# draft-ietf-6tisch-minimal-security


- Authors: Mališa Vučinić (Ed.)
  Jonathan Simon
  Kris Pister
  Michael Richardson

# Status

- Published -09 after Bangkok
- 2nd WGLC call ended, comments from
    - Göran Selander
- IESG approved OSCORE!
- Goal of the presentation
    - Quick summary of changes in -09
    - Discuss 2nd WGLC comments from Göran Selander

# Updates in -09

- Remove network_identifier from Join Response
- Add join_rate

```
 These values may be configured to values specific to the deployment.
 The default values have been chosen to accommodate a wide range of deployments, taking into account dense networks.
-Increased values of NSTART and PROBING_RATE at the JP enable multiple pledges (approximately 3 pledges by default) to concurrently join through the same JP.
-Following {{RFC7252}}, the average data rate in sending to JP or JRC must not exceed PROBING_RATE.
+
+The PROBING_RATE value at the JP is controlled by the join rate parameter, see {{configuration_object}}.
+Following {{RFC7252}}, the average data rate in sending to the JRC must not exceed PROBING_RATE.
```

- Clarify usage of blacklist

```
-When present, the blacklist parameter MUST contain at least one pledge identifier.
-When the joined node receives this parameter, it MUST silently drop any link-layer frames originating from the indicated pledge identifiers.
+When present, the array MUST contain zero or more byte strings encoding pledge identifiers.
+The joined node MUST silently drop any link-layer frames originating from the pledge identifiers enclosed in the blacklist parameter.
+When this parameter is received, its value MUST overwrite any previously set values.
 This parameter allows the JRC to configure the node acting as a JP to filter out traffic from misconfigured or malicious pledges before their traffic is for
+If the JRC decides to remove a given pledge identifier from a blacklist, it omits the pledge identifier in the blacklist parameter value it sends next.
```

# 2nd WGLC Comments from Göran Selander 1/3

Implementations MUST ensure that multiple CoAP requests to different JRCs are properly incrementing the sequence numbers.

- Göran Selander: *I don't know why this is restricted to different JRCs*

**PROPOSED RESOLUTION:** Implementations MUST ensure that multiple CoAP requests, including to different JRCs, are properly incrementing the sequence numbers.

The (6LBR) pledge and the JRC use the OSCORE security context parameters (e.g. sender and recipient identifiers) as they were used at the moment of context derivation, regardless of whether they currently act as a CoAP client or a CoAP server.

- Göran Selander: I think this text is intended as a clarification of OSCORE functionality, but it doesn't make clear that that is what it is.

**PROPOSED RESOLUTION:** Note that when the (6LBR) pledge and JRC change roles between CoAP client and CoAP server, the same OSCORE security context as initially derived in each endpoint remains in use and the derived parameters are unchanged, for example Sender ID when sending and Recipient ID when receiving, see section 3.1 of [OSCORE].

# 2nd WGLC Comments from Göran Selander 2/3

A technique that prevents reuse of sequence numbers, detailed in Section 7.5.1 of [I-D.ietf-core-object-security], MUST be implemented. Each update of the OSCORE Replay Window MUST be written to persistent memory.

- Göran Selander: Section 7.5.1 is now B.1.1, and the procedure is changed, there are two parameters to set, K and F.

