M.I.T. Laboratory for Computer Science

Request for Comments No. 207
May 21, 1981

Authentication Server Protocol

by D. Daniels, J. Lucassen, W. Rubin

1. Introduction

This paper describes the interface to a conventional encryption authentication server of the type proposed by [Needham78]. The server is intended to provide a key distribution service to internet users. Keys obtained from an authentication server may be used in secure data communication protocols such as those described by [Kent 76].

This server will use the internet user datagram protocol [Postel 79a]. Encryption will be provided by the Data Encryption Standard [FIPS ??] in cipher block chaining mode [Kent 79].

2. Encrypted Blocks

The encrypted blocks returned by the authentication server will be generated using the cipher block chaining mode of DES. Encrypted blocks will be multiples of eight bytes in length and are padded at the end if necessary. The prefix used in cipher block chaining to exclusive-or with the first eight byte block shall be the encryption key.

The integrity of encrypted blocks is insured by the inclusion of known values at the end of the block.

3. Time Stamp

Needham and Schroeder present a protocol for initiating a conversation using a cached response from the authentication server. Since DES ciphers can be broken by brute force search, it may not be acceptable to start a conversation using too old a key. For this reason a time stamp is included in the encrypted block which might be cached. The recipient of this block may decide upon decrypting the block whether the key is too old to be acceptable for a conversation.

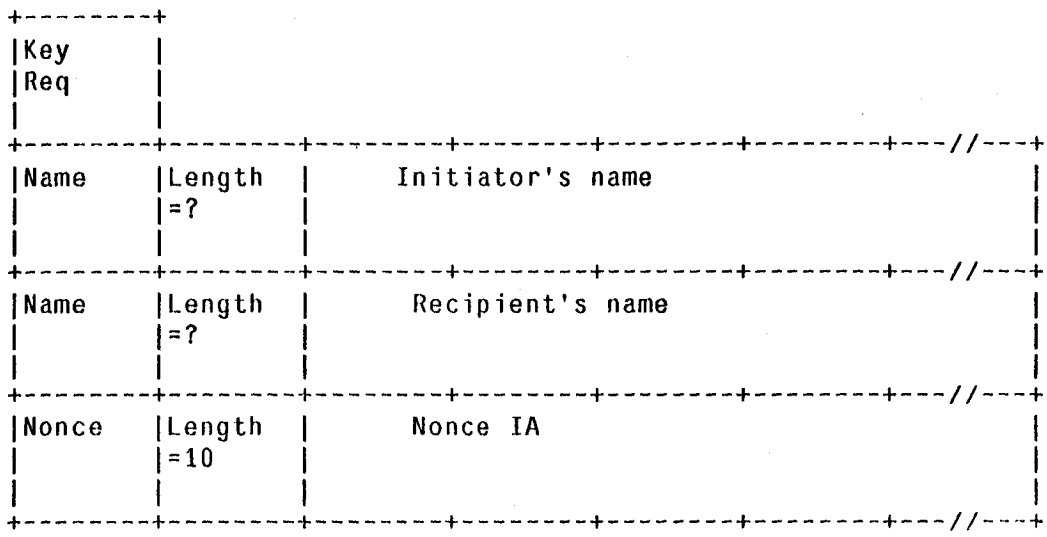
4. Datagrams

The maximum length of the datagrams used by the authentication server may be implementation dependent. The internet protocol [Postel 80], which the user datagram protocol calls on, recommends that hosts limit datagram length to 576 bytes. The internet protocol requires from 20 to 60 bytes of header and the user datagram protocol requires 8 additional bytes of header leaving 508 to 548 bytes as a practical limitation. If names are very long (200 bytes or more) then it may not be possible to complete a reply in one datagram. If this happens then the authentication server returns an error response. Note that this problem only occurs when one client with a long name wishes to talk to another client with a long name. It is recommended that names be kept to reasonable length.

