Network Working Group
Internet Draft
Updates RFC 2845 (if approved)
Intended status: Standards Track

H. Rafiee
Hasso Plattner Institute
M. v. Loewis
Hasso Plattner Institute
C. Meinel
Hasso Plattner Institute

Expires: April 2013

October 2, 2012

Transaction SIGnature (TSIG) using CGA Algorithm in IPv6
<draft-rafiee-cga-tsig-00.txt>

Status of this Memo

Copyright Notice

Abstract

The first step of Transaction SIGnature (TSIG) (RFC 2845) is to generate a shared secret and exchange it manually between a DNS server and a host. This document, CGA-TSIG, proposes a possible way to automate the now manual process for the authentication of a node with a DNS server during the DNS Update process by using the same parameters as are used in generating a secure address in IPv6 networks, i.e., Cryptographically Generated Addresses (CGA) (RFC 3972). CGA-TSIG facilitates this authentication process and reduces the time needed for DNS Updates. The current signature generation process and verification mechanism in TSIG are thus replaced with CGA. This algorithm is added, as an extension, to TSIG to eliminate the human intervention needed for generation and exchange of keys between a DNS server and a host when SEcure Neighbor Discovery (SEND) (RFC 3971) is used.

Table of Contents

1. Introduction

Transaction SIGnature (TSIG) [RFC2845] is a protocol that provides endpoint authentication and data integrity by using one-way hashing and shared secret keys to establish a trust relationship between two hosts which, can be, either a client and a server or two servers. The TSIG keys are manually exchanged between these two hosts and they

must be maintained in a secure manner. This protocol is used to secure a Dynamic Update or to give assurance to the slave named server that the zone transfer is from the original master named server and that it has not been spoofed by hackers. It does this by verifying the signature with a cryptographic key shared with that of the receiver.

The TSIG protocol can be extended using newly defined algorithms. This document defines an algorithm based on the Cryptographically Generated Addresses (CGA) [RFC3972]. CGA is one of the important options in SEcure Neighbor Discovery (SEND) [RFC3971] that can easily provide nodes with the necessary proof of address ownership by providing a cryptographic binding between a host and its IP address without the introduction of any new infrastructure.

2. Conventions used in this document

In examples, "C:" and "S:" indicate lines sent by the client and server respectively.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the characters ">>" preceding an indented line(s) indicates a compliance requirement statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the explicit compliance requirements of this RFC.

3. Algorithm Overview

CGA is a one-way hashing algorithm used to generate Interface IDs for IPv6 addresses in a secure manner. An interface ID consists of the rightmost 64 bits of the 128 bit IPv6 address. CGA verifies the address ownership of the sender by finding a relationship between the sender’s IP address and his public key [1,2].

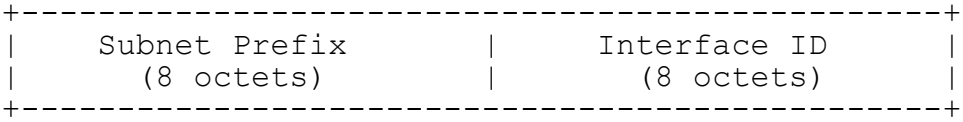

Figure 1 IPv6 addresses

3.1. CGA Generation Algorithm

A node proceeds with the following steps to generate the CGA:

1. Key pairs, called public/private keys, and a random number, called a modifier, are generated [key pair format: Section 3. RFC3972]

   It is recommended that key pairs be generated on the fly for the following reasons:

   - To decrease the randomization of IP addresses
   - To eliminate the need to have the keys manually generated and saved in a particular path, in the node, before the start of IP address generation
   - To decrease the chance of private key theft

2. The modifier is concatenated with other parameters such as a zero value prefix (64 bits), a zero value collision count (1 bit) and the RSA public key

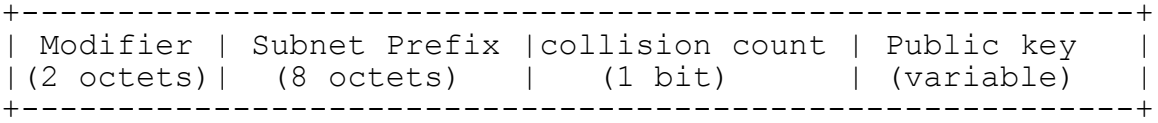

Figure 2 CGA Parameters

3. The Secure Hash Algorithm (SHA1) is executed using the output from step 2. The first leftmost 112 bits of the resulting digest is called Hash2.

