

Announcing Supported Authentication Methods in IKEv2

`draft-smyslov-ipsecme-ikev2-auth-announce`

Valery Smyslov
svan@elvis.ru

IETF 108

Authentication in IKEv2

- Unlike IKEv1, authentication method in IKEv2 is not negotiated, each peer is free to use whichever method he thinks is appropriate
- Generally works well if there is only one way of doing authentication or there is no ambiguity in choosing among several of them
- If peers can use several methods to authenticate each other, it is possible that initiator selects authentication method unsupported by the responder
 - less likely in the opposite direction, but still possible

The Problem

- The problem was first encountered when RSA-PSS signature format appeared in IKEv2
 - newer initiators tried to use PSS signatures while older responders didn't support it, sending back **AUTHENTICATION_FAILED**
 - if initiators knew responders' capabilities they would have chosen PKCS#1 and the SA succeeded

Source of the Problem

- Currently there is no way for the peers to explicitly indicate the supported authentication methods
 - it is possible to guess them via indirect means, e.g. `CERTREQ` content, but this is unreliable
- With new signature formats and authentication methods appearing in the future (including PQ and hybrid ones) the situation of mis-selecting may happen more often

Proposed Solution

- Add new optional notification `SUPPORTED_AUTH_METHODS` to indicate the supported authentication methods
 - for certificate-based authentication add an ability for the peers to indicate which signing algorithms can be used with each of CA in the `CERTREQ` payload
 - avoid creating new IANA registries

SUPPORTED_AUTH_METHODS

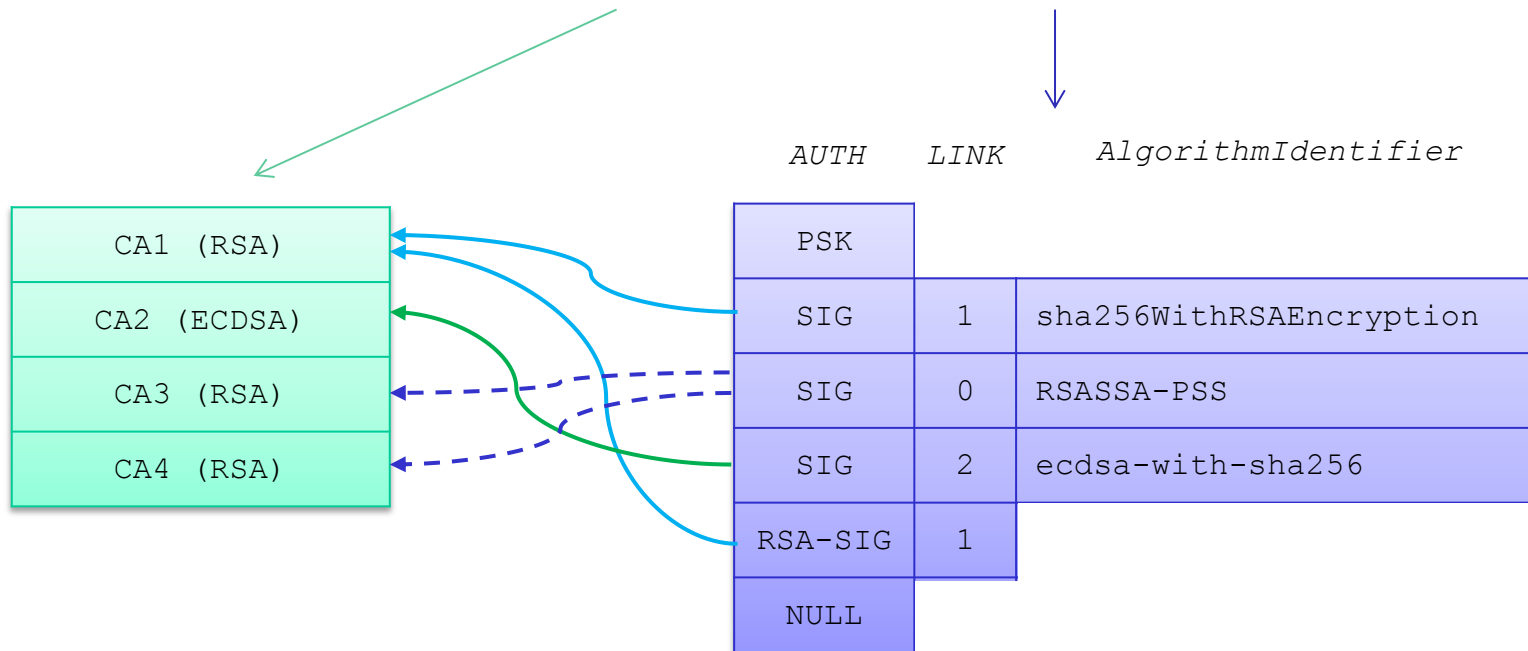
Notification Format

- Notification data consists of a list of supported authentication methods in the following formats:
 1. Two-octet format for the methods that are not linked to `CERTREQ` payload (`PSK`, `NULL`)
 2. Three-octet format that allows optional linking to `CERTREQ` payload (`RSA-SIG` etc.)
 3. Multi-octet format that allows optional linking to `CERTREQ` payload and specifying `ASN.1 AlgorithmIdentifier` for use with particular CA (`SIG`)
- The linking to CAs is done by specifying the CA number within the `CERTREQ` payload the method can be used with

SUPPORTED_AUTH_METHODS

Notification Format Illustration

HDR, SAr1, KEr, Nr, CERTREQ, N(SUPPORTED_AUTH_METHODS)



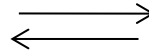
Exchanges (Option 1)

Initiator

Responder

IKE_SA_INIT

HDR, SAI1, KEi, Ni

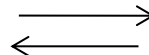


IKE_SA_INIT

HDR, SAR1, KEr, Nr, [CERTREQ,]
[N(SUPPORTED_AUTH_METHODS) (...)]

IKE_AUTH

HDR, SK{IDi, [CERT,] [CERTREQ,]
[IDr,] AUTH, SAI2, TSi, TSr,
[N(SUPPORTED_AUTH_METHODS) (...)]}



IKE_AUTH

HDR, SK{IDr, [CERT,]
AUTH, SAI2, TSi, TSr}

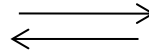
Exchanges (Option 2)

Initiator

Responder

IKE_SA_INIT

HDR, SAi1, KEi, Ni

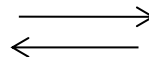


IKE_SA_INIT

HDR, SAR1, KEr, Nr, [CERTREQ,]
[N(SUPPORTED_AUTH_METHODS)]

IKE_INTERMEDIATE

HDR, SK{...}

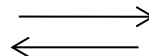


IKE_INTERMEDIATE

HDR, SK{...,
N(SUPPORTED_AUTH_METHODS) (...)}
N(SUPPORTED_AUTH_METHODS) (...)}

IKE_AUTH

HDR, SK{IDi, [CERT,] [CERTREQ,]
[IDr,] AUTH, SAi2, TSi, TSr,
[N(SUPPORTED_AUTH_METHODS) (...)]}



IKE_AUTH

HDR, SK{IDr, [CERT,]
AUTH, SAi2, TSi, TSr}

Thanks

- Comments? Questions?
- More details in the draft
- WG adoption?