Network Working Group
Internet Draft
expires in six months

A D Keromytis [U-Penn]
W A Simpson [DayDreamer]

September 1997

SPKI: ShrinkWrap
draft-simpson-spki-shrinkwrap-01.txt (A)

Status of this Memo

Abstract

This protocol facilitates the use of Simple Public Key Infrastructure [SPKI] certificate chains with Internet Protocol Security key management protocols.

1. Raison d'etre

Currently proposed session-key management protocols use UDP [RFC-768] for transport. Internet Protocol version 4 [RFC-791] restricts the maximum reassembled datagram to 576 bytes. Internet Protocol version 6 [RFC-1883] restricts the maximum reassembled datagram to 1500 bytes.

Some SPKI certificate chains of delegation could be quite large. Should one of these session-key management protocols need to transmit a lengthy certificate chain, it is quite possible that the protocol will fail.

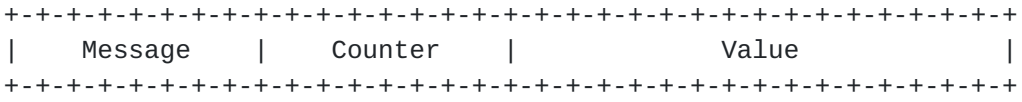
SPKI allows the verifier to reduce a certificate chain to a single certificate. This ShrinkWrap protocol utilizes TCP [RFC-761, RFC-793] to transport long certificate chains, and request a single certificate for subsequent use.

1.1. Terminology

In this document, the key words "MAY", "MUST, "MUST NOT", "optional", "recommended", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [RFC-2119].

byte	An 8-bit quantity; also known as "octet" in standardese.

1.2. Message Header



Message 1 byte. This document defines the following values:

0	reserved
1	Delegation_Certificate
2	Reduction_Request
3	Reduction_Response
11	Resource_Limit
12	Verification_Failure
13	Message_Reject

Counter 1 byte. Aids in matching requests and responses.

The first value sent is 1. Thereafter, the value is monotonically increased for each message sent by the prover.

Value 2 bytes. An optional value field. Although the use of the field is optional, this field is always present to facilitate 32-bit alignment.

The messages may have additional fields beyond the common header, as described later.

1.3. Protocol Overview

The prover is responsible for obtaining any intermediate certificates to complete the delegation chain from the verifier to the target subject. The prover sends the intermediate Delegation_Certificates to the verifier, in the correct order, followed by a Reduction_Request certificate indicating the target subject.

The verification server (usually residing in the same machine as the key management daemon) listens for requests at TCP port 358. The verifier will attempt to do the chain reduction specified in [SPKI], and return a Reduction_Response self-signed certificate (or an error message). The prover checks the response.

More than one reduction can be requested in the same session. The prover sends any additional Delegation_Certificates needed,

interleaved by appropriate Reduction_Request certificates, and collects the Reduction_Responses from the verifier.

When multiple delegation chains exist with different authorizations, the prover sends only those certificates needed to establish the desired delegation chain from the verifier to the target subject. Even when a Delegation_Certificate has been previously sent in a session, it is necessary to repeat the same certificate in the correct order for every Reduction_Request.

When all desired reduced certificates have been obtained, the prover will close the connection.

1.4. Error Recovery

The Counter limits the number of messages that may be sent. A maximum of 254 intermediate delegations are supported in a single delegation chain. Whenever insufficient numbers remain for completion of the delegation chain, the prover MUST close the current connection, open another connection, and send the delegation chain from its beginning.

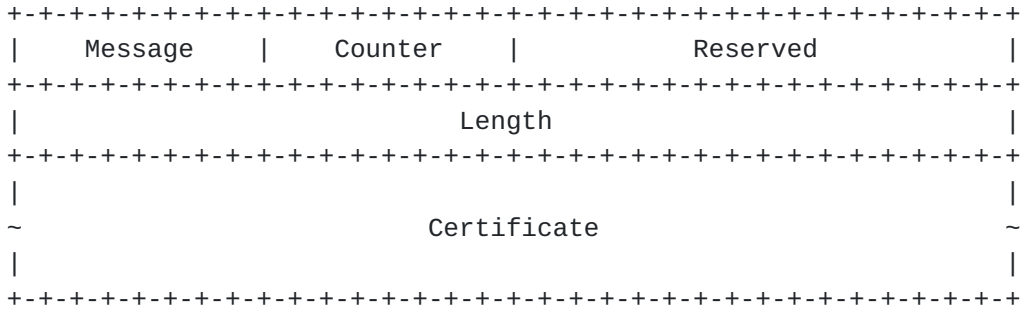
The Counter is required to be monotonically incremented. Whenever an invalid Counter (zero or out of order) is detected, the verifier MUST send a Message_Reject and close the connection.