The datagrams sent and received by the authentication server consist of one byte request type codes followed by one or more items. Items consist of a one byte item type code followed by a one or two byte item length, followed by item data. The item length should count the item code byte and the (one or two) item length field byte(s) as well as the data.

4.1. Key Distribution

4.1.1. Request to the Authentication Server



Where:

Key Req — is a one byte request code indicating that this is a key request (= 1).

Name — is a one byte item code indicating that this item is a name (= 1).

Length — is a one byte binary number giving the length of the item in bytes. The item code and item length are counted.

Initiator's name — is a string of ASCII characters.

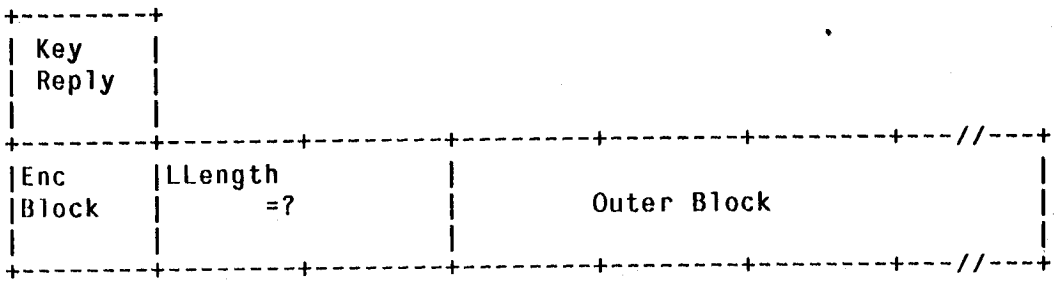
Recipient's name — is a string of ASCII characters.

Nonce — is a one byte item code indicating that this item is a nonce identifier (=4).

Nonce IA — is a eight byte binary number chosen by the sender to use to verify the time integrity of the response.

4.1.2. Response from the Authentication Server:

The following is a normal response from the authentication server. Error responses are explained below.



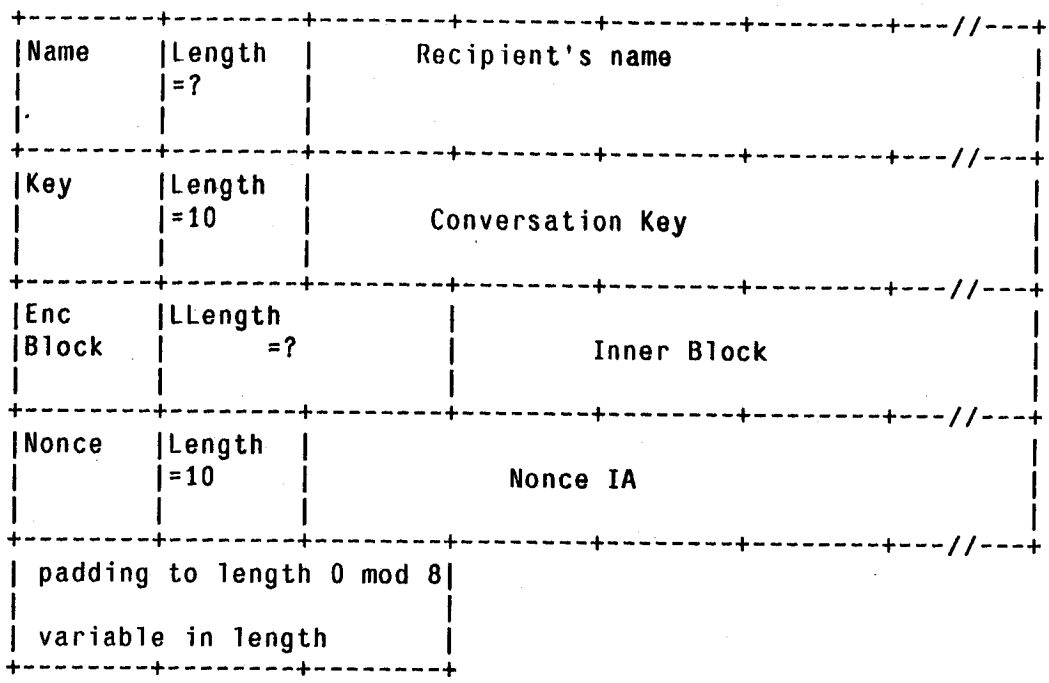
Where:

