New Safe Curves for IKEv2 Key Agreement

draft-nir-ipsecme-curve25519-01

Yoav Nir - IETF 93

Safe Curves

- In recent years there have been some concerns raised about the so-called “NIST curves”:
  - They have arbitrary-looking parameters
  - It is very hard to write a constant-time implementation
  - They are considerably slower than the state of the art.

Safe Curves

- CFRG has a document in the works for new curves: draft-irtf-cfrg-curves-02
- It defines two curves:
  - Curve25519
  - Curve448 (“Goldilocks”)
- Both are fast, easy to get right, and a rigorous process was used to obtain the parameters.
- Curve25519 is already in wide use in SSH and PGP.

Speed

cycles:	Intel generate	Intel derive	ARM generate	ARM derive
P-256	284,040	706,068	1,367,220	4,647,923
Curve25519	169,920	161,648	424,255	410,128
P-384	~2,200,000 ?	~2,500,000 ?	~5,000,000 ?	~15,000,000 ?
Curve448	177,432	532,056	679,519	1,698,441

Source: http://bench.cr.yp.to
“Intel” = Xeon E3-1275 V3. “ARM” = 2011 Qualcomm Snapdragon S3

This draft

- Defines the use of Curve25519 and Curve448 for use in IKEv2.
- Similar to draft-ietf-tls-curve25519.
- Short document: 4 pages. 11 including boilerplate, references, and appendix with arithmetic.
- Please read, please comment, please adopt

What about signatures?

- The safe curves can be used for signatures as well as key agreement - just like the NIST curves.
- CFRG are working on a signature scheme right now, and there is a candidate TLS draft.
- Do we need one?
- I don’t think so. Once they assign an OID, we can use RFC 7427 for signatures.