At any particular time, the verifier may not have sufficient resources available to support reduction of any delegation chain. Whenever insufficient reqources are available, the verifier MUST send a Resource_Limit and close the connection.

The verifier SHOULD set an idle timeout for receiving the next message (default 30 seconds). Following that, the verifier SHOULD close the connection.

2. Data Messages

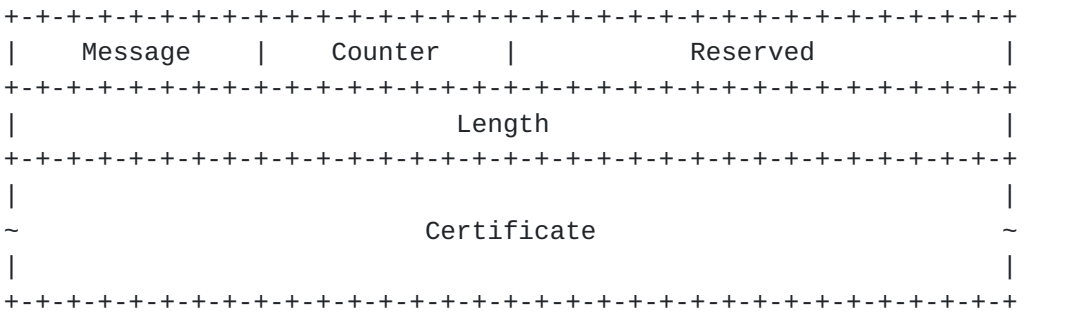
2.1. Delegation_Certificate



Message	1
Counter	1 byte. The value is monotonically increased from the previous message sent.
Reserved	2 bytes. For future use; MUST be set to zero when transmitted, and MUST be ignored when received.
Length	4 bytes. Indicates the number of bytes in the following certificate.
Certificate	The certificate to use for verification.

Any number of Delegation_Certificates may be sent by the prover prior to the Reduction_Request. However, the verifier is not required to devote enough resources to support the maximum of 254 certificates in a single delegation chain. Each verifier will support at least one Delegation_Certificate followed by one Reduction_Request.

2.2. Reduction_Request



Message 2

Counter 1 byte. The value is monotonically increased from the previous (Delegation_Certificate) message sent.

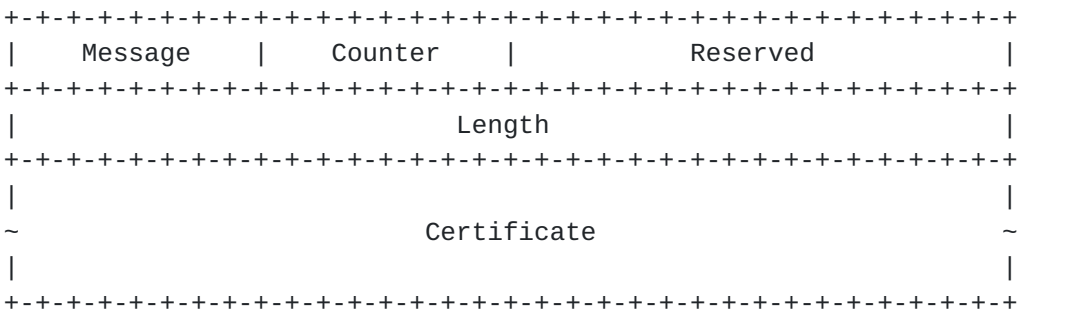
Reserved 2 bytes. For future use; MUST be set to zero when transmitted, and MUST be ignored when received.

Length 4 bytes. Indicates the number of bytes in the following certificate.

Certificate The certificate to be verified and reduced.

Sending the Reduction_Request triggers a Reduction_Response.

2.3. Reduction_Response



Message 3

Counter 1 byte. Copied from the Reduction_Request.

Reserved 2 bytes. For future use; MUST be set to zero when transmitted, and MUST be ignored when received.

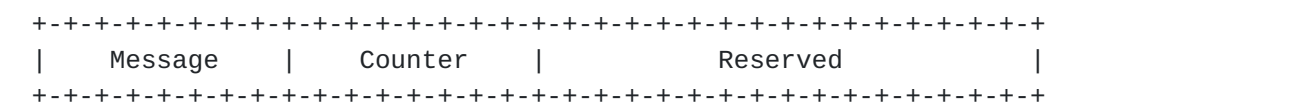
Length 4 bytes. Indicates the number of bytes in the following certificate.

