and verified against CGA Parameters data structure as described in [2]. The CGA authentication procedure includes the following steps:

1. The verification of binding between the CGA and the proposed public key/CGA Parameters data structure. Peer has to verify: a) the produced hash of the CGA Parameters data structure, b) the prefix of the CGA. The lower 64 bits of the hash MUST match the iid of the CGA. Also, the prefix contained in the CGA Parameters data structure MUST be the same as the prefix in the CGA.

2. The verification that the peer's private key corresponds to the public key from the CGA Parameters data structure. For example, the responder MUST verify the challenge signed with the initiator's private key with the initiator's public key received in the CGA Parameters data structure.

Succesfull verification denotes that the public key in the CGA parameters is the authentic public key used to generate CGA. Both CGA enabled Nodes and CGA enabled Security Gateways use their CGA as IKEv2 identities.

Thus, each CGA enabled Node and Security Gateway generate a RSA public (PK) - private (SK) key pair in accordance with Internet X.509 certificate profile (rfc3280). Usually, keys will be generated together with the self-signed X.509 certificate.

The following requirements are related to tunnel mode scenarios and are actually open issues:

> Beside the check of the PK-CGA binding of peer GW's CGA, each CGA enabled Security Gateway SHOULD verify also the initiator's and responder's PK-CGA binding. In that case, gateways need to exchange CGA Parameters related to initiator's and responder's CGA. The PAD database could be used to store the initiator's/ responder's CGA Parameters (TBD). Those CGA Parameters could be exchanged in the IKE_AUTH exchange (TBD).
>
> Beside the address-proof-of ownership, gateways (GW_i and GW_r) SHOULD mutually prove themselves that they had been authorized to act as CGA enabled Security Gateways for the initiator's or responder's CGA (respectively). This could be solved with the authorization certificates, eg. initiator could issue certificate to GW_i to allow GW_i to be its CGA enabled Security Gateway. However, authorization certificates SHOULD not be used for this

purpose because of the incompliance with the IKEv2 spec and CGA spec. Both specifications require use of the X.509 Certificate - Signature. This issue SHOULD be rather solved with making use of PAD database to store additional authentication and authorization parameters (TBD).

4. Usage Scenarios

CGA authentication within an IKEv2 exchange can be applied in several different usage scenarios. The following sections describe some of these scenarios while emphasizing on easiness of Security Policy configuration.

Self-evident authentication mode for IPSec bootstraps an IPsec Security Association between two previously unknown nodes. Some schemes have been proposed to achieve this goal: FreeS/WAN Opportunistic IPsec uses the standard IKE protocol and DNS queries to retrieve IKE peers' public keys. While these schemes certainly allow to bootstrap such an SA, we argue that it is not convenient to rely on upper layer infrastructure (e.g., DNS) to secure the network layer. This causes cyclic dependencies that ends up in a chicken-and-egg problem: DNS is carried over {TCP|UDP}/IP and a related Security Policy should require that this traffic be protected as well, thus requiring Opportunistic negotiation to secure needed KEY RR lookups. On the other hand, a CGA-based scheme achieves true independence because the security gateways can discover each other and verify authorization by relying solely on IP infrastructure.

While the supported IKEv2 identites are IPv4 addresses, IPv6 addresses, FQDN, email addresses, X.500 Distinguished Name, X.500 General Name and vendor-specific identity, the self-evident authentication mode of IPSec provides the possibility to identify with IPv6 address or FQDN. In case of FQDN identity, the DNBS would provide the binding between the FQDN and the IPv6 address and then the CGA would provide the binding between the IPv6 address and the crypto material.

In the rest of the section, we describe the following three application scenarios: transport mode, tunnel mode gateway to gateway and tunnel mode node to gateway.

4.1. IPsec Transport Mode with self-evident authentication

IPSec Transport Mode with the self-evident authentication secures end-to-end communications between any two previously unknown CGA enabled Nodes and thus provides any-to-any trust relationships. For instance, let's assume that initiator initially wants to send a data packet to responder. Transport Mode CGA SP requires protection of this data packet. As no trust relationship exists between initiator and responder prior to this, they needs to establish a Transport Mode IPsec Security Association.

```
[initiator]<=i=p=s=e=c==t=r=a=n=s=p=o=r=t=>[responder]
```

Bootstrapping an IPsec SA between two CGA enabled Nodes is straightforward: the two peers merely prove ownership of their CGA's while performing the IKEv2 exchange, and configure negotiated IPsec SA's.