**PROPOSED RESOLUTION:** Update the reference.

# 2nd WGLC Comments from Göran Selander 3/3

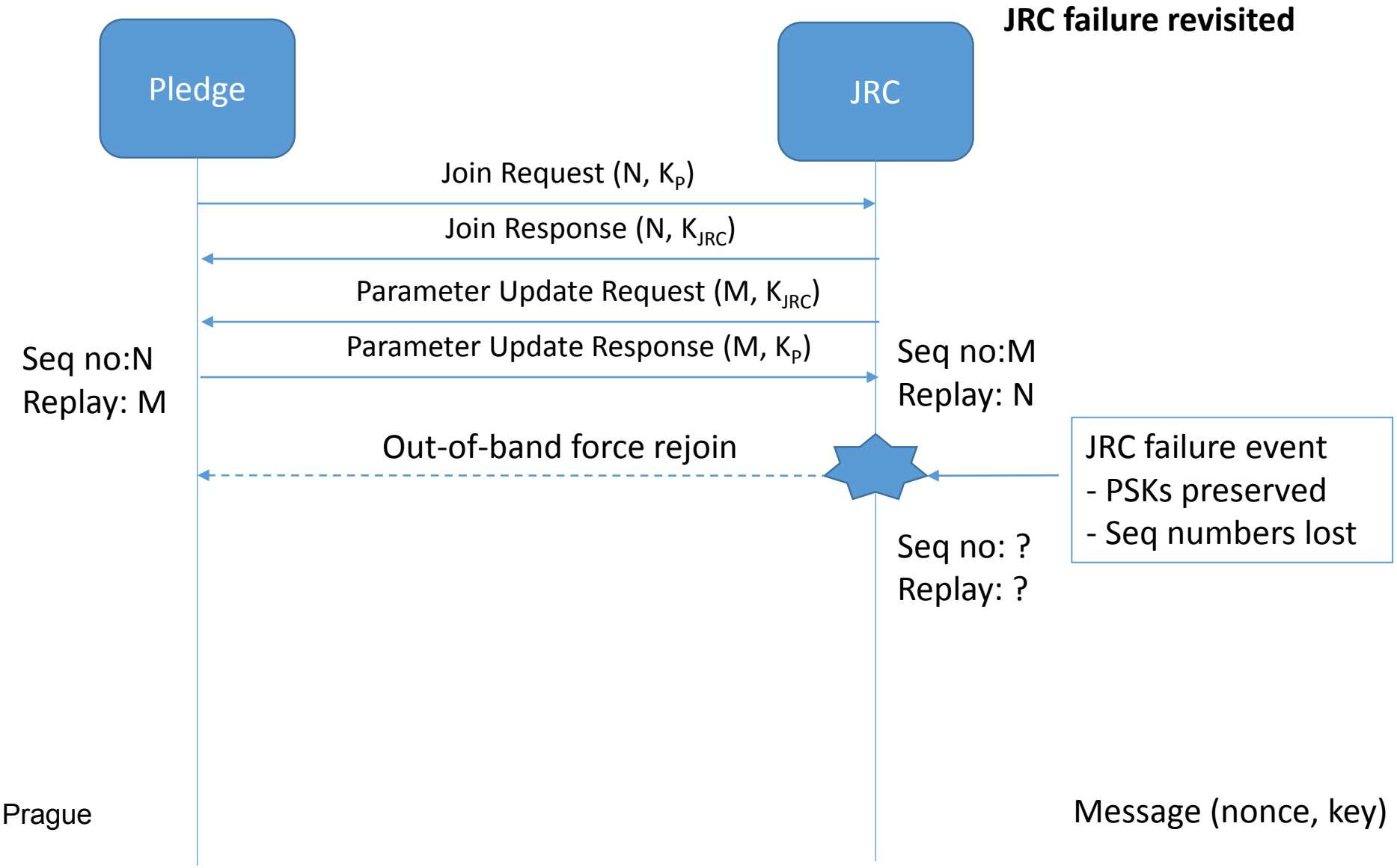
**JRC failure revisited**

- Loss of sequence numbers at JRC
- Nodes forced to rejoin
- Nonce reuse during first Parameter Update Exchange triggered by JRC
  - Aware of the situation, JRC includes a random payload in the first request

