Network Working Group J. Laganier
Internet-Draft ENS Lyon / Sun Microsystems, Inc.
Expires: December 29, 2003 G. Montenegro
Sun Microsystems, Inc.
June 30, 2003

Using IKE with IPv6 Cryptographically Generated Address
draft-laganier-ike-ipv6-cga-01

Status of this Memo

Copyright Notice

Abstract

This document describes IKE peer authentication via IPv6 Cryptographically Generated Addresses (CGA). This technique can be used to provide 'Opportunistic IPsec' between IPv6 nodes or security gateways. These CGA's have been proposed to solve several security issues in the absence of any centralized trusted security infrastructure.

Table of Contents

1. Introduction and Problem Statement

This document describes how to use IKE with IPv6 Cryptographically Generated Address (CGA). This technique only requires slight modifications and can be used by one or both peers to achieve 'Opportunistic IPsec' ([4]).

One use of CGA is address proof-of-ownership, but it can also be used with authorization certificates (e.g. SPKI, Keynote2) to enable a flexible authorization framework. CGA's have been proposed to solve several security issues in the absence of any centralized trusted security infrastructure, for example, securing Binding Updates in Mobile IPv6, securing Neighbor Discovery for IPv6, and securing Group Membership in Multicast and Anycast communications.

Until now, Opportunistic IPsec deployment 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. The impossibility of always having the choice of obtaining a security association by leveraging a centralized infrastructure has led to this type of cryptographic techniques commonly known as CGA or SUCV to be applied to IPsec.

This note is organized as follows: we will first describe related work and some usage scenarios for a CGA-enabled peer authentication within an IKE exchange, then we will enumerate requirements for those ISAKMP payloads that need it, and finally, describe precisely how to validate the IKE_AUTH_SA and CREATE_CHILD_SA steps of such an IKE exchange.

2. Related work

CGA usage has lacked generality as it has been applied either within specific frameworks like Mobile IP ([5], [6]) or using its own custom protocol, Statistically Unique and Cryptographically Verifiable protocol (sucvP) [5]. Lately, a proposal using Just Fast Keying (JFK) has been put forth ([9]). Nevertheless, we believe that a full-blown key exchange protocol is redundant. Moreover, because the design, implementation and debugging of a new security protocol is especially costly and error-prone, we think that it is not worth "reinventing the wheel". 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 IKE daemons (e.g. racoon, isakmpd, pluto, etc) is another considerable advantage.

Accordingly, this note presents an overview of how to use the Internet Key Exchange 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 IKE via CGA in a standard manner, thus preventing subsequent conflicting definitions.

3. Terminology

Cryptographically Generated Address (CGA): An adress which is obtained using cryptographic material as input parameter to a hash function.

Crypto-Based Identifier (CBID): An identifier which is obtained using cryptographic material as input parameter to a hash function.

CGA enabled Node: An IPv6 node that has one CGA configured as its IPv6 address.

Opportunistic IPsec (OIPsec): Opportunistic IPsec denotes the use of the IPsec family of protocols between previously unknown nodes implementing an Opportunistic Security Policy. This includes running IKE between those peers to establish an IPsec Security Association (ESP and/or AH in either Transport or Tunnel Mode).

Opportunistic Security Gateway (OSGW): An opportunistic gateway is an IPsec security gateway which applies a tunnel mode Opportunistic Security Policy (OSP) to traffic originating from or sinking at its "local network". It has a CBID instead of a CGA, and holds SPKI authorization certificates issued by the CGA's it protects with OIPsec.

Opportunistic Security Policy (OSP): An Opportunistic 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 OSGW's, while in transit in the public internet).