A typical Transport Mode CGA SP policy should look like that:

```
INBOUND:

::0/0[ike] -> cga_addr/128[any] udp bypass

::0/0[any] -> cga_addr/128[ike] udp bypass

::0/0[any] -> cga_addr/128[any] any require (ipsec/ah/esp/
transport)

OUTBOUND:

cga_addr/128[any] -> ::0/0[ike] udp bypass

cga_addr/128[ike] -> ::0/0[any] udp bypass

cga_addr/128[any] -> ::0/0[any] any require (ipsec/ah/esp/
transport)
```

4.2. IPSec Tunnel Mode with self-evident authentication

IPSec Tunnel Mode with self-evident authentication is used to secure communications between two CGA enabled nodes (initiator and responder), while this traffic is in transit between their gateways (GW_i and GW_r).

```
[initiator]<---[GW_i]<=i=p=s=e=c==t=u=n=n=e=l=>[GW_r]--->[responder]
```

Bootstrapping a tunnel mode IPsec SA between two CGA enabled nodes is not as straightforward as it is for transport mode, because (1) the GW_r needs to be discovered by the GW_i, and (2) both GW_i and GW_r need to be authorized by the source and destination CGA's respectively of the data packet that initially triggered this exchange. Thus, a Tunnel Mode CGA SP always contains an entry with the unspecified IPv6 address (i.e., ::0) as a placeholder for both tunnel endpoints (local and remote). If we denote by NetworkPrefix/pflen the network prefix and associated length where initiator resides, a typical Tunnel Mode CGA SP should look like that on the interface of GW_i attached to the Internet:

```
INBOUND:

::0/0[ike] -> GW_i/128[any] udp bypass

::0/0[any] -> GW_i/128[ike] udp bypass

::0/0[any] -> NetworkPrefix/pflen[any] any require (ipsec/ah/esp/
tunnel=::0->::0)

OUTBOUND:

GW_i/128[any] -> ::0/0[ike] udp bypass

GW_i/128[ike] -> ::0/0[any] udp bypass

NetworkPrefix/pflen[any] -> ::0/0[any] any require (ipsec/ah/esp/
tunnel=::0->::0)
```

GW_i can discover GW_r by initiating the IKEv2 exchange towards a per network prefix anycast address allocated by IANA. Others discovery means are also possible, for example, DNS queries to retrieve the CGA enabled Security Gateway associated with a given host. Gateway's authorization during the IKE_AUTH exchange in case of the self-evident authentication mode of IPSec imply the verification of TS_i and TS_r payloads (initiator's and responder's CGA) against their CGA parameters. Initiator's and responder's CGA parameters are stored in the GW_i's and GW_r PAD (respectively) and exchanged during the IKE_AUTH exchange in CERT payloads (TBD).

When a packet from initiator to responder triggers an IKEv2 exchange, GW_i and GW_r merely prove ownership of their CGAs. Additionally, they check the ownership of CGA for the initiator and responder. In the same time, while GW_i and GW_r posses initiator's and responder's CGA Parameters (respectively) stored in their PAD, GW_i and GW_r prove that they are authorized to be the initiator's and responder's CGA enabled Security Gateway. Following that, they negotiate and configure a pair of bidirectional SA's between the two gateways:

```
GW_i -> GW_r spi=0x... ipsec tunnel ah/esp keys=...

GW_i -> GW_r spi=0x... ipsec tunnel ah/esp keys=...
```

And they finally add two news SPD entries specifying that subsequent communications between initiator and responder's CGA require IPsec protection:

```
initiator_CGA/128[any] -> responder_CGA/128[any] any require
(ipsec/ah/esp/tunnel=GW_i->GW_r)
```

```
      responder_CGA/128[any] -> initiator_CGA/128[any] any require
      (ipsec/ah/esp/tunnel=GW_i->GW_r)
```

5. IKEv2 Payload usage and requirements

A peer implementing the self-evident authentication mode of IPSec has to use IKEv2 payloads in a specific manner. The following subsections describe usage and requirements of some of the IKEv2 payloads while performing IKE_AUTH exchange.

5.1. Identification Payload

The Identification (ID) Payload of IKE contains the name of the entity to be authenticated with the Authentication (AUTH) Payload. When using CGA, the name of the node is its CGA (initiator's or responder's CGA in the transport mode, and GW_i's or GW_r's CGA in the tunnel mode). CGA is embedded within the ID payload under the known IKEv2 type ID_IPV6_ADDR. CGA enabled node or gateway MAY use also IKEv2 ID_FQDN type (TBD). In that case the CGA technique is a natural complement of the DNSSec.

5.2. Certificate Payload

The name of the Certificate (CERT) payload is rather misleading and the CERT payload is not used to transport only certificates, but also different authentication material/credentials. In case of the self-evident authentication mode of IPSec, the CERT payload is used to transmit the CGA Parameters data structure.

Each CGA enabled Node or Security gateway wanting to prove CGA ownership MUST send to peer its CGA and CGA Parameters used when generating its CGA. CGA Parameters data structure requires the new type of CERT payload. That new type of CERT payload will trigger the CGA verification during the IKE_AUTH exchange. In the tunnel mode, beside the CGA Parameters related to their CGA, GW_i and GW_r SHOULD exchange initiator's and responder's CGA Parameters (TBD).