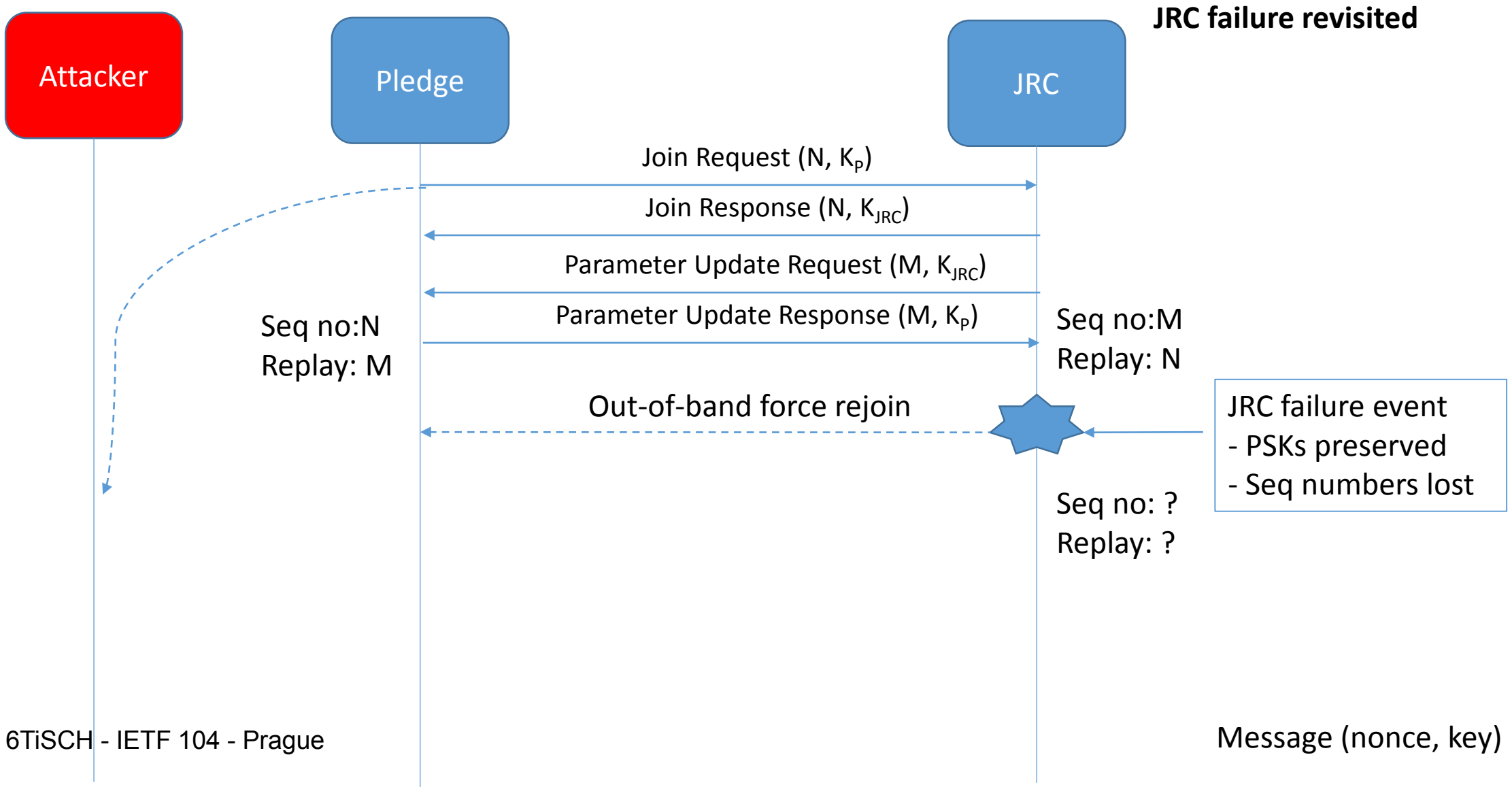
- Göran Selander: I assume the replaced JRC is involved in this rejoin. What prevents a replay attack of an old rejoin? Would that not lead to a replay of the nonce in the CoJP response? Are you sending random in the CoJP response also?

- Göran Selander: If there is always an extra roundtrip in the case of complete failure, then perhaps the new procedure in OSCORE B.2 could be used

