DoT(DNS over TLS)

INTRODUCTION

DNS over TLS (DoT) is a network security protocol for encrypting and wrapping Domain Name System (DNS) queries and answers via the Transport Layer Security (TLS) protocol. The goal of the method is to increase user privacy and security by preventing eavesdropping and manipulation of DNS data via man-in-the-middle attacks. The well-known port number for DoT is 853.To use DNS over TLS, both the DNS client and server need to support the protocol. Many DNS servers and clients, including popular web browsers and operating systems, have started to support DNS over TLS as a privacy and security enhancement.

MESSAGE FORMAT:

The DNS over TLS (DoT) protocol uses the standard DNS message format encapsulated within the TLS transport. The DNS header is 12 octets (bytes) in length and has the following structure:

Transaction ID (ID) (16 bits):A 16-bit identifier assigned by the program that generates the DNS query. It is copied into the corresponding DNS response.

Flags (16 bits)

Question Count (QDCOUNT) (16 bits):An unsigned 16-bit integer specifying the number of entries in the question section.

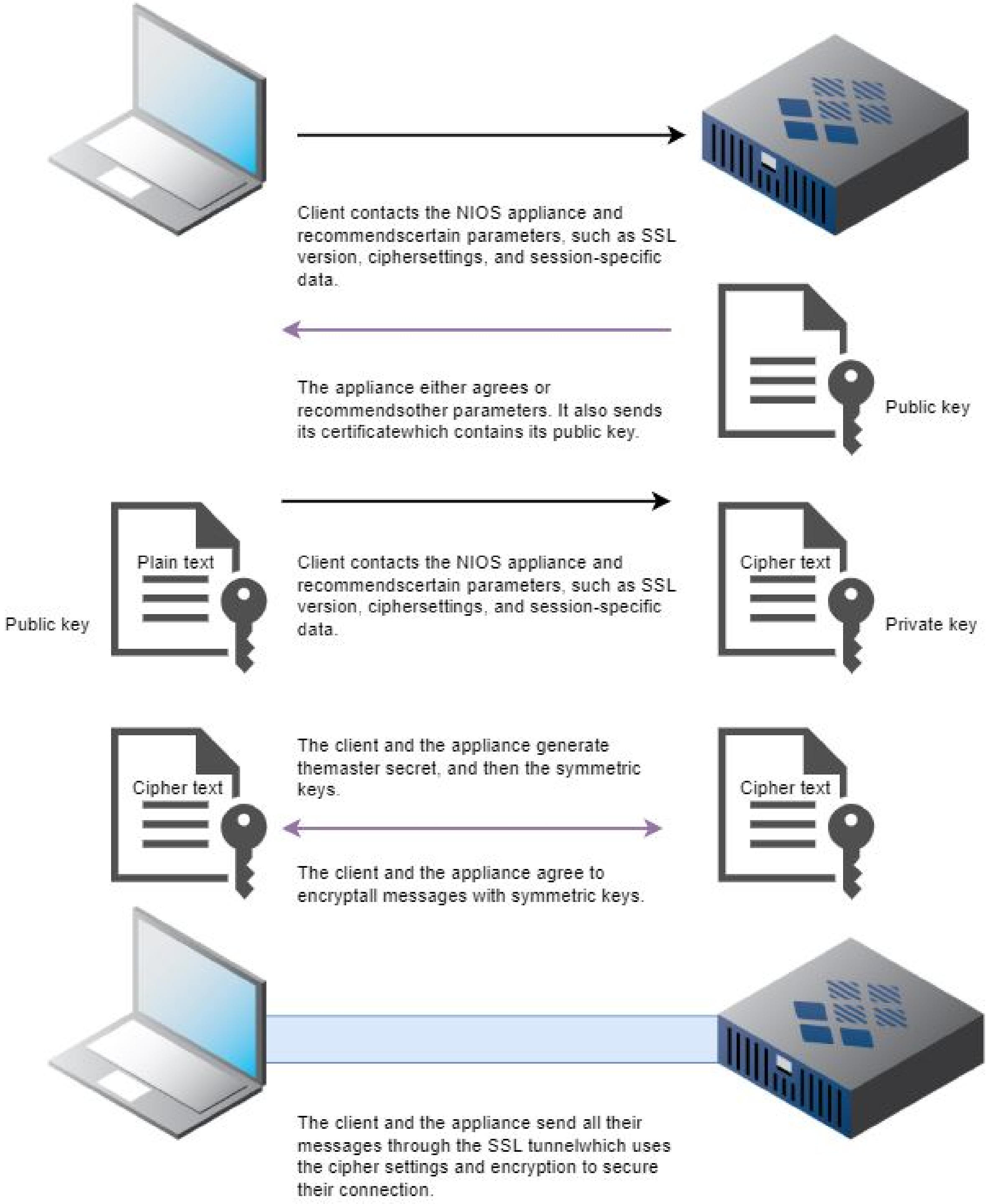
Answer Record Count (ANCOUNT) (16 bits):An unsigned 16-bit integer specifying the number of resource records in the answer section.