4. Node Configuration and Requirements

Each node needs to prove address ownership of their CGA. Similarly, OSGW's only need to prove identifier ownership of their CBID. Thus, they generates a public-private key pair, PK and SK, respectively. The nodes then use PK to obtain and configure a CGA and the OSGW's a CBID as specified in [5]:

CBID = SHA1_128 (PK)

CGA = NetworkPrefix | SHA1_64 (PK)

Where PK can be a PK certificate (see below)

Those nodes that want to prove that they own their CGA should use it as their so-called IKE "peer" address while sending IKE packets. OSGW's can use any of their addresses, but they need to have an SPKI authorization certificate issued on their behalf by each CGA holder:

CGA_1 => CBID : OSGW

CGA_2 => CBID : OSGW

(...)

CGA_n => CBID : OSGW

Meaning that each CGA_i (0<i<n+1) authorize CBID to act as an OSGW.

For practical reasons, we choose to define CBID and CGA as the hash of the node's X509 certificates, as they contains validity dates and other data that may be useful. In order to limit the validity scope of the PK<->CGA mapping to the network prefix in which the CGA resides, the certificate is concatenated with this network prefix before hashing, as per [6]. This alleviate the problem of pre-computation attacks on the CGA key-space (2^62). Thus, one is not able to re-use the results of a previous brute-force search attacks on CGA.

CBID = SHA1_128 (X509{PK})

CGA = NetworkPrefix | SHA1_64 (X509Cert{PK} | NetworkPrefix)

Notice that the CBID generated from a certificate is indeed very similar to the FullID proposal [7]. Another technique to generate CGA's with an increased security level of 112 bits (instead of the 62 bits provided in the IID) has been described for SEND purposes [8].

5. Usage Scenarios

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

Opportunistic 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 consistent Opportunistic 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. We propose one CGA Opportunistic IPsec scheme per IPsec mode (transport and tunnel).

5.1 Transport Mode Opportunistic IPsec

Transport Mode Opportunistic IPsec secures end-to-end communications between any two previously unknown CGA-enabled nodes implementing an OSP. For instance, let's assume that Alice initially wants to send a data packet to Bob. Transport Mode OSP requires protection of this data packet. As no trust relationship exists between Alice and Bob prior to this, they needs to establish a Transport Mode IPsec Security Association.

[Alice]<=i=p=s=e=c==t=r=a=n=s=p=o=r=t=>[Bob]