4. The computational complexity of Hash2 depends on the Sec value. The Sec is an unsigned 3-bit integer having a value between 0 and 7 (0 being the least secure while 7 the most) which indicates the security level of the generated address against brute-force attacks.

   The 16×Sec leftmost bits of Hash2 are compared to zero. If the condition is not met, the modifier is incremented by one and steps 2 through 4 are repeated. If the condition is met, the next step is executed

5. The modifier is concatenated with the prefix, the collision count, and the public key. SHA1 is executed using that output to create Hash1. The CGA algorithm then uses the leftmost 64 bits from Hash1 and sets the first leftmost 3 bits to the sec value. It also sets bits 7 and 8 (bits u and g) and calls this the Interface ID (IID)

6. The subnet prefix is then concatenated with the IID and the Duplicate Address Detection (DAD) process is executed in order to detect address collision on the network. The node then includes the CGA parameters (modifier, subnet prefix, collision count, public key) with the messages to give other nodes the ability to verify the address ownership of the sender by finding a relationship between the sender's IP address and his public key.

3.2. Modification to TSIG protocol

Normally, to initiate a secure DNS Update process between a DNS server and a host (another DNS server or a client), a minimum of four messages are required to establish a secure channel. A modification to RFC-2845, CGA-TSIG, decreases the number of messages needed in the exchange. The messages used in RFC-2930 (TKEY RR) are not needed when CGA-TSIG is used.

The CGA-TSIG extension uses the creation of a TSIG Resource Record (RR). This RR uses the same data as used to generate a new IP address in a node-- for example, the key pairs (public/private keys), and the output value of the CGA generation function (Interface ID). It is recommended that these values be cached in the node's memory for later use.

3.2.1. Modified TSIG Record format

The modified TSIG RR uses the same format as other RRs in use in the DNS field. This is explained in section 3.2.1 RFC-1035, where the algorithm type must be set to TSIG. The RDATA is also extended in order to store the CGA parameters and the modified CGA signature. The RDATA's algorithm type must be set to CGA-TSIG, a detailed explanation of the RDATA standard fields can be found in section 2.3 RFC-2845. This document focuses only on the new extensions added to RDATA. These new fields are CGA-TSIG Len and CGA-TSIG DATA.