Authority Record Count (NSCOUNT) (16 bits):An unsigned 16-bit integer specifying the number of name server resource records in the authority records section.

Additional Record Count (ARCOUNT) (16 bits):An unsigned 16-bit integer specifying the number of resource records in the additional records section.

The DNS header is followed by the question section, answer section, authority records section, and additional records section, each with its own format.

CLIENT-SERVER INTERACTION:



Client contacts the NIOS appliance and recommendscertain parameters, such as SSL version, ciphersettings, and session-specific data.

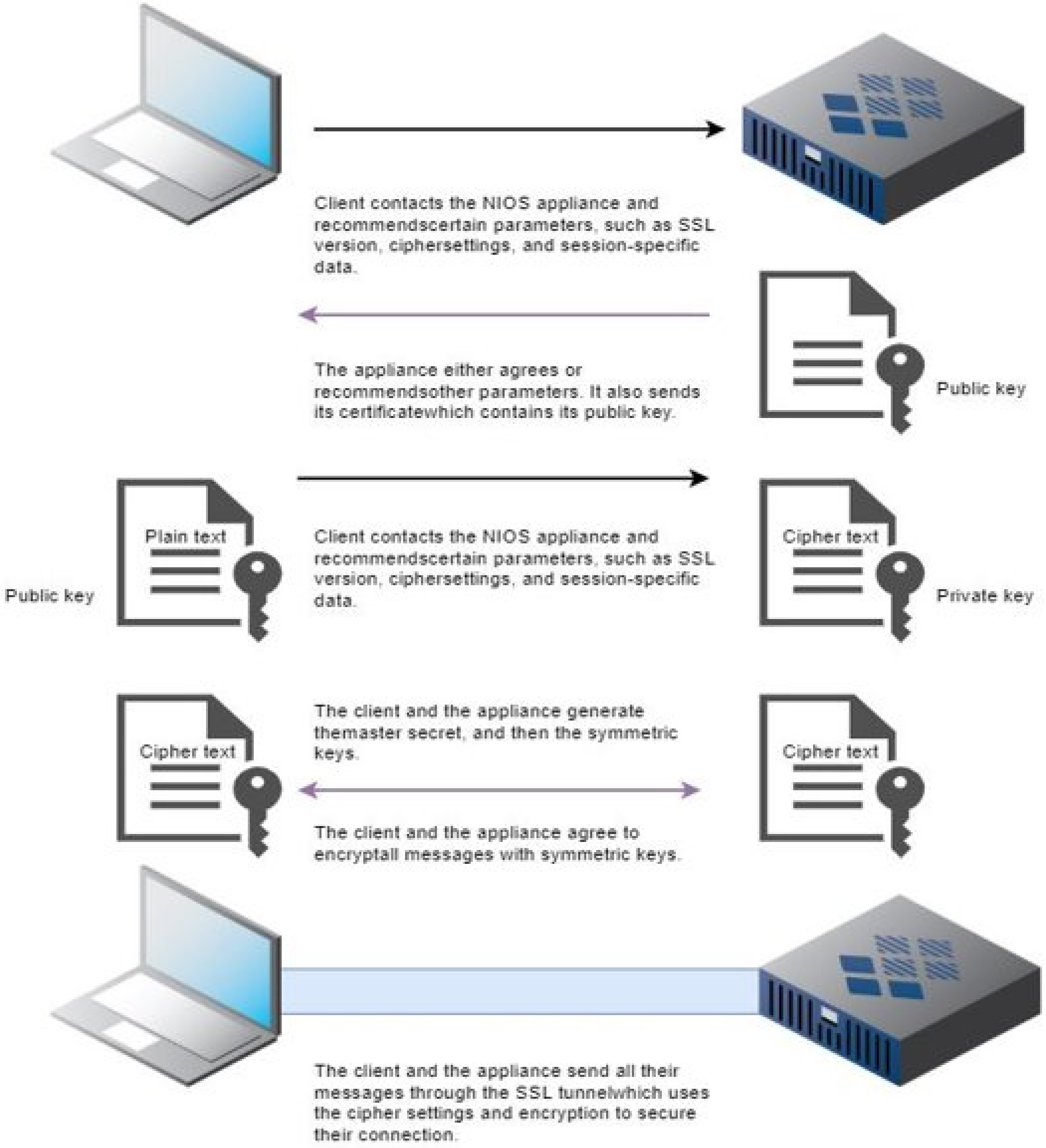
The appliance either agrees or recommendsother parameters. It also sends its certificatewhich contains its public key.