5.3. Certificate Request Payload

The Certificate Request (CERTREQ) Payload is used by a peer to request preferred certificates to its correspondent. A preference is the type of certificate requested as well as an acceptable certificate authority for this type. A peer can include multiples preferences using several CERTREQ payload. For CGA, certificates used would usually be self-signed, though this does not preclude one to generate its CGA using a CA-signed certificate. The related CERTREQ payload MUST be set to the same type as the CERT payload. Thus, the same as for the CERT payload, we need the new type of the CERTREQ payload.

5.4. Authentication Payload

The Authentication (AUTH) Payload contains data used to authenticate the entity named in the ID payload (i.e., the CGA owner). Since CGA are generated using public key cryptography, the AUTH payload has to contain a digital signature of the message computed using the public key contained in the CERT payload in the CGA Parameters data structure. Currently specified digital signature algorithms includes RSA and DSA, but this scheme could be used with any public key cryptographic algorithm.

5.5. Traffic Selector Payload

The Traffic Selector (TS) Payload contains headers used to identify IP packet flows which need IPsec processing. In the case of the self-evident authentication mode of IPSec, those flows will fly between two CGA's. Both in the transport and tunnel mode, TS payloads will contain initiator's and responder's CGA. Hence we require that the TS payloads used contains CGAs. This imply that the TS Type is set to TS_IPV6_ADDR_RANGE. In the transport mode, both the ID payload and the TS payloads contain initiator's and responder's CGAs. In the tunnel mode, ID payload contains gateways' CGAs, while TS payloads contain initiator's and responder's CGA. Those CGA's from TS payloads will subsequently need to be validated against CGA Parameters exchanged in the CERT payload of new type.

6. IKE_AUTH validation

[1] does not mandate that two peers exchanging keys use the same means of authenticating themselves. Available means of authentication are Digital Signatures, Public Key Encryption and Pre-shared Secret. It is explicitly stated that end-points are not required to use the same means of authenticating themselves. One could use pre-shared secret, while the other could use a digital signature. This note does not conflict with that, allowing one or both entities to prove CGA ownership, thus allowing one to possibly use another means of authenticating itself.

CGA-aware IKE peers wanting to exchange traffic with CGA enabled Nodes (e.g. nodes or gateways) MUST verify CGA ownership. CGA-aware IKEv2 implementation should thus be modified to handle CGA verification, which is very similar to how they currently handle self-signed certificates. The implementation MUST verify that the public key contained in the received CGA Parameters generate the address used as IKEv2 identity.

6.1. IPSec Transport Mode with self-evident authentication

Validation of the IKE_AUTH only requires CGA-PK binding verification. The initial IKEv2 exchanges will be as follows:

```
IKE_SA_INIT exchange:

1. initiator -> responder:  HDR, SAi1, KEi, Ni
2. responder -> initiator:  HDR, SAr1, KEr, Nr, CERTREQ=CGA_type

IKE_AUTH exchange:

3. initiator -> responder: HDR,
                           SK {IDi=CGA_i,
                               CERT=CGA_Parameters_i,
                               CERTREQ=CGA_type,
                               AUTH=dig_sig(CGA_Parameters_i_PK),
                               SAi2,
                               TSi=CGA_i, TSr=CGA_r}
4. responder -> initiator: HDR,
                           SK {IDr=CGA_r,
                               CERT=CGA_Parameters_r,
                               AUTH=dig_sig(CGA_Paramters_r_PK),
                               SAr2,
                               TSi=CGA_i, TSr=CGA_r}
```

6.2. IPSec Tunnel Mode with self-evident authentication

Tunnel mode requires the following:

verification of binding between received gateway's CGA Parameters and its CGA/IKEv2 identity (MUST),

verification of binding between received initiator's and responder's CGA in TS payloads (TBD),

verification that the GW_i/GW_r is authorized to act as initiator's/responder's CGA gateway (TBD).

Initial IKEv2 exchanges will be as follows (TBD):

```
 IKE_SA_INIT exchange:

 1. GW_i -> GW_r: HDR, SAi1, KEi, Ni
 2. GW_r -> GW_i: HDR, SAr1, KEr, Nr, CERTREQ=CGA_type

 IKE_AUTH exchange:

 3. GW_i -> GW_r: HDR, SK {IDi=CGA_GW_i,
                           CERT=CGA_Parameters_GW_i,
                           CERT=CGA_Parameters_i,
                           CERTREQ=CGA_type,
                           AUTH=dig_sig(CGA_Parameters_GW_i_PK),
                           SAi2,
                           TSi=CGA_i, TSr=CGA_r}
 4. GW_r -> GW_i: HDR, SK {IDr=CGA_GW_r,
                           CERT=CGA_Parameters_GW_r,
                           CERT=CGA_Parameters_r,
                           AUTH=dig_sig(CGA_Paramters_GW_r_PK),
                           SAr2,
                           TSi=CGA_i, TSr=CGA_r}
```

7. Comparation with Better Than Nothing Security (BTNS)

Differences between the IPSec self-evident verification mode and the BTNS are listed along the following lines.