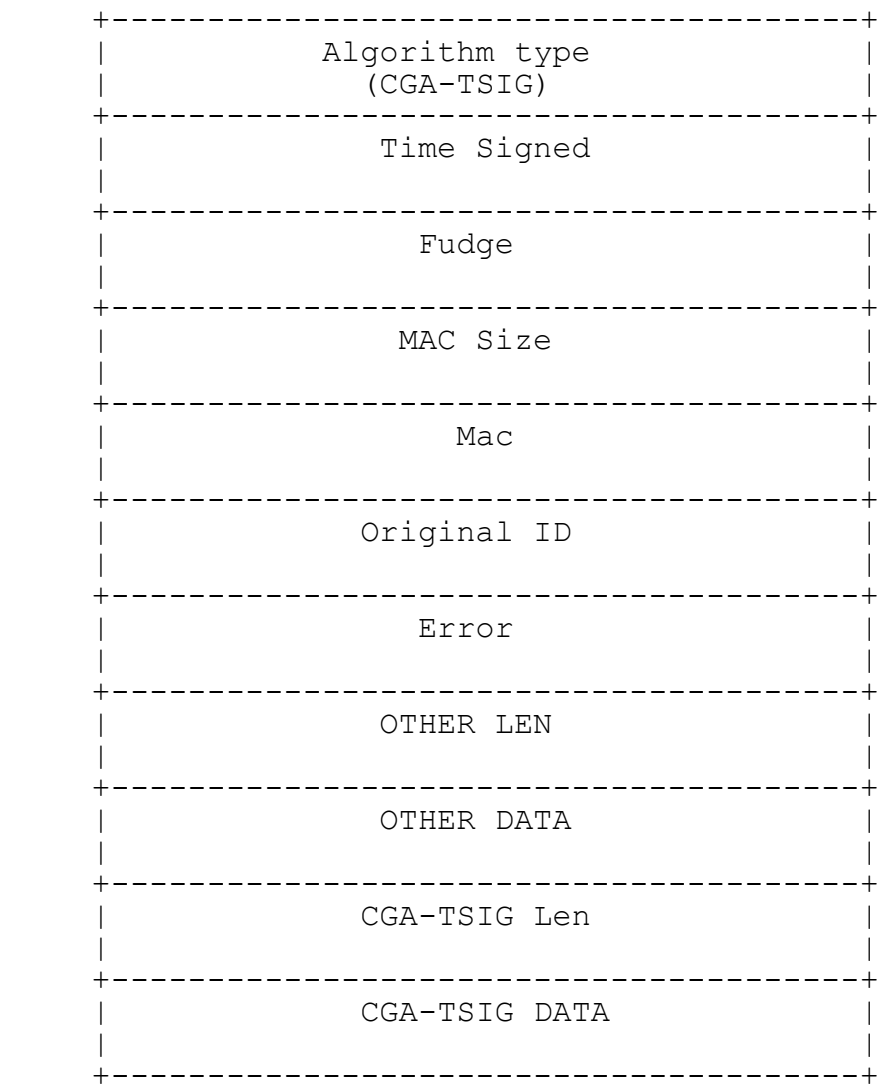

Figure 3 Modified TSIG RDATA

CGA-TSIG DATA

Field Name	Data Type	Notes
Algorithm type	u_int16_t	Name of the algorithm [RFC3972] RSA (by default)
CGA Parameters Len	Octet	the length of CGA parameters

CGA Parameters	variable	Section 3.1 this document
CGA Signature Len	Octet	the length of CGA signature
CGA Signature	variable	Section 3.2.1.1 This document

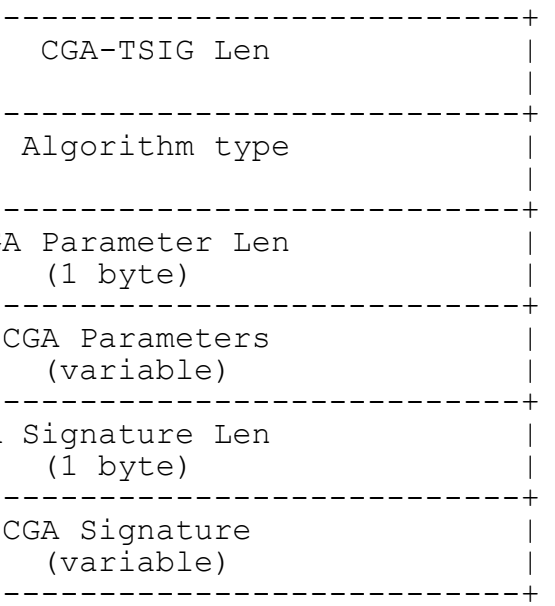

Figure 4 CGA-TSIG Len and CGA-TSIG DATA

3.2.1.1. Generation of the DNS Update request/response

Both the DNS update request and response messages must contain the CGA-TSIG option. To generate the CGA-TSIG DATA, a DNS server and a host must follow steps 1 and 2. In the case where key pairs and CGA parameters are cached during the generation of the IP address by using SEND, the value for the first two steps can be obtained from cache.

