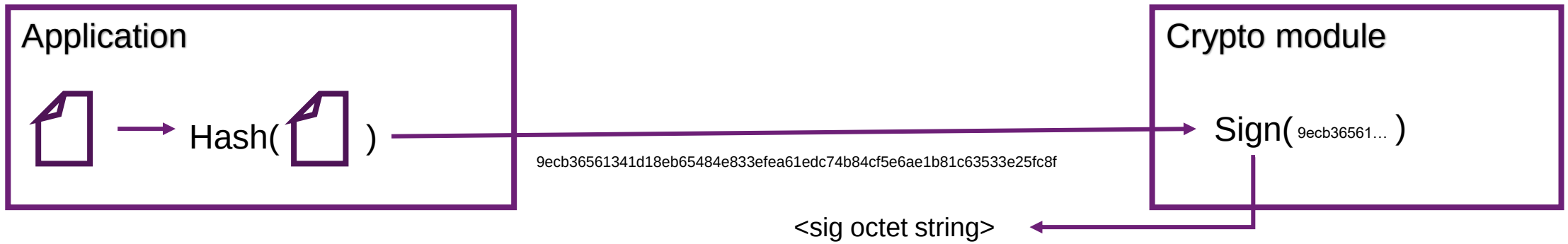
PQ HASH-THEN-SIGN ISSUES

Mike Ounsworth

IETF LAMPS INTERIM - September 2022

HASH-THEN-SIGN TODAY – RSA / ECDSA



Examples:

- Application -> Network HSM
- Golang x509/x509.go -> crypto/ecdsa.go
- Smartcard reader -> smartcard
- etc

ENTRUST

PROBLEM 1

- SPHINCS+, Dilithium, Falcon all have a first step of hashing the message with a nonce – either random, or deterministic from the pub key:

SPHINCS+

```
spx_sign(M, SK){
    // init
    ADRS = toByte(0, 32);

    // generate randomizer
    opt = toByte(0, n);
    if(RANDOMIZE){
      opt = rand(n);
    }
    R = PRF_msg(SK.prf, opt, M);
    SIG = SIG || R;


    digest = H_msg(R, PK.seed, PK.root, M);
    tmp_md = first floor((ka +7)/ 8) bytes of digest;
    tmp_idx_tree = next floor((h - h/d +7)/ 8) bytes of digest;
    tmp_idx_leaf = next floor((h/d +7)/ 8) bytes of digest;
```

Dilithium

Sign(sk, M)

09 $\mathbf{A} \in R_q^{k \times \ell} := \mathsf{ExpandA}(\rho)$ ▷ $\mathbf{A}$ is generated and stored in NTT Representation as $\hat{\mathbf{A}}$

10 $\mu \in \{0,1\}^{384} := \mathsf{CRH}(tr \parallel M)$

11 $\kappa := 0$, $(\mathbf{z}, \mathbf{h}) := \perp$

12 $\rho' \in \{0,1\}^{384} := \mathsf{CRH}(K \parallel \mu)$ (or $\rho' \leftarrow \{0,1\}^{384}$ for randomized signing)

Falcon

Algorithm 10 Sign (m, sk, $\lfloor \beta^2 \rfloor$)

Require: A message m, a secret key sk, a bound $\lfloor \beta^2 \rfloor$

Ensure: A signature sig of m

1: r $\leftarrow \{0,1\}^{320}$ uniformly

2: $c \leftarrow$ HashToPoint(r‖m, q, n)

ENTRUST

PROBLEM 2

- Naïvely doing .Sign(HASH(m)) instead of .Sign(m) does not impact the non-deterministic signature properties, but introduces a dependence on the collision resistance property of HASH.

- **Question 1: Do we care?**
- **Question 2: If yes, can we safely externalize the hash step of SPHINCS+, Dilithium, Falcon outside the crypto module?**
  - For deterministic Dilithium & SPHINCS+ (where pre-hash is with pubkey) the answer seems to be "maybe".
  - For SPHINCS+, randomized Dilithium, Falcon, the answer seems to be "probably not" b/c of its reliance on an RNG.
- **Question 3: Can we pre-hash with a randomized hash such as .Sign(r || HASH(r || m)) ?**

ENTRUST