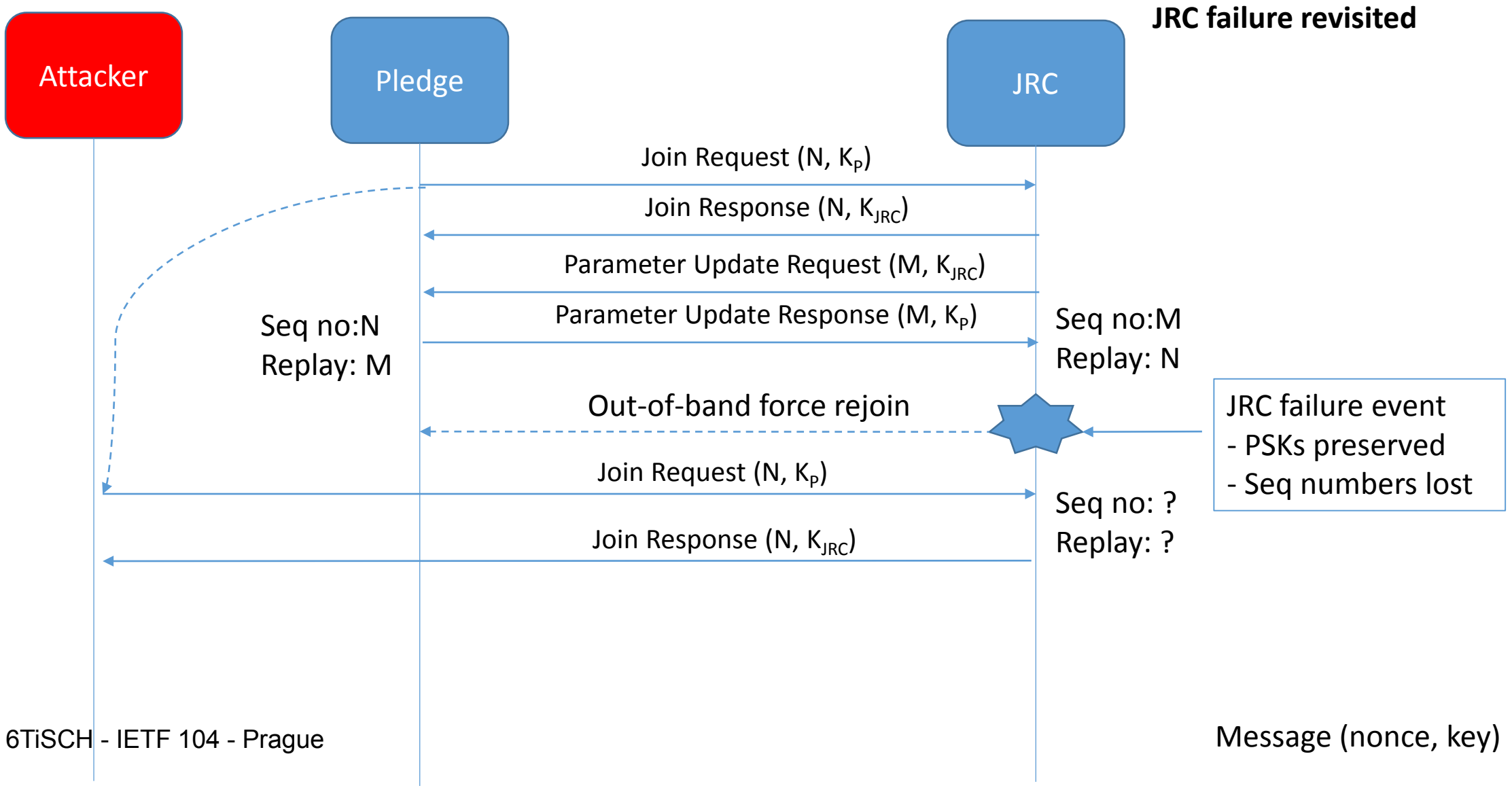
# 2nd WGLC Comments from Göran Selander 3/3