1. Execute steps 1 through 4 of section 3.1.

   It is recommended that the same algorithm be used to generate both CGA key pairs and to generate and sign the CGA in the CGA-TSIG DATA Field. In the case of multiple DNS servers (authentication of two DNS servers), there are three possible scenarios with regard to the authentication process, which differs from that of the authentication of a node (client) with one DNS server because of the need for human intervention.

   a. Add DNS servers' IP address to a slave configuration file

A DNS server administrator should only manually add the IP address of the master DNS server to the configuration file of the slave DNS server. When the DNS update message is processed, the slave DNS server can authenticate the master DNS server based on the source IP address and then prove the ownership of this address by using CGA. This scenario is valid until the IP address in any of these DNS servers changes. To automate this step's process, the DNS Update message sender's public key may be saved on the other DNS server after the source IP address has been successfully verified for the first time. In this case, when the sender generates a new IP address by executing the CGA algorithm using the same public key, the other DNS server can still verify it and add its new IP address to the DNS configuration file automatically.

b. Manually exchange the public/private keys

An administrator of the DNS servers may need to manually save the public/private keys of a master DNS server in the slave DNS server. This approach does not have the disadvantage of the first approach since, any time any DNS server wants to change its IP address, it will use the public/private keys.

c. Retrieve public/private keys from a third party Trusted Authority (TA)

The message exchange option of SEND [RFC3971] may be used for the retrieval of the third party certificate. This may be done automatically from the TA by using the Certificate Path Solicitation and the Certificate Path Advertisement messages. Like in scenario b, saving the certificate on the DNS server for later use in the generation of its address or in the DNS update process. In this case, whenever any of these servers wants to generate a new IP address, the DNS update process can still be done automatically without the need for human intervention.

2. Generate signature

For signature generation, all CGA parameters (modifier, public key, collision count and subnet prefix), that are concatenated with the DNS update message and the Time Signed field, are signed by using a RSA algorithm and the private key which was generated in the first step. This signature must be added as an extended option to the TSIG RDATA field. Time Signed is the same timestamp as is used in RDATA. This value is the UTC date and time value obtained from the signature generator. This approach will prevent replay attacks by changing the content of the signature each time a node wants to send a DNS Update Request. The Update Message contains all of the DNS update message with the exclusion of the TSIG Resource Records (RRs). A DNS update message consists of a header, a zone, a prerequisite, an update and additional data. The header contains the control information [RFC2136], the zone identifies the zones to which this update should be applied [Section 4.1.2 RFC1035], the prerequisite prescribes the RRs that must be in the DNS database, the update contains the RR that needs to be modified or added and the additional data is the data that is not part of the DNS update, but is necessary in order to process this update.

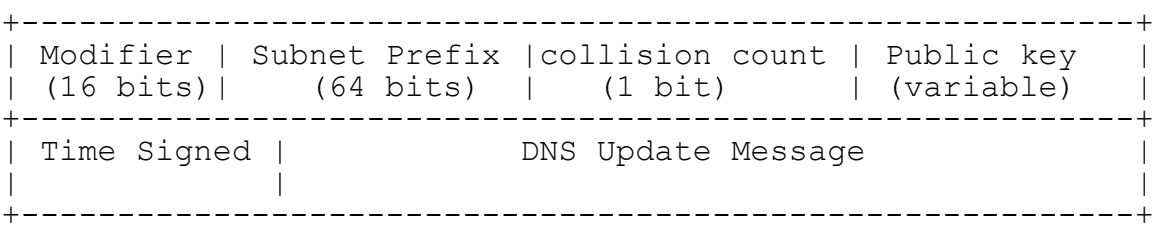

Figure 5 CGA-TSIG Signature

3.2.1.2. Verification of the DNS Update request/response

Sender authentication is necessary to prevent attackers from making unauthorized modifications to DNS servers by use of spoofed DNS Update messages.