The IPSec self-evident verification mode, as a slight modification of the regular IKEv2, does not raise new security concerns to IPSec/ IKEv2. The BTNS lacks the authentication, and therefore raises some security concerns that are described below.

Due to the lack of authentication, BTNS does not protect the key exchange itself. Contrary to the regular IKEv2, first IKEv2 exhcange (IKE_SA_INIT) is not integrity protected. This opens the possibility for the masquerader, MITM and DOS attacks. An attacker can easily masquerade as a legitimate client and acquire a sensitive authentication information. It can also establish two different Security Associations between endpoints and thus perform the MITM attack. As described in [3] BTNS detectes mentioned attacks only after the session establishment, which can lead to the CPU exhaustion during the initial IKEv2 exchanges.

The lack of authentication in BTNS constraints the IPSec usage only to services that use the anonymous access.

While BTNS does not require the deployment of identities, the IPSec self-evident verification mode requires the use of either IPv6 addresses or FQDNs as IKEv2 identities. The reduced number of IKEv2 identites does not constrain the IPSec deployment, if we take into account two assumptions: a) it is reasonable to expext that in IPv6 (even with) adresses would be stable, b) it is also reasonable to expect that DNS mappings are up-to-date.

8. Conclusion

This note presents an overview of how IKEv2 and CGA can be combined to achieve self-evident authentication mode of IPSec. The CGA technique is sufficiently well understood and can use widely deployed and implemented mechanisms. This proposal works in the absence of any previously established direct or indirect (via a broker, AAA roaming operator or trusted third party) security relationship. Because of this, these methods are a very practical and deployable means of using IPsec between previously unknown peers.

9. Security Considerations

This document discusses possible use of IKEv2 as a means to prove CGA ownership and exchange keys to bootstrap IPsec SA's. Because IKEv2 has already been specified and this technique only slightly modifies it, we believe that this should not raise other security concerns that those incurred by CGA proof-of-ownership. Though the cryptographic algorithm used are the same, CGA proof-of-ownership is very different in nature to authentication. One must be especially careful when establishing the security policy, as this technique allows nodes that use their own IPv6 CGA to be successfully authenticated as their "owner". This is similar in essence to IKE used with self-signed certificates, with the additional consideration that CGA binds the address to the public key. A CGA may be considered as a verifiable self-generated address.

The self-evident authentication mode of IPsec might be subject to Denial of Service (DoS) attacks. There are two types of such attacks: fake/malicious initiator and fake/malicious destination.

A rogue CGA enabled security gateway may attack from 'outside', trying to exhaust the gateway's resources by attempting to establish as many IPsec tunnels as it can towards machine of the protected network prefix. This is done by initiating many IKEv2 exchanges. The fake initiator typically sends a lot of spoofed packets with random source addresses. This does not cause much harm as the IKEv2 exchange will not progress any further. On the other hand, the malicious initiator sends regular packets to progress into the IKEv2 exchange. The process of mutual gateway's authorization (which is still marked as TBD) could solve this issue.

10. Open Issues

This document introduce a new CERT/CERTREQ payload type (CGA Parameters) which, among other, triggers the CGA self-evident authentication mode within IPSec/IKEv2.

11. Intellectual Property Rights Considerations

The IETF takes no position regarding the validity or scope of intellectual property or other rights that might be claimed pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

The IETF has been notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

12. References

12.1. Normative References

[1] Kaufman, C., "The Internet Key Exchange version 2 (IKEv2)", RFC 4306, December 2005.

[2] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.

[3] Touch, J., Black, D., and Y. Wang, "Problem and Applicability Statement for Better Than Nothing Security (BTNS)", draft-ietf-btns-prob-and-applic-05 (work in progress), February 2007.

12.2. Informative References

[4] Castelluccia, C., Montenegro, G., Laganier, J., and C. Neumann, "Hindering Eavsdropping via IPv6 Opportunistic Encryption", LNCS ESORICS 2004, September 2004.

Authors' Addresses

Julien Laganier
DoCoMo Euro-Labs
Landsbergerstrasse 312
D-80687 Muenchen
Germany

Email: julien.IETF@laposte.net

Gabriel Montenegro
Microsoft Corporation
One Microsoft Way
Redmonds, WA 98052
USA

Email: gabriel_montenegro_2000@yahoo.com

Ana Kukec
University of Zagreb
Unska bb
Zagreb
Croatia

Email: anchie@tel.fer.hr

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).