MTI signature schemes for small devices

stephen.farrell@cs.tcd.ie

CFRG @ IETF-113
March 2022

Background

- LAKE WG defining EDHOC protocol for authentication and key establishment for “small” devices
  - https://datatracker.ietf.org/doc/draft-ietf-lake-edhoc/
  - Think of that as like-TLS but e.g., where octets-sent is a far higher priority than with TLS due to small MTU sizes – reqs draft: https://datatracker.ietf.org/doc/draft-ietf-lake-reqs/
- EDHOC has “ciphersuites”, also similar to TLS, that involve a signature algorithm so the question arises as to which, if any, signature algorithms ought be mandatory to implement (MTI) for EDHOC
  - Suites are defined that involve ECDSA (p256, p384) and EdDSA (25519, 448)
- Picking MTI algorithms always has the potential to be contentious, for well-known reasons, (we don’t need an MTI, I prefer that one, etc.) and EDHOC is no exception, but…

Attack Context

- An additional argument arises in this context where an adversary controls a provisioned device and mounts e.g. fault injection attacks to extract a signing key
  - Thanks to Rene Struik for raising this, most recently in the thread at https://mailarchive.ietf.org/arch/msg/lake/0-9X6McGoQCoqZ0q2v97eH9i4iY/
- Context here includes small, relatively inexpensive, commercial devices, so private key may not necessarily be that well protected
  - But I assume that simply reading the private key from storage via JTAG or similar isn’t trivial, so the adversary has to do more
- Private key extraction could enable various significant attacks, e.g. masquerade as controller to actuator

Some relevant publications I’ve found

- This isn’t my area but I wondered if any signature scheme was really that much better than any other in this respect...
- Barenghi et al, “Low Voltage Fault Attacks to AES and RSA on General Purpose Processors” , 2010
  - IMO nicely explains how undervoltage can lead to single bit flips in memory reads and how that can lead to leaking a private key
  - https://eprint.iacr.org/2010/130.pdf
- Barenghi et al, “A Novel Fault Attack Against ECDSA”, 2011
  - Not-that-different attack (to the above) on ECDSA
  - https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1091.528&rep=rep1&type=pdf
- Samwel et al., “Breaking Ed25519 in WolfSSL”, 2017
  - Power analysis attack on EdDSA
  - https://eprint.iacr.org/2017/985.pdf

Signature determinism

- EdDSA is deterministic, as can be ECDSA ala RFC 6979
  - That may be considered to help this adversary, leading to a conclusion that ECDSA is “better” as part of an MTI ciphersuite
  - ...but see above pubs, and e.g. Sony “oops” from 2010 https://www.schneier.com/blog/archives/2011/01/sony_ps3_securi.html
- There have been suggestions to add “noise” to signature schemes that are otherwise like EdDSA or RFC 6979
  - https://datatracker.ietf.org/doc/draft-mattsson-cfrg-det-sigs-with-noise/
- I’m not expressing a personal opinion on any of the above (as I’m mostly not qualified:-)

An “ask” to CFRG...

- I think this boils down to two questions to CFRG:
  1) Which of RSA/ECDSA/EdDSA are ok as part of an MTI ciphersuite in such contexts?
  2) Can CFRG recommend/develop anything better than all of the above?
- I don’t expect the LAKE WG wants to wait for formal answers, if they’re not available “soon”, but will ask, after this CFRG discussion
- I do expect LAKE won’t be the only IETF WG considering this topic, and guess a number of non-IETF activities and implementors might be interested if there’re rough-consensus CFRG answers here
- Moar background – a CFRG list thread: https://mailarchive.ietf.org/arch/msg/cfrg/Ev8hgyojKeObXMZ7SF2m3_yekMo/

Summary

- Adversary controls a provisioned device and mounts e.g. fault injection attacks to extract a signing key
- Context includes small, relatively inexpensive, commercial devices, so private key may not be that well protected but reading keys from storage isn't trivial

1) Which of RSA/ECDSA/EdDSA are ok as part of an MTI ciphersuite in such contexts?
2) Can CFRG recommend/develop anything better than all of the above?