1. Check the subnet prefix

   The leftmost 64 bits of IPv6 addresses constitute the subnet prefix. The receiver obtains the subnet prefix from the source IP address in the sender's message. Then, the subnet prefix is obtained from the CGA parameters in the TSIG RDATA field of the received message. A comparison is then made between these two subnet prefixes. If the subnet prefixes match step 2 is executed, otherwise the node is considered as an attacker and the message should be discarded without further action.

2. Check the Time Signed

   The Time Signed value is obtained from the TSIG RDATA and is denoted t1. The current system time is then obtained and converted to UTC time and is denoted t2. If t1 is in the range of t2 and t2 minus 2 seconds (see formula 1, 2 seconds may vary according to the transmission lag time) step 3 is executed, otherwise, the message is considered a spoofed message and the message should be discarded without further action. The range of two seconds is used because the update message may experience a delay during its transmission over TCP or UDP. Both times must use UTC time to avoid any differences in the time based on different geographical locations.

   t2-2 <= t1 <= t2 (1)

3. Compare Hash1 to the Interface ID

   The receiver should obtain all CGA parameters from the TSIG RDATA field and execute SHA1 against them. The leftmost 64 bits of the resulting output constitutes Hash1. Hash1 is then compared to the rightmost 64 bits of the sender's IP address, which is known as the Interface ID (IID). Any differences in the first three leftmost bits of the IID (Sec value) and the u and the g bits (Section 3.1) are ignored. u and g are bits 7 and 8 of the first byte of the IID. If they match step 4 is executed, otherwise, the source is considered as a spoofed source IP address and the message should be discarded without further action.

4. Evaluate Hash2 with CGA parameters

   The receiver obtains the CGA parameters. The collision count and the subnet prefix are set to zero and SHA1 is executed on the resulting data in order to obtain a result of which the leftmost 112 bits are denoted as Hash2. The leftmost 16×sec bits of Hash2 are compared to zero. If the condition is met step 5 is executed, otherwise, the CGA parameters should be consider as spoofed CGA parameters and the message should be discarded without further action.

5. Verify the signature

The signature contained in the TSIG RDATA field of the DNS update message should be verified. This can be done by retrieving the public key from the TSIG RDATA and using it to verify the signature. If the verification process is successful and the node does not want to update another node's RR, then the Update Message will be processed. If the signature verification is successful and the node wants to update another node's RRs, then step 6 is executed. If the verification fails, then the message should be discarded without further action.

6. Verify the Source IP address

If a node wants to update a/many RR(s) on another DNS server, like a master DNS server wanting to update RRs on the slave DNS server, the requester's source IP address must be checked against the one in the DNS configuration file. If it is the same the Update Message should be processed, otherwise, step 7 is executed.

7. Verify the public key

The DNS server checks whether or not the public key retrieved from the TSIG RDATA is the same as what was saved manually by the administrator. If it is the same, step 8 is executed, otherwise, the message should be discarded without further action.

8. Re-generate the signature

The DNS server retrieves the CGA parameters, the Time Signed and the Update Message from the content of the DNS Update Request. These fields are then concatenated. The algorithm type is obtained from the Other Data field of the TSIG RDATA which, by default, should be the RSA, and then a signature is generated using the private key of the sender obtained from the local storage in the DNS server (the key pairs manually saved by administrator). This signature is compared with the signature obtained from the TSIG RDATA. If there is a match, then the Update Message is processed, otherwise, the message should be discarded without further action.

4. Security Considerations

There are several attacks that CGA-TSIG can prevent. Here we evaluate some of these attacks.

4.1. IP Spoofing and Reflector Attacks

During the DNS Update process it is important for both communicating parties to know that the one they are communicating with is the owner

of that IP address and that messages have not been sent from a spoofed IP address. This can be fulfilled by using the CGA algorithm that utilizes the node to verify the address ownership of the other node. The reflector attack is also a kind of distributed Denial of Service attack. It uses the IP address of the victim as a source of the DNS message and sends several queries to the DNS server which then redirects traffic to this victim thus keeping the victim busy processing these packets. Using the CGA signature and authentication approach will prevent this type of attack.