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

A typical Transport Mode OSP 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)
```

5.2 Tunnel Mode Opportunistic IPsec

This section uses the model and mechanism described in ([9]) applied with IKE. Tunnel Mode Opportunistic IPsec is used to secure communications between two CGA-enabled nodes (Alice and Bob), while this traffic is in transit between Alice and Bob's OSGW's (GW_i denotes the IKE initiator and GW_r the responder).

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

Bootstrapping a tunnel mode IPsec SA between two CGA-enabled nodes is not as straightforward as it is for transport mode, because (1) the responder OSGW GW_r needs to be discovered by the initiator OSGW GW_i, and (2) both initiator and responder OSGW 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 OSP 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 Alice resides, a typical Tunnel Mode OSP 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 IKE exchange towards a per network prefix anycast address allocated by IANA. Others discovery means are also possible, like those described in [4] that makes use of DNS queries to retrieve the OSGW associated with a given host. OSGW authorization imply the verification of authorization (a.k.a. delegation) certificates with the TS_i and TS_r payloads. Each GW holds a Crypto-Based Identifier (CBID) and each node that want its traffic to be protected by this gateway uses a CGA. The gateway holds one SPKI authorization certificate per node it protects. For instance, Alice should provide its OSGW GW_i with an authorization certificate issued by her CGA authorizing the CBID of GW_i to act as an OSGW:

Alice_CGA =>GW_i_CBID : OSGW

Bob should similarly provide its OSGW GW_r with a certificate issued by his CGA authorizing the CBID of GW_r to to act as an OSGW:

Bob_CGA =>GW_r_CBID : OSGW

When a packet from Alice to Bob triggers an IKE exchange, the two OSGW's GW_i and GW_r merely prove ownership of their CBID's and exchange authorization certificates issued by Alice and Bob's CGAs authorizing their respective OSGW's to act as such. 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 Alice and Bob's CGA's require IPsec protection:

Alice_CGA/128[any] -> Bob_CGA/128[any] any require (ipsec/ah/esp/ tunnel=GW_i->GW_r)

Bob_CGA/128[any] -> Alice_CGA/128[any] any require (ipsec/ah/esp/ tunnel=GW_i->GW_r)

6. ISAKMP Payload usage and requirements

A peer implementing OIPsec has to use ISAKMP payloads in a specific manner. The following subsections describe usage and requirements of some of the ISAKMP payloads while performing IKE_AUTH and CREATE_CHILD_SA exchanges.

6.1 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. Though CGA are IPv6 Addresses as well, a peer embedding its CGA within the ID payload under the type ID_IPV6_ADDR would not trigger any verification of the PK-CGA binding on the other side. Hence, we believe that a new ID type is needed to explicitly state the cryptographic nature of a CGA and require verification of the binding. Thus, a peer wanting to prove CGA ownership MUST use an ID payload of type ID_IPV6_CGADDR containing its CGA. The value of type ID_IPV6_CGADDR is initially assigned out of the range 249-255 reserved for "private use amongst cooperating systems", as per [2]. If justified, a subsequent, more official assignment will imply IANA involvement. As per CGA, CBID might require a new ID type as well. This is however very similar to the already proposed FullID type [7].

6.2 Certificate Payload

The Certificate (CERT) Payload provide a means to transport certificates within IKE packets. When performing CGA ownership exchange, certificates should be used to transmit to the correspondent the public key used to generate the CGA. When performing a tunnel mode CREATE_CHILD_SA exchange, authorization certificates issued by the data packet source and destination CGA's should be exchanged. Though several types of certificates are specified in [1], we only use those that contains either a public key for CGA proof-of-ownership (i.e., PKCS7_WRAPP_X509_CERT, PGP_CERT, DNS_SIGNED_KEY, X509_CERT_SIGNATURE and SPKI_CERT) or an authorization certificates (i.e., SPKI_CERT). A peer wanting to prove CGA ownership MUST send a CERT payload that contains the public key used when generating its CGA. An OSGW's wanting to prove that it is authorized to act as an OSGW for a given CGA MUST send a CERT payload containing a SPKI authorization certificates issued by this CGA.

6.3 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.

6.4 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. Currently specified digital signature algorithms includes RSA and DSA, but this scheme could be used with any public key cryptographic algorithm.

6.5 Traffic Selector Payload

The Traffic Selector (TS) Payload contains headers used to identify IP packet flows which need IPsec processing. In the case of CGA OIPsec, those flows will fly between two CGA's. Hence we require that the TS payloads used contains CGA's. This imply that the TS Type is set to TS_IPV6_ADDR. Those CGA's will subsequently need to be validated against X509 and possibly SPKI certificates contained in the CERT payloads exchanged.

