(challenge,response)
Schnorr

Mike Hamburg

Schnorr signing, roughly

Let G generate a group of prime order q

Secret key is x, public key P = xG

Choose (pseudo)random nonce n, ephemeral E = nG

Challenge c = hash(E, message)

Response r = n - cx mod q

Then E = rG + cP

Ed25519-style Schnorr

Signature is (E,r)

Check E == rG + Hash(E, msg)*P

Benefit: can batch verification

Benefit: more natural in eg BLINKER

Good for smartphones and servers

Old-style Schnorr

Signature is (c,r)

Check Hash(rG+cP, msg) == c

Benefit: c can be half-length —> shorter sigs

Benefit: can do EC first, hash later

Benefit: don’t have to parse untrusted points

Good for deeply embedded devices, IoT

Bells and whistles

Hashing separated by setting and curve perso

Propose fixed-length for less code

Setting can include “msg is actually H(msg)”

Hashing includes public key

Sigs are derandomized with a PRF (req’d by CFRG)

Encode everything in fixed length (no ASN.1)!

That’s all

Questions?