Key Reply — is a 1 byte reply code indicating that this datagram is a reply to a key request (=2).

Enc Block — is a 1 byte item code indicating that this item is an encrypted block (=254).

LLength — is a two byte binary number giving the length in bytes of the encrypted block including the length and item code bytes.

Outer Block — is a block of items encrypted with the secret key which the Authentication Server had stored under the same name as the ffInitiator's name in the request. This block contains:



Where:

Name — is a one byte item code indicating that this item is a name (=1).

Length — is a one byte binary number giving the length of the item in bytes. The item code and item length are counted.

Recipient's name — is the ASCII character string from the ff1Recipient's Name field of the request datagram.

Key — is a one byte item code indicating that this item is an DES key (=5).

Conversation Key — is an eight byte binary number suitable for use as an DES key. Each byte has odd parity as called for in the DES standard.

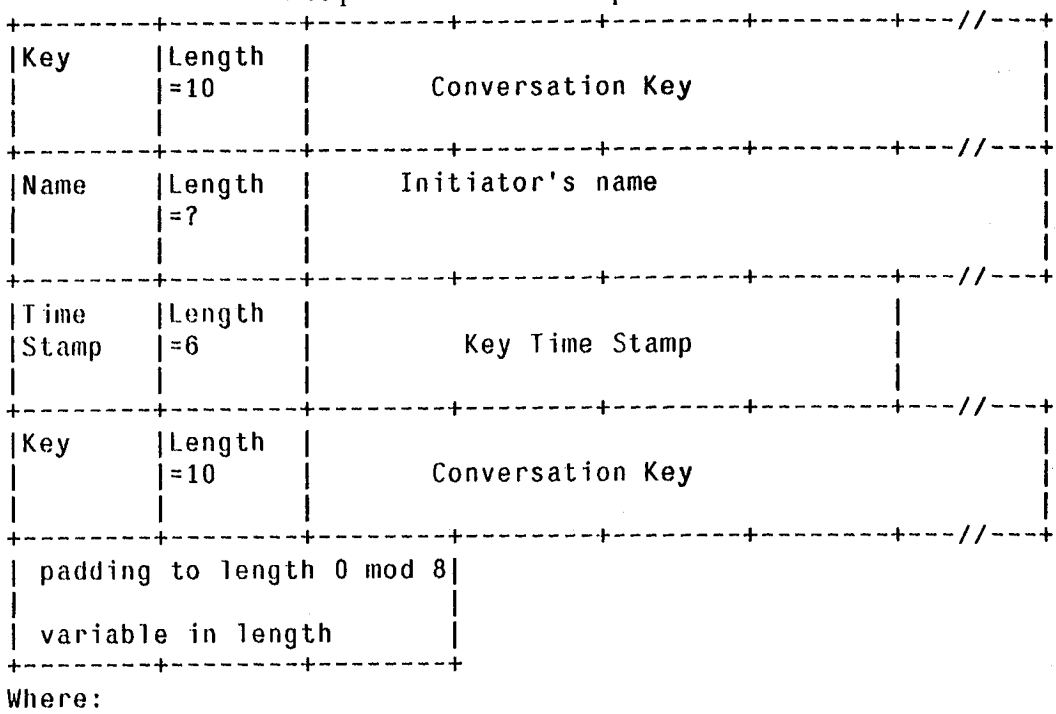
Enc Block — is a 1 byte item code indicating that this item is an encrypted block (=254).