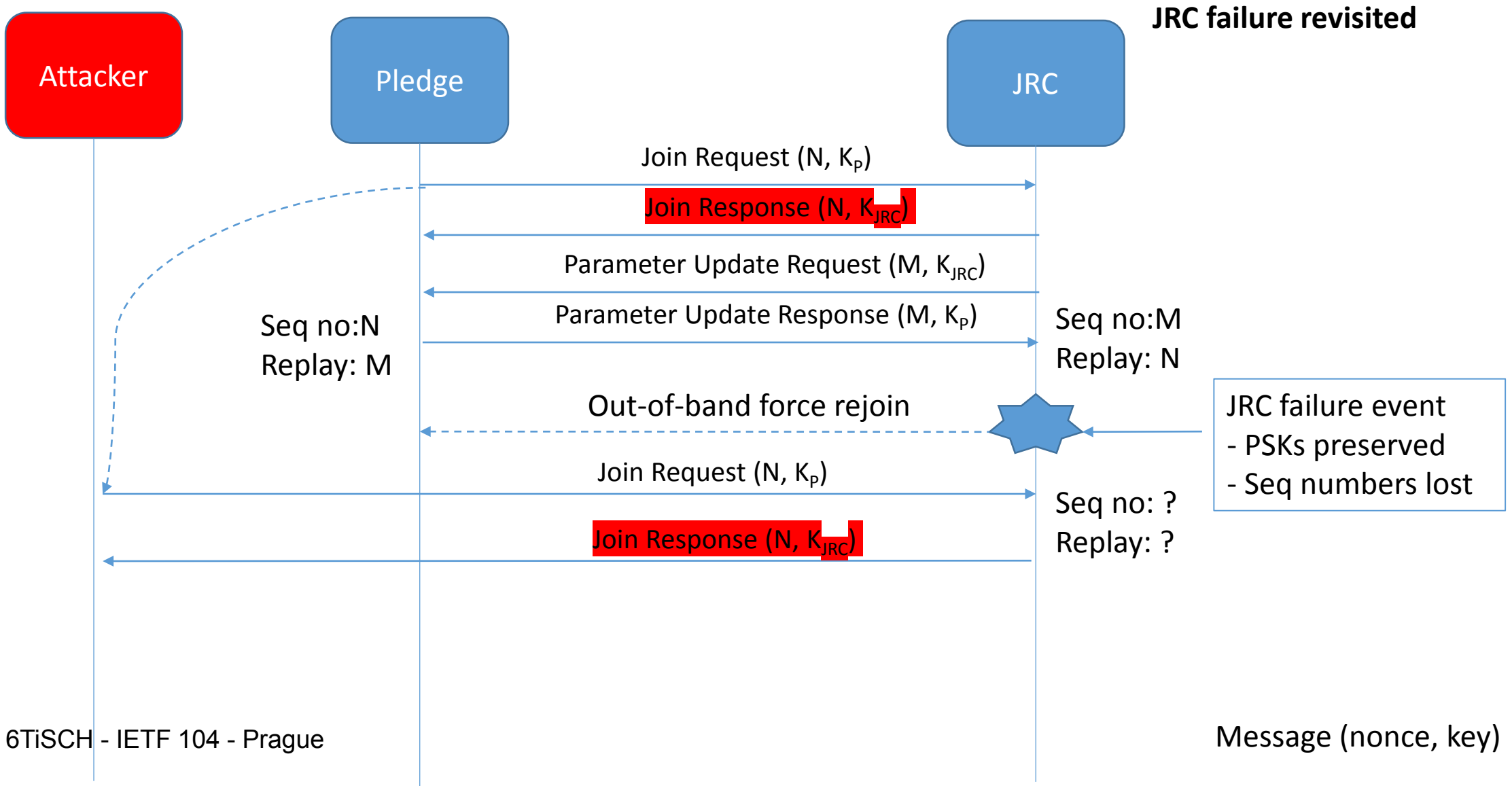
# 2nd WGLC Comments from Göran Selander 3/3



# 2nd WGLC Comments from Göran Selander 3/3



# 2nd WGLC Comments from Göran Selander 3/3



# Possible ways forward

- Challenge-response based on OSCORE Appendix B.2
  - Derivation of a new context based on the old PSK and exchanged nonces
  - Additional round-trip only in the JRC failure case
  - Not compatible with Lightweight Implementation Option in Appendix B of minimal-security as it requires HKDF in firmware
- Design of a custom sequence number synchronization mechanism
  - Additional round-trip only in the JRC failure case
  - Compatible with Lightweight Implementation Option, HKDF can be invoked during provisioning
  - 6TiSCH specific
  - Not fully baked, additional considerations and security review needed
- Require mutable (sequence numbers) and immutable (PSKs) parameters to be stored at all times together
  - Text in Security Considerations