Certificate The result certificate.

Sent by the verifier to fulfill a Reduction_Request.

3. Error Messages

3.1. Resource_Limit



Message 11

Counter 1 byte. Copied from the offending message.

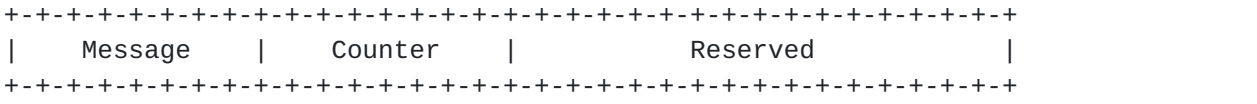
Reserved 2 bytes. For future use; MUST be set to zero when transmitted, and MUST be ignored when received.

This error message is sent by the verifier when some resource is unavailable. The counter indicates the particular Delegation_Certificate for which resources failed.

When too many other reduction sessions are in progress, the counter will indicate the first Delegation_Certificate.

When too many certificates are included in a single transaction, the counter will indicate the unacceptable Delegation_Certificate. If the same reduction is attempted at a later time, the prover SHOULD divide the reduction into multiple transaction sessions, each with fewer delegation certificates.

3.2. Verification_Failure



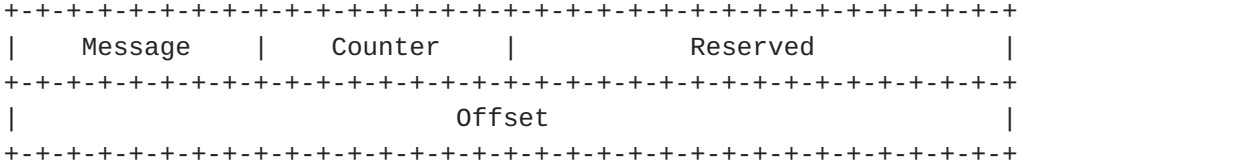
Message 12

Counter 1 byte. Copied from the offending message.

Reserved 2 bytes. For future use; MUST be set to zero when transmitted, and MUST be ignored when received.

This error message is sent by the verifier when unable to fulfill a Reduction_Request. The counter indicates the particular certificate for which verification failed.

3.3. Message_Reject



Message 13

Counter 1 byte. Copied from the offending message.

Reserved 2 bytes. For future use; MUST be set to zero when transmitted, and MUST be ignored when received.

Offset 4 bytes. The number of bytes from the beginning of the offending message where the unrecognized field starts.

(0) indicates a bad Message number.
(1) indicates an invalid Counter.

Note that the Offset is 8 greater than a corresponding Certificate Length.

This error message is sent by the verifier to indicate an unrecognized message or certificate format, and when any single message is too large to process.

Security Considerations

These messages are likely to be used prior to establishing a security association between the parties. Thus, the messages rely upon the TCP synchronization handshake, and the security of the certificates themselves, to protect against attacks.

There are several opportunities for Denial of Service attacks. The simplest is to swamp the verifier with certificates, exhausting the processing resources during verification. The TCP handshake assists in detecting the source of such attacks. +

Delegation_Certificates SHOULD NOT be verified until the Reduction_Request is received, preventing an indefinite stream of bogus certificates. Only those certificates necessary for fulfilling the + Reduction_Request are examined and verified. Caching of active certificates will mitigate repetitive requests.

An eavesdropper can insert valid TCP sequence numbers with invalid data. This invalid data will be detected by the recipient during certificate verification, but the other party will be locked out of the TCP session. The receipt of TCP acknowledgments beyond the data sent MUST cause a reset of the TCP connection.

Contacts

Comments about this document should be discussed on the spki@c2.net mailing list.

Questions about this document can also be directed to:

Angelos D. Keromytis
Distributed Systems Lab
Computer and Information Science Department
University of Pennsylvania
200 South 33rd Street
Philadelphia, Pennsylvania 19104-6389

angelos@adk.gr
angelos@dsl.cis.upenn.edu

William Allen Simpson
DayDreamer
Computer Systems Consulting Services
1384 Fontaine
Madison Heights, Michigan 48071-4818

wsimpson@UMich.edu
wsimpson@GreenDragon.com (preferred)
bsimpson@MorningStar.com