Client contacts the NIOS appliance and recommendscertain parameters, such as SSL version, ciphersettings, and session-specific data.

The client and the appliance generate themaster secret, and then the symmetric keys.

The client and the appliance agree to encryptall messages with symmetric keys.

The client and the appliance send all their messages through the SSL tunnelwhich uses the cipher settings and encryption to secure their connection.



Client contacts the NIOS appliance and recommendscertain parameters, such as SSL version, ciphersettings, and session-specific data.

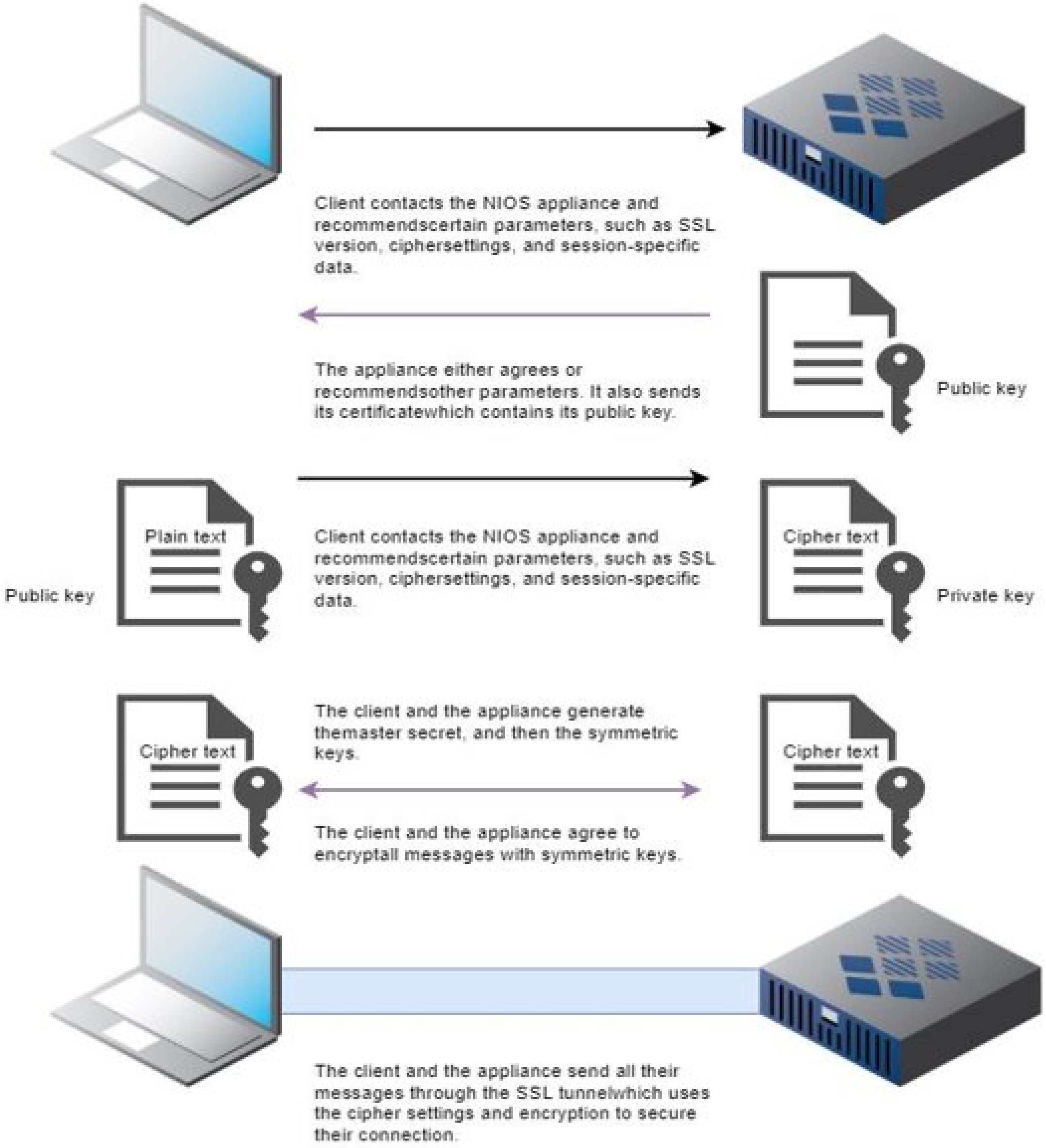
The appliance either agrees or recommendsother parameters. It also sends its certificatewhich contains its public key.

Client contacts the NIOS appliance and recommendscertain parameters, such as SSL version, ciphersettings, and session-specific data.

The client and the appliance generate themaster secret, and then the symmetric keys.

The client and the appliance agree to encryptall messages with symmetric keys.

The client and the appliance send all their messages through the SSL tunnelwhich uses the cipher settings and encryption to secure their connection.



Client contacts the NIOS appliance and recommendscertain parameters, such as SSL version, ciphersettings, and session-specific data.