LLength — is a two byte binary number giving the length in bytes of the encrypted block including the length and item code bytes.

Nonce — is a one byte item code indicating that this item is a nonce identifier.

Nonce IA — is the eight byte binary number from the ffINonce IA field of the request datagram. The authentication server client may check this field to verify the integrity of the response.

Inner Block — is a block of items encrypted with the secret key which the Authentication Server had stored under the same name as the Recipients name in the request. The block contains:



Where:

Key — is a one byte item code indicating that this item is an DES key (=5).

Length — is a one byte binary number giving the length of the item in bytes. The item code and item length are counted.

Conversation Key — is an eight byte binary number suitable for use as an DES key. Each byte has odd parity as called for in the DES standard. This is the same key as ffIConversation Key in ffIOuter Block This item is repeated at the end of the block to provide an integrity check.

Name — is a one byte item code indicating that this item is a name (=1).

Initiator's name — is the string of ASCII characters from the ffIInitiator's name field of the request datagram.

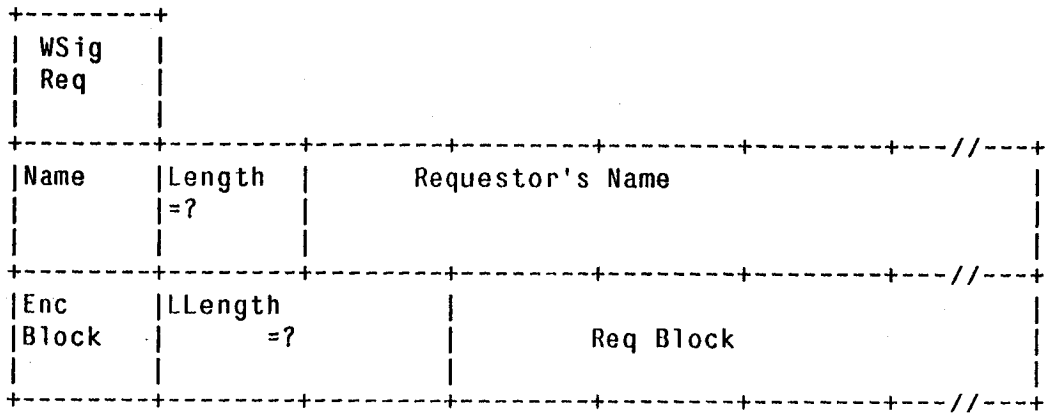
Time Stamp — is a one byte item code indicating that this item is a time stamp (=6).

Key Time Stamp is a four byte binary number giving the time at which the conversation key was created in seconds since 00:00 Jan 1 1900.

4.2. Digital Signatures

4.2.1. Requesting a digital signature:

To request a digital signature from the authentication server the following datagram is sent to the authentication server.



Where:

WSig Req is a one byte request code indicating that this is a request to write a signature block(= 3).

Name is a one byte item code indicating that this item is a name (=1).

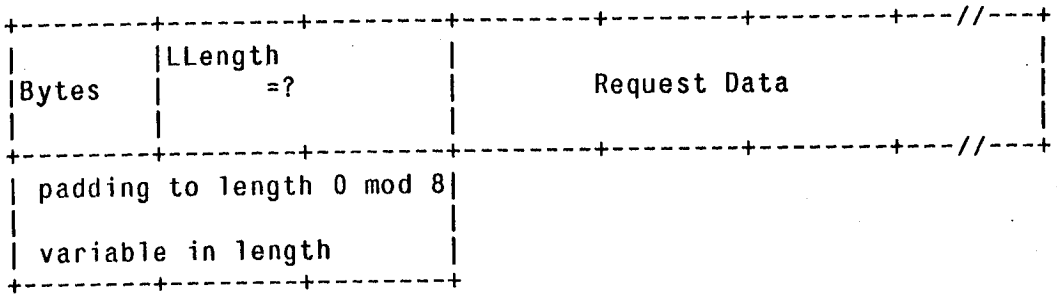
Length is a one byte binary number giving the length of the item in bytes. The item code and item length are counted.

