

Use of IKE Fragmentation for Large Messages

`draft-smyslov-ipsecme-ikev2-fragm-large-msg-01`

Valery Smyslov (svan@elvis.ru)

IETF 125

Large IKE Messages

- What is “large”?
 1. 2-3 Kbytes? (we thought so when developed IKE fragmentation)
 - IKE fragmentation handles this well
 2. 20-60 Kbytes? (some PQ KEMs and SIG algorithms)
 - still fits into IKE payloads
 - we believe IKE fragmentation still works well (with some definition of “well”)
 3. Several hundred Kbytes? As a notable KEM public key...
 - new IKE payload format is needed
 - we have **good candidate** - [draft-nir-ipsecme-big-payload!](#)
 - **possible issues with large number of fragments?**

Experiment

- Test setup
 - 2 desktops connected over Ethernet LAN
 - 1Gbit / 100Mbit
 - fragment size 576 bytes for UDP
 - configured KE methods
 - KE X25519
 - AKE1 ML-KEM1024
 - AKE2 HQC256
 - AKE3 FrodoKEM1344 AES
 - AKE4 Classic McEliece 348864
 - AKE5 Classic McEliece 460896
 - AKE6 Classic McEliece 6688128
 - AKE7 Classic McEliece 8192128

Experiment Data 1

(TCP, 1Gbit)

- SA created in 21-27 sec

KE Method	IKE msg size, bytes	Completed in, sec	Fragments
X25519	440 443	< 1	-
ML-KEM1024	1648 1648	< 1	-
HQC256	7328 14496	< 1	-
FrodoKEM1344 AES	21600 21712	1	-
Classic McEliece 348864	261202 208	2-3	5
Classic McEliece 460896	524242 272	4-5	9
Classic McEliece 6688128	1045074 320	6-10	17
Classic McEliece 8192128	1357906 320	7-16	21

Experiment Data 2

(TCP, 100Mbit)

- SA created in 28-39 sec

KE Method	IKE msg size, bytes	Completed in, sec	Fragments
X25519	440 443	< 1	-
ML-KEM1024	1648 1648	< 1	-
HQC256	7328 14496	< 1	-
FrodoKEM1344 AES	21600 21712	1	-
Classic McEliece 348864	261202 208	2-3	5
Classic McEliece 460896	524242 272	6-17	9
Classic McEliece 6688128	1045074 320	8-11	17
Classic McEliece 8192128	1357906 320	7-20	21

Experiment Data 3

(UDP, 1Gbit)

- SA created in 26-36 sec

KE Method	IKE msg size, bytes	Completed in, sec	Fragments	Retransmitted fragments (max)
X25519	416 427	< 1	-	-
ML-KEM1024	1648 1648	< 1	4 4	0 0
HQC256	7328 14496	1	16 32	16 32
FrodoKEM1344 AES	21600 21712	1	47 47	47 47
Classic McEliece 348864	261202 208	2	564 -	564 -
Classic McEliece 460896	524242 272	5-9	1133 -	1133 -
Classic McEliece 6688128	1045074 320	7-10	2258 -	4516 -
Classic McEliece 8192128	1357906 320	8-18	2933 -	8799 -

Experiment Data 4

(UDP, 100Mbit)

- SA created in 36-56 sec

KE Method	IKE msg size, bytes	Completed in, sec	Fragments	Retransmitted fragments (max)
X25519	416 427	< 1	-	-
ML-KEM1024	1648 1648	< 1	4 4	0 0
HQC256	7328 14496	1	16 32	16 32
FrodoKEM1344 AES	21600 21712	1	47 47	47 47
Classic McEliece 348864	261202 208	1-4	564 -	564 -
Classic McEliece 460896	524242 272	5-10	1133 -	1133 -
Classic McEliece 6688128	1045074 320	10-34	2258 -	6774 -
Classic McEliece 8192128	1357906 320	14-18	2933 -	8799 -

Problem with IKE Fragmentation

- In case any fragment is lost, all message fragments are retransmitted
 - not a big deal if there are few fragments (as in case 1)
 - a bigger problem for cases 2 and (especially) 3
 - network is being filled with a garbage traffic
 - receiver has to process and discard all already received fragments
 - possible congestion on the receiver

Proposed Solution

- Make it possible for the receiver of the message to send back an info about the received (or missing) fragments, so that only missing fragments are retransmitted
 - the idea was thought out when IKE fragmentation was being developed, but was considered as unnecessary complication at that time
 - two proposals that differ in details
 1. IKEv2 Fragment Acknowledgment Extension ([draft-antony-ipsecme-ikev2-fragment-acknowledgment](#))
 2. Using IKE Fragmentation for Large Messages ([draft-smyslov-ipsecme-ikev2-fragm-large-msg](#))

Proposals Overview

- Receiver of fragmented message sends receipt status report to the sender if not all fragments are received, indicating which fragments are missing
- Sender retransmits only missing fragments
- This continues until all fragments are received or sender gives up

Proposals Comparison

- Receipt status reports are sent
 1. inside Notify payload in non-fragmented IKE message
 2. inside a specially crafted message fragment (with both `Fragment Number` and `Total Fragments` set to 65535 and with a protocol hack that makes such fragment invalid for unsupported receiver)
- Missing fragments are reported as
 1. array of fragment numbers
 2. bitmap
- Extension is
 1. negotiated
 2. works w/o negotiation (but it is possible to add negotiation too; that could slightly improve performance and avoid the protocol hack)

Experiments with Proposal 2

- PoC implementation of [draft-smyslov-ipsecme-ikev2-fragm-large-msg](#)
- The same test setup

Experiment Data 5

(Selective Resend, UDP, 1Gbit)

- SA created in 27-32 sec

KE Method	IKE msg size, bytes	Completed in, sec	Fragments	Retransmitted fragments (max)	Retransmitted fragments (max) w/o selective resend
X25519	416 427	< 1	-	-	-
ML-KEM1024	1648 1648	< 1	4 4	0 0	0 0
HQC256	7328 14496	1	16 32	0 0	16 32
FrodoKEM1344 AES	21600 21712	1	47 47	0 0	47 47
Classic McEliece 348864	261202 208	1-3	564 -	43 -	564 -
Classic McEliece 460896	524242 272	5-8	1133 -	357 -	1133 -
Classic McEliece 6688128	1045074 320	9-17	2258 -	2043 -	4516 -
Classic McEliece 8192128	1357906 320	8-10	2933 -	3152 -	8799 -

Experiment Data 6

(Selective Resend, UDP, 100Mbit)

- SA created in 25-46 sec

KE Method	IKE msg size, bytes	Completed in, sec	Fragments	Retransmitted fragments (max)	Retransmitted fragments (max) w/o selective resend
X25519	416 427	< 1	-	-	-
ML-KEM1024	1648 1648	< 1	4 4	0 0	0 0
HQC256	7328 14496	1	16 32	2 10	16 32
FrodoKEM1344 AES	21600 21712	1	47 47	0 0	47 47
Classic McEliece 348864	261202 208	1-3	564 -	331 -	564 -
Classic McEliece 460896	524242 272	5-8	1133 -	912 -	1133 -
Classic McEliece 6688128	1045074 320	8-21	2258 -	1218 -	6774 -
Classic McEliece 8192128	1357906 320	8-17	2933 -	3637 -	8799 -

Thanks!

Comments?

Is it needed?

WG adoption?