4.2. DNS Dynamic Update Spoofing

Because the signature contains both CGA parameters and the DNS update message, proof is offered of the sender’s address ownership (CGA parameters) and the validity of the update message.

4.3. Resolver Configuration Attack

In CGA-TSIG, the DNS server or the client might not need further configuration. This may reduce the possibility for human errors being inserted into the DNS configuration file. Since this type of attack is predicated on human error, the chances of it occurring when our proposed extension is used are minimized.

4.4. Shared Secret (key pairs) Exposing

On-the-fly key pair generation is recommended to decrease the chances of giving attackers unauthorized access to private keys on a node.

4.5. Replay attack

Using Time Signed in the signature modifies the content of the signature each time the node generates it and sends it to DNS server. This value is the node's current time in UTC. If the attacker tries to spoof this value with another timestamp to show that the update message is current, the DNS server checks this message by verifying and regenerating the signature (when the private key of the other DNS server is manually set in this DNS server). In this case steps 2 and 8 of verification process fail. Therefore, this type of attack is also prevented.

5. IANA Considerations

The IANA has allowed for choosing new algorithm(s) for use in the TSIG Algorithm name. Algorithm name refers to the algorithm described in this document. The requirement to have this name registered with IANA is specified

6. Conclusions

In TSIG, not all processing is done automatically and some steps might need to be done offline. To address this issue, and to automate this process when Secure Neighbor Discovery (SEND) (RFC3971) is used, this document is introduced as an extension to the TSIG protocol (CGA-TSIG) in order to take advantage of the use of CGA for the DNS Update authentication process of a node within a DNS server. CGA-TSIG also decreases the number of messages needed in the exchange between the DNS server and the DNS client during the update process. This enhances the performance of the DNS update process. Since CGA does not need Public Key Infrastructure (PKI) framework to verify the node's address ownerships, the authentication of a node with a DNS server in the DNS update process is automated. This document also makes use of SEND for the authentication of two DNS servers against each other when processing DNS Update messages. However ,the first step should be done manually the first time it is used to afford greater security for this process.

7. References

7.1. Normative References

[RFC2845] Vixie, P., Gudmundsson, O. and Eastlake 3rd, D., Wellington, B., "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)," RFC 3972, March 2005.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2930] Eastlake 3rd, D., "Secret Key Establishment for DNS (TKEY RR)", RFC 2930, September 2000.

[RFC1035] Mockapetris, P., "Domain Names - Implementation And Specification", RFC 1035, November 1987.

[RFC2136] Vixie, P. (Editor), Thomson, S., Rekhter, Y., Bound, J., "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997.

7.2. Informative References

[1] Aura, T., "Cryptographically Generated Addresses (CGA)", Lecture Notes in Computer Science, Springer, vol. 2851/2003, pp. 29-43, 2003.

[2] Montenegro, G. and Castelluccia, C., "Statistically Unique and Cryptographically Verifiable (SUCV) Identifiers and Addresses," ISOC Symposium on Network and Distributed System Security (NDSS 2002), the Internet Society, 2002, http://www.isoc.org/isoc/conferences/ndss/02/papers/monten.pdf.

8. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

Appendix A.

A.1. Copyright

Authors' Addresses

Hosnieh Rafiee
Hasso-Plattner-Institute
Prof.-Dr.-Helmert-Str. 2-3
Potsdam, Germany
Phone: +49 (0)331-5509-546
Email: rafiee@hpi.uni-potsdam.de

Dr. Martin von Loewis
Hasso-Plattner-Institute
Prof.-Dr.-Helmert-Str. 2-3
Potsdam, Germany
Email: martin.vonloewis@hpi.uni-potsdam.de

Professor Dr. Christoph Meinel
Hasso-Plattner-Institute
Prof.-Dr.-Helmert-Str. 2-3
Potsdam, Germany
Phone: +49(0)331-5509-222
Email: meinel@hpi.uni-potsdam.de