Requestor's Name is a string of ASCII characters.

Enc Block is a 1 byte item code indicating that this item is an encrypted block (=254).

LLength is a two byte binary number giving the length in bytes of the encrypted block including the length and item code bytes.

Req Block is a block containg a single item, padded to length 0 mod 8, and encrypted with the secret key which the Authentication Server has stored under the same name as the Requestor's Name in the request. The block contains:



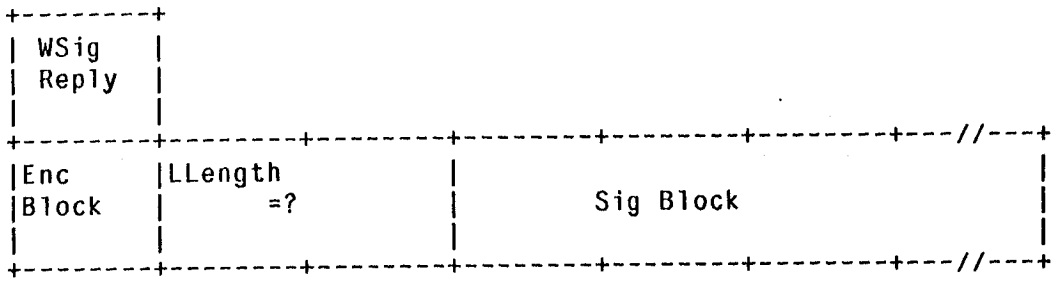
Where:

Bytes — is a one byte item code indicating that this item is an uninterpreted byte string (=253).

LLength — is a two byte binary number giving the length in bytes of the bytes block including the length and item code bytes.

Request Data — is a block of data, variable in length, which the authentication server does not interpret. The data should be the "characteristic value" of the digital signature. An integrity check should be provided by the client.

4.2.2. Response from the Authentication server:



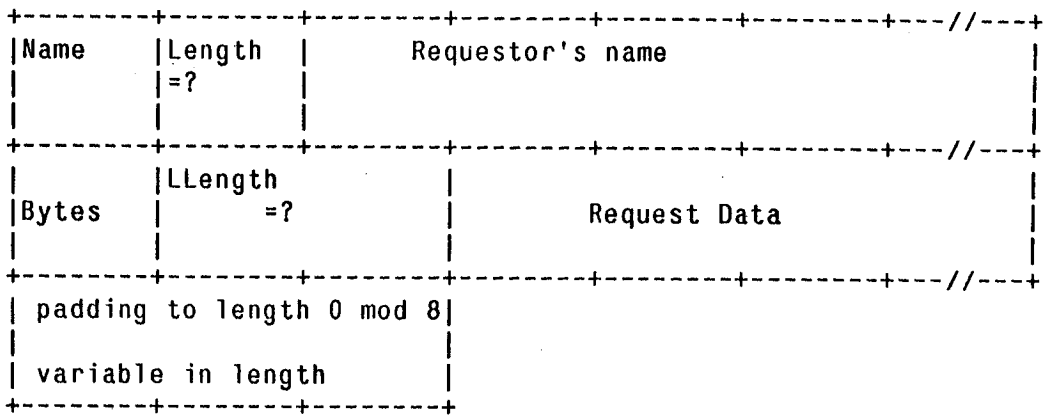
Where:

WSig Reply — is a one byte reply code indicating that this is a reply to a request for a signature block(= 4).

Enc Block — is a 1 byte item code indicating that this item is an encrypted block (=254).

LLength — is a two byte binary number giving the length in bytes of the encrypted block including the length and item code bytes.