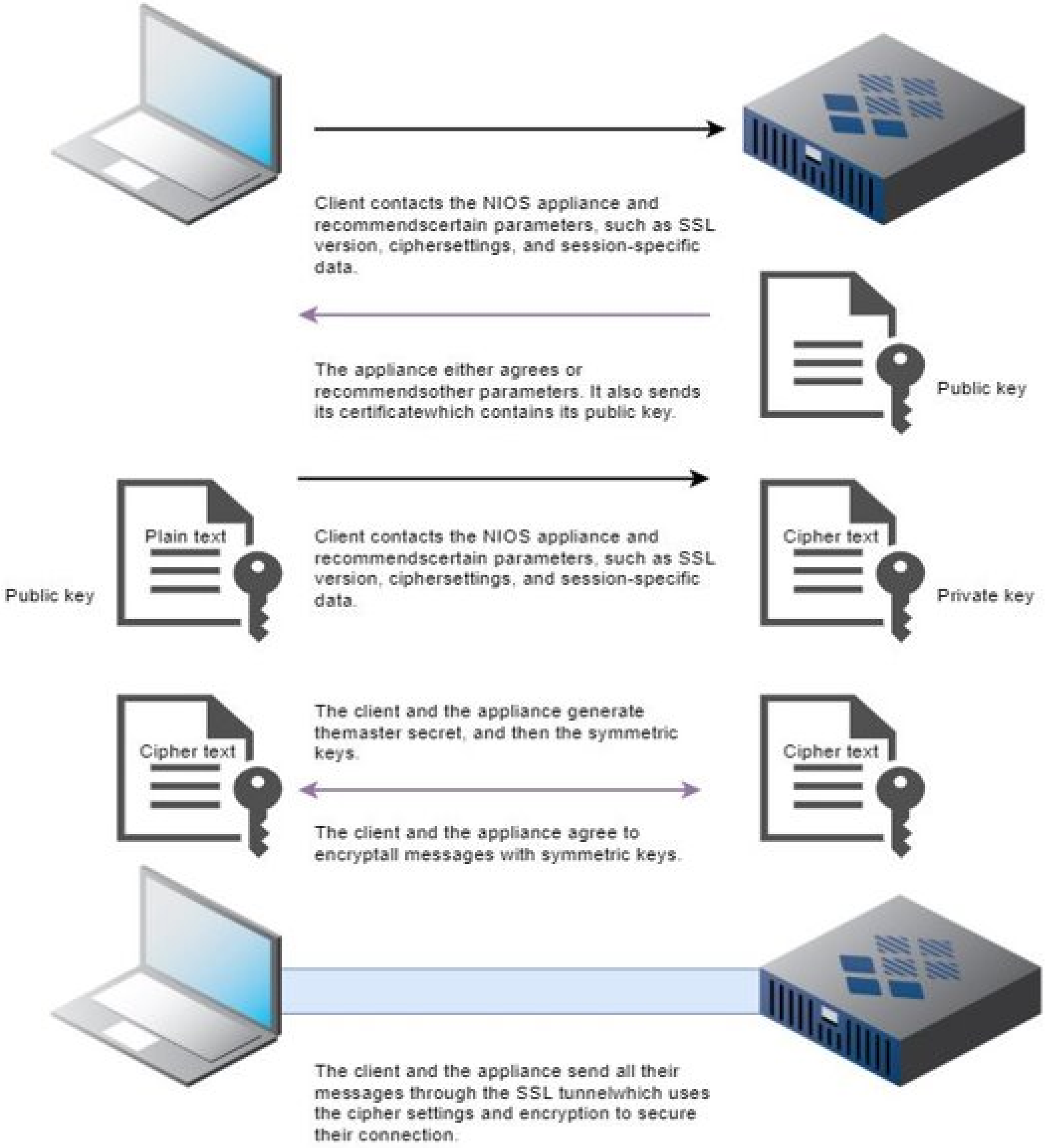
The appliance either agrees or recommendsother parameters. It also sends its certificatewhich contains its public key.

Client contacts the NIOS appliance and recommendscertain parameters, such as SSL version, ciphersettings, and session-specific data.

The client and the appliance generate themaster secret, and then the symmetric keys.

The client and the appliance agree to encryptall messages with symmetric keys.

The client and the appliance send all their messages through the SSL tunnelwhich uses the cipher settings and encryption to secure their connection.


Client contacts the NIOS appliance and recommendscertain parameters, such as SSL version, ciphersettings, and session-specific data.

The appliance either agrees or recommendsother parameters. It also sends its certificatewhich contains its public key.

Client contacts the NIOS appliance and recommendscertain parameters, such as SSL version, ciphersettings, and session-specific data.

The client and the appliance generate themaster secret, and then the symmetric keys.

The client and the appliance agree to encryptall messages with symmetric keys.

The client and the appliance send all their messages through the SSL tunnelwhich uses the cipher settings and encryption to secure their connection.

BENEFITS:

1.Encryption of DNS traffic

2.Securing sensitive information

3.Mitigating DNS spoofing and preventing DNS tampering

4.Standardization and Interoperability

CHALLENGES AND PERFORMANCE CONSIDERATIONS:

1.Encrpytion & Handshake overhead

2.Server load & resource consumption

3.Impact on caching

4.Compatibility and Deployment Challenges

FUTURE TRENDS:

1.Integration with Operating Systems and Browsers.

2.DNS Security Extensions (DNSSEC) Integration

3.Cloud-Based DNS Services