Transaction SIGnature (TSIG) using CGA Algorithm in IPv6

rafiee-intarea-cga-tsig-00

http://tools.ietf.org/html/draft-rafiee-intarea-cga-tsig-00

Hosnieh Rafiee, Dr. Martin von Löwis &
Prof. Dr. Christoph Meinel
Hasso Plattner Institute, Germany

IETF85
Intarea WG
Atlanta, GA
November 5, 2012

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Agenda

- Introduction
- Brief description of TSIG & problem statement
- What CGA-TSIG aims to do (problem targeted)
- Why it is necessary
- Is it a local solution or global solution?
- Changes made to version 1 and incorporated in version 2 of the draft

Introduction

- I am currently working on research projects at Hasso Plattner Institute concerning:
  - IPv6 security
    - Secure IPv6 addressing
  - Security flaws in application layer services such as DNS, Email and Web in IPv6 networks
- Why we wrote this draft RFC
  - In my research work I uncovered a deficiency in the IPv6 autoconfiguration protocol that might lead to privacy and security issues.
    - The use of NDP with IPv6 addressing scheme leads to security issues
    - To secure NDP, SEND is currently used which leads to privacy issues
    - The use of nodes' temporary addresses solves the privacy issue, but leads to authentication problems in application layers that use IP addresses (DNS, Email…)

Brief description of TSIG (RFC 2845)

- Usage : securing Dynamic DNS Update (DDNS)
  - Allow for transaction level authentication using shared secrets
  - Provide data integrity by using one-way hashing and shared secret keys to establish a trust relationship
- Shortcomings : Manual steps prior to secure processing
  - Generation of shared secret for each pair of hosts manually
  - Change DNS configuration file (named.conf) to accept updates from a particular host.
    - Since shared secret between two hosts only, the server must be told what key is to be used.
  - Update the DNS configuration file whenever any of the two hosts changes its IP address.

if the IP address of hostb is 192.168.1.1 the configuration file for hosta is:

```
server 192.168.1.1 {
  keys { hosta-b. ;};
};
```

What CGA-TSIG aims to do (problem targeted)

- To resolve the authentication issue a DNS server has during the DNS Update process without, or with minimal, human intervention, while remaining within the confines of NDP in IPv6 networks.
- To enhance DNS Update when there is a need to maintain privacy. This can be done by reducing the lifetime of the IP address.
  - Maintaining privacy is an important issue in IPv6, when nodes, on the network, must frequently change their IP addresses in order to prevent being tracked by attackers. This makes it difficult to authenticate the update requestor of the DNS RRs, based solely on the source IP address.
- To reduce the DNS traffic by reducing the number of messages exchanged between a host and a DNS server. (in DNSSEC (RFC 4033), another security mechanism in DNS Update)
- To reduce the authentication time by using the data cached from the generation, using SEND, of the new IP address.
- To protect DNS Update messages against IP spoofing and other types of spoofing attacks -- Man-in the middle, Replay attacks.

Why it is necessary

- To minimize the role of humans in DNS configuration and simplify this process
  - When a node's IP address is changed, if the TSIG mechanism is used, then the DNS administrator will need to change the settings in the DNS configuration file
  - **Without our change, human intervention is required for every change, but with our scheme human intervention is only required the first time**
    - for two DNS servers: To add the IP address of one DNS server to the other's configuration file
    - For a DNS server and a client: No administrative intervention is needed. The user himself can install a CGA-TSIG enabled DNS client application
- To use the same values for the authentication of the DNS update that were used in generating an IP address
  - This secures the authentication and saves time (kill two birds with one stone)

Is it a local solution or global solution?

- It is actually a solution for both local and global networks

How?

- CGA-TSIG works in the application layer and uses the same format as other DNS Resource Records. Therefore it is not limited to a local link. For sending this message, TCP can be used as a transport layer protocol.
- DNS masters or slaves can also use SEND to set their IP addresses dynamically. CGA parameters can be cached in those servers like clients. This also might reduce the chance of attacks against DNS servers since the IP address is temporary and of short duration. Thus there might not be enough time for an attacker to prepare an attack against it.

This slide depicts changes made to version 1 and incorporated in version 2 of the draft