Sig Block — is a block of items, padded to length 0 mod 8, and encrypted with the secret key of the Authentication Server. This block contains:



Where:

Name — is a one byte item code indicating that this item is a name (=1).

Length — is a one byte binary number giving the length of the item in bytes. The item code and item length are counted.

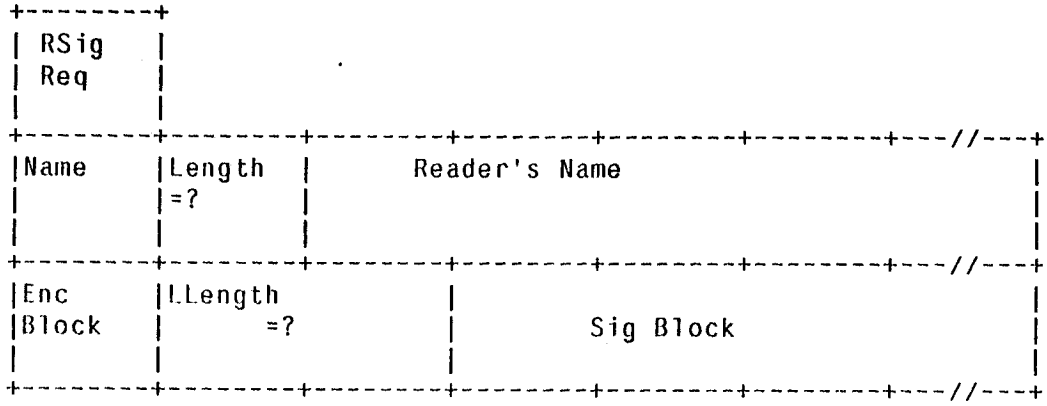
Requestor's Name — is the string of ASCII characters from the requestor's Name field of the request to the authentication server.

Bytes — is a one byte item code indicating that this item is an uninterpreted byte string (=253).

Request Data — is the request data from the request to the authentication server.

4.2.3. Reading digital signatures:

To read a digital signature the following datagram should be send to the authentication server:



where:

RSig Req — is a one byte request type code indicating that this is a request to read a digital signature(= 5).

Name — is a one byte item code indicating that this item is a name (=1).

Length — is a one byte binary number giving the length of the item in bytes. The item code and item length are counted.

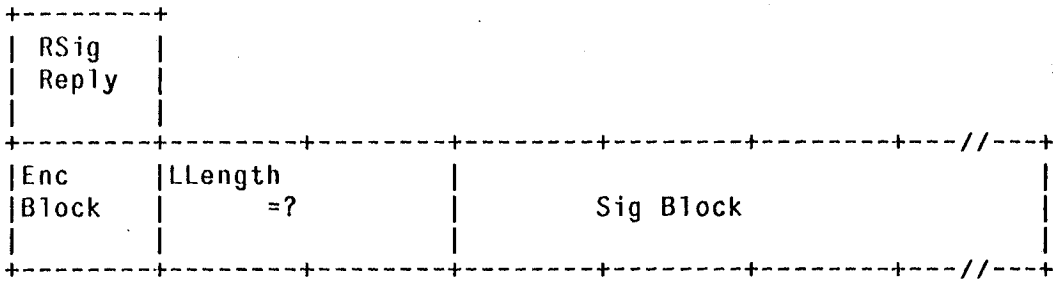
Reader's Name — is a string of ascii characters identifying the client of the authentication server wishing to read the digital signature.

Enc Block — is a 1 byte item code indicating that this item is an encrypted block (=254).

LLength — is a two byte binary number giving the length in bytes of the encrypted block including the length and item code bytes.

Sig Block — is the signature block, encrypted with the secret key of the authentication server, as defined above.

4.2.4. Response from the Authentication server:



Where:

RSig Reply — is a one byte reply code indicating that this is a reply to a request for a signature block(= 6).