7. IKE_AUTH and CREATE_CHILD_SA 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 OSGW's) MUST verify CGA ownership. CGA-aware IKE implementation should thus be modified to handle CGA verification, which is very similar to how they currently handle self-signed certificates. Apart from verifying the self-signed certificate, the implementation MUST verify that the public key contained in the certificate (or the certificate itself) generate the address used in the identity payload as previously detailed (ID_IPV6_CGA == SHA1(X509Cert{PK} | NetworkPrefix).

7.1 Opportunistic Transport Mode

Validation of the IKE_SA_AUTH only requires CGA-PK binding verification. Because the IKE peers that just prove CGA ownership will also be the endpoints of any subsequently created transport mode CHILD_SA, validation of future CREATE_CHILD_SA requests will obviously not require additional verification since the endpoints CGA's are already verified.

7.2 Opportunistic Tunnel Mode

Tunnel Mode requires that an OSGW verify the PK-CBID binding of its correspondent OSGW (instead of PK-CGA), and the PK-CGA binding of the source and destination CGA's of the data packet that initially triggered this exchange. Those CGA's are embedded within the TS_i and TS_r payloads. Then the two OSGW's mutually prove themselves that they add been authorized to act as OSGW's for the traffic implied by TS_i and TS_r.

- The responder verifies that it has an SPKI authorization certificate issued by the destination CGA embedded in the TS_r payload, and vice versa for the initiator.

- The responder verify that it received a CERT payload containing a valid SPKI authorization certificate issued by the CGA embedded within the TS_i payload, and vice versa for the initiator.

8. Conclusion

This note presents an overview of how IKE and CGA can be combined to achieve Opportunistic 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 IKE as a means to prove CGA ownership and exchange keys to bootstrap IPsec SA's. Because IKE 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 Opportunistic IPsec application of this scheme might be subject to Denial of Service (DoS) attacks. There is two types of such attacks: fake/malicious initiator and fake/malicious destination.

A rogue opportunistic security gateway may attack from 'outside', trying to exhaust the gateway's resources by attempting to establish as many opportunistic IPsec tunnels as it can towards machine of the protected network prefix. This is done by initiating many IKE exchanges. The fake initiator typically sends a lot of spoofed packets with random source addresses. This does not cause much harm as the IKE exchange will not progress any further. On the other hand, the malicious initiator sends regular packets to progress into the IKE exchange. Fortunately, as the gateway will refuse an exchange that is not about protecting a node for which it had a SPKI delegation certificate, the attacker need to know which protected node to attacks to succeed in its attack. Solutions are either to perform a brute-force 'search' on a possible destination CGA while negotiating the CHILD-SA, but then the attacker is committed to complete an IKE exchange per attacked address. This might eventually lead to a detection of the attack.

10. Open Issues

This document introduce a new ID payload type, ID_IPV6_CGADDR. However, it is not yet clear what is the most appropriate means of requiring peers to verify the PK-CGA binding. Other means are possible. In particular, the revised identity proposal [7] seems to fulfill the requirements for CGA's and CBID's proof-of-ownership.

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.

Normative References

[1] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[2] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.

[3] Maughan, D., Schneider, M., Schertler, M. and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.

Informative References

[4] Richardson, M. and H. Redelmeier, "Opportunistic Encryption using The Internet Key Exchange (IKE)", draft-richardson-ipsec-opportunistic-11.txt (work in progress), January 2003.

[5] Montenegro, G. and C. Castelluccia, "Statistically Unique and Cryptographically Verifiable (SUCV) Identifiers and Addresses.", NDSS 2002, February 2002.

[6] Roe, M., Aura, T., O'Shea, G. and J. Arkko, "Authentication of Mobile IPv6 Binding Updates and Acknowledgments", draft-roe-mobileip-updateauth-02 (work in progress), February 2002.

[7] Hoffman, P., Aura, T., O'Shea, G. and J. Arkko, "Revised Use of Identity in Successors to IKE", draft-ietf-ipsec-revised-identity-00 (work in progress), April 2002.

[8] Aura, T., "Cryptographically Generated Addresses (CGA)", draft-aura-cga-00 (work in progress), February 2003.

[9] Castelluccia, C. and G. Montenegro, "IPv6 Opportunistic Encryption", INRIA Research Report RR-4568, October 2002.

[10] Kaufmann, C., "Internet Key Exchange version 2", draft-ietf-ipsec-ikev2 (work in progress), 2003.

[11] Castelluccia, C. and G. Montenegro, "Securing Group Management in IPv6 with Cryptographically Generated Addresses", IEEE ISCC 2003, July 2003.

Authors' Addresses

Julien Laganier
ENS Lyon / Sun Microsystems, Inc.
180, avenue de l'Europe
38334 Saint Ismier CEDEX
France

EMail: julien.laganier@sun.com

Gabriel Montenegro
Sun Microsystems, Inc.
180, avenue de l'Europe
38334 Saint Ismier CEDEX
France

EMail: gab@sun.com

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.