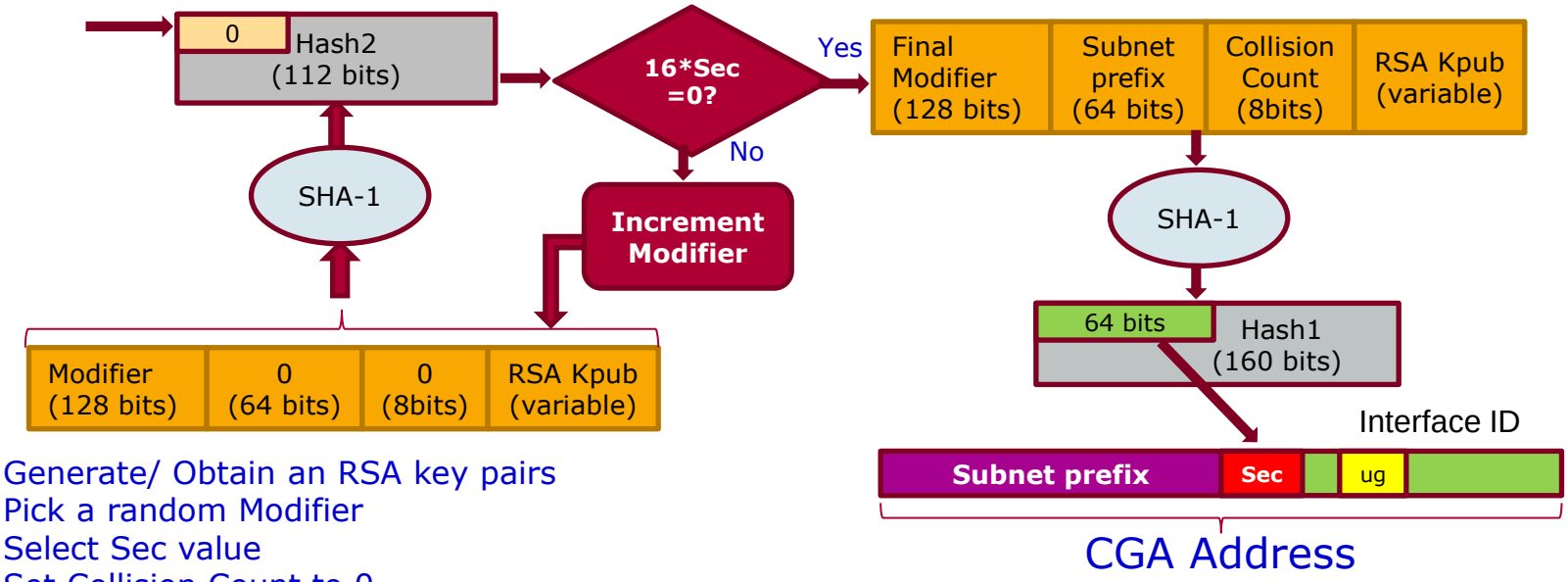
- Modified the two scenarios with respect to a DNS server and client
  - The DNS server needs to store the client's public key the first time the client connects to this server to allow it to update its own Resource Records after the generation of a new IP address
- Clarified the explanation of the authentication process between two DNS servers
  - The administrator of the DNS server only needs to manually add the IP addresses of these servers to the DNS configuration file, once.
- The modified TSIG RR uses the same format as other RRs in use in the DNS field. This is explained in section …. changed to the DNS RR format is explained…
- Resolved the formatting problems
- CGA-TSIG data field moved to Other DATA.
  - If the other DNS does not support CGA-TSIG, it just ignores it

Questions asked

- Question: If you enable CGA-TSIG, any client can send any update and keep doing it forever. how this attack can be addressed without further access control
  - A client can only update its own Resource Records on DNS and not for others. CGA-TSIG is not intended to prevent DoS attacks! We prevent spoofing and spoofing type attacks (Man in the middle and …)
- Question: I doubt any DNS server would accept update from any client just because it can verify the CGA. (using random subnet prefix if there is no >ingress filtering)
  - CGA generates an Interface ID for IPv6 addresses which is made up of the 64 rightmost bits of an IPv6 address. The subnet prefix is not random. Thus filtering will have no effect on CGA, as the host will not regularly change the router prefix, when it is in the same network.

Thank you

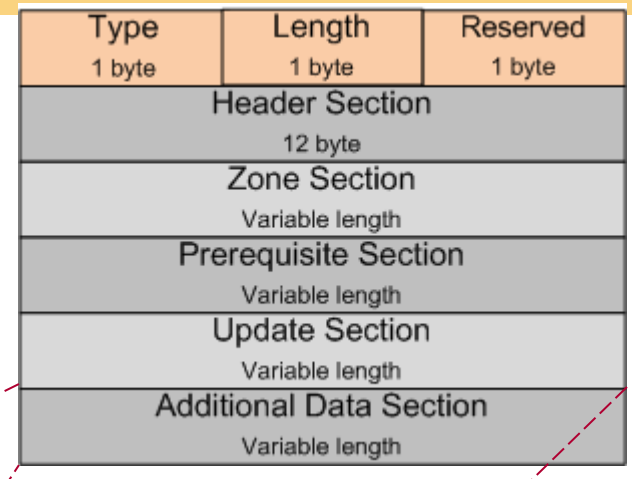
CGA algorithm



- Generate/ Obtain an RSA key pairs
- Pick a random Modifier
- Select Sec value
- Set Collision Count to 0

1. Set CGA initial values
2. Concatenate (Modifier, 0, 0, Kpub)
3. Execute SHA-1 algorithm
4. Compare the 16xSec to 0
5. Concatenate (CGA parameters)
6. Execute SHA-1 algorith
7. Form an interface ID
8. Concatenate (Prefix, Interface ID)
9. Check the uniqniess of IPv6 address

CGA-TSIG message format



DNS Update Request
With CGA-TSIG data field
(RFC 1035, RFC 2845
CGA-TSIG draft)