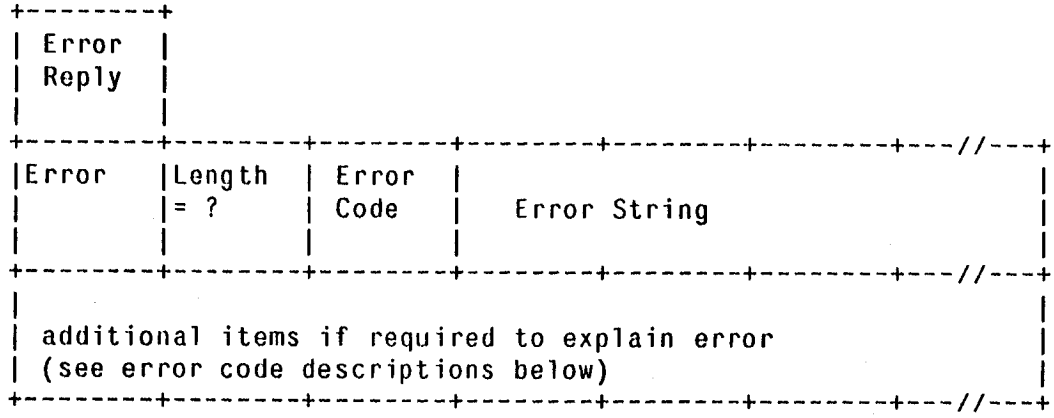
Enc Block — is a 1 byte item code indicating that this item is an encrypted block (=254).

LLength — is a two byte binary number giving the length in bytes of the encrypted block including the length and item code bytes.

Sig Block — is a block of items, padded to length 0 mod 8, and encrypted with the secret key which the authentication server had associated with the Reader's Name from the RSig Req message. The contents of this block are defined above.

4.3. Error Responses

If an error occurs in processing a request the authentication server returns the following response:



where:

Error Reply — is a one byte reply code indicating that this is a response resulting from some error(= 7).

Error — is a one byte item code indicating that this is an error item.

Error Code — is a one byte code specifying the error. The following error codes are defined.

Code	Meaning
0	Undetermined or undefined error
1	Name not found (a name item with the unrecognized name follows)
2	Response would not fit in one datagram

Error String — is a string of ASCII characters explaining the error.

5. Code Summary

Codes were chosen to be compatible with those used by [Postel 79b].

Request/Reply Type Codes:

Type	Value
Key Req	1
Key Reply	2
WSig Req	3
WSig Reply	4
RSig Req	5
RSig Reply	6
Error Reply	7

Item Type Codes:

Type	Value	Length or LLength
Name	1	variable
Error	3	variable
Nonce	4	10
Key	5	10
Time Stamp	6	6
bytes	253	variable
Enc Block	254	variable ((multiple of 8) plus 3)

Error Codes:

Code	Meaning
0	Undetermined or undefined error
1	Name not found
2	Response would not fit in one datagram

References

[FIPS ??]
Federal Information Processing Standards, Specifications for the Data Encryption Standard. National Bureau of Standards, FIPS PUB 46, January, 1977

[Kent 76]
Kent, S. T.
Encryption-Based Protection Protocols for Interactive User-Computer Communication.
Technical Report TR-162, MIT Lab for Computer Science, May, 1976.

[Kent 79]
Kent,S. T.
Protocol Design Considerations for Network Security.
In K.G. Beauchamp, editor, *Interlinking of Computer Networks*. D Reidel, Dordrecht, Holland, 1979.

[Postel 79a]
Postel, J.
User Datagram Protocol.
Internet Experiment Note IEN 88, USC-Information Sciences Institute, May, 1979.

[Postel 79b]
Postel, J.
Internet Name Server.
Internet Experiment Note IEN 89, USC-Information Sciences Institute, May, 1979.

[Postel 80]
Postel, J.
DOD Standard Internet Protocol.
Internet Experiment Note IEN 128, USC-Information Sciences Institute, January, 1980.