Composite Crypto Updates & Hackaton Report

Presentation:

Massimiliano Pala
CableLabs and OpenCA Labs

Joint Work:

Mike Ounsworth (Entrust), John Gray (Entrust), Serge Mister (Entrust),
Scott Fluhrer (Cisco), Jan Klaußner (D-Trust), Klaus-Dieter Wirth (D-Trust),
Tim Hollebeek (DigiCert), Corey Bonnell (DigiCert)

Composite Crypto Summary

- Composite Crypto provides a mechanism to combine keys and signatures that use different cryptographic algorithms
- Composite Crypto defines new OIDs for Keys and Signatures and associated data structures definitions that extend them to SEQUENCEs of the structures we already support (e.g., `subjectPublicKeyInfo`)
- Explicit combinations of algorithms have been defined to guide PKI architects in their choice of combinations
- Validation policy for Composite Crypto requires all signatures to be validated correctly (updates on this later)

Composite Crypto Summary

- Composite Crypto has several use-cases for both migration and backward compatibility for long-lived (device) environments
  - Leveraging security properties from different algorithms, especially useful to manage risk for deployed networks in transitioning periods (confidence in algorithms)
  - One migration tool that provides backward compatibility across newer and older devices, thus extending the lifetime of deployed devices up to the end of the “classic” crypto period
- Beyond Classic/PQC use-case
  - Allows for testing of new algorithms in the field, assuming at least one algorithm in the combined key is sound
  - Enables new use-cases (e.g., crypto evolution/planning, etc.)

New Case: Planning for Crypto Evolution

- Although X.509 provides all the needed data structures to provide Crypto-Agility, things are more complex when it comes to real deployments
- There are many characteristics of an algorithm, especially when it comes to Public Key operations, that might affect it’s deploy-ability in access networks, databases, etc.
- Cryptography is becoming a critical component of a device life cycle
  - Affects its longevity and it is becoming a non-negligible cost factor
- There are long-term dependencies to consider
  - Will the involved parties / devices have enough memory to run the new algorithms? Will the protocols support the new sizes? How about speed?

What if we could measure and plan?

- Let’s imagine we are planning the transition to the next generation of public key algorithms to be deployed in specific ecosystem (e.g., DOCSIS, OCF, Matter, etc.)
  - How can we understand if our networks can support the new algorithms?
  - How can we decide which type of algorithm will have better performances?
  - How can we plan for device replacements for the next 10 years?
- Let’s imagine that we define a new algorithm that we call “Algorithm X” that instead of providing security via signatures and encryption, it can simulate different algorithm’s characteristics and requirements and collect measurements
- Let’s also imagine we can define such algorithm without the need to also provide security to be able to measure fielded devices…

The “Algorithm X” paradigm

- Let’s imagine a Composite Key where we combine a “real” cryptographic algorithm (e.g., a classic algorithm or a PQ one) with the Algorithm X
  - Algorithm X can simulate different characteristics of public key algorithms such as ***sizes of keys and signatures***, memory requirements for different operations (e.g., key generation, signatures, validations, etc.)
  - Algorithm X can collect ***performance/execution data*** for the simulated algorithm and report it back directly into generated signatures
  - Algorithm X can ***safely fail*** without affecting the network operations
- Composite Crypto enables the “Algorithm X” paradigm that reduces the deployment risks without requiring complex/costly test beds

Documents Updates

Changes and New I-Ds

Major Changes in –keys-03

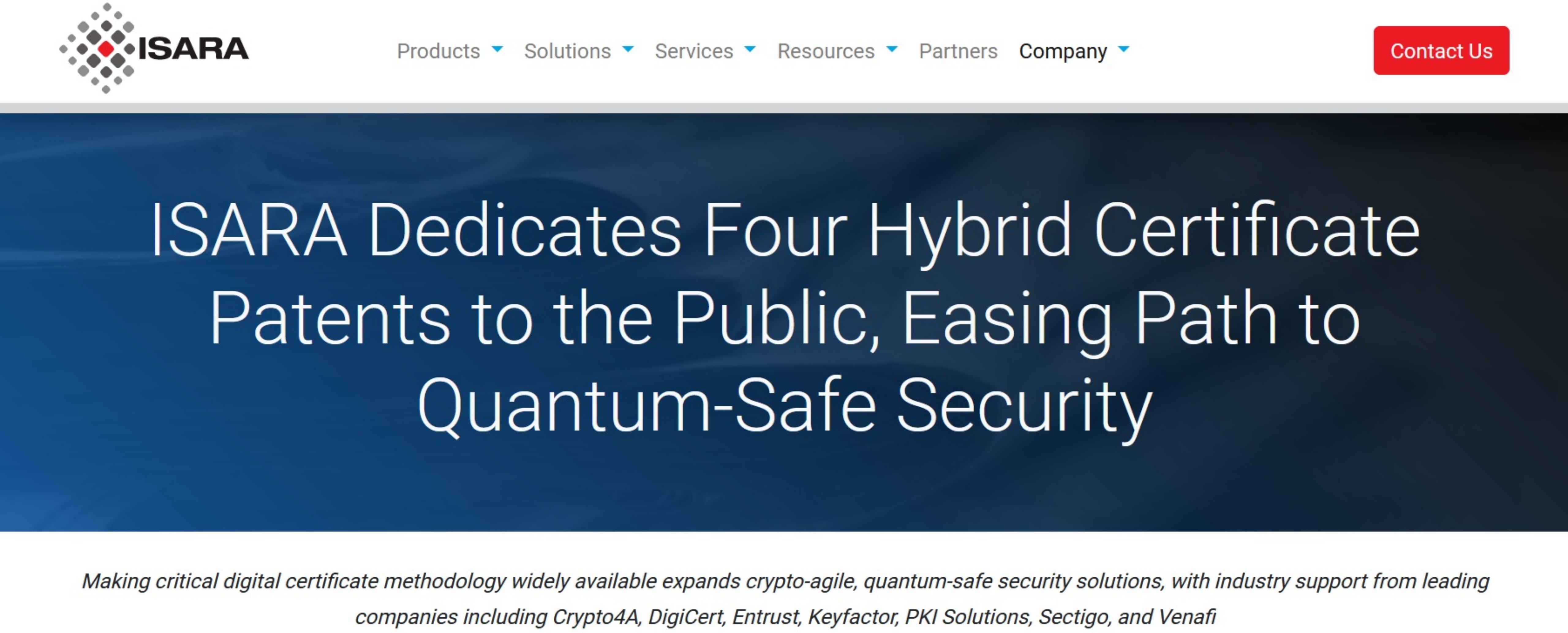
- Added the following explicit composite key types
  - Explicit Composite Signature Keys (7)
    - id-Dilithium3-ECDSA-P256
    - id-Dilithium3-RSA
    - id-Falcon512-ECDSA-P256
    - id-Falcon512-Ed25519
    - id-SPHINCSsha256256frobust-ECDSA-P256
    - id-Dilithium5-Falcon1024-ECDSA-P521
    - id-Dilithium5-Falcon1024-RSA
  - Explicit Composite KEM Keys (3)
    - id-Kyber512-RSA
    - id-Kyber512-ECDH-P256
    - id-Kyber512-x25519

- Added samples of (most of) the above explicit composites in appendices.
- Marked generic composite for prototyping; to be removed in final publication.
  - Authors group is not convinced this is right; more debate needed
  - Generic use-cases and non-standard algorithms use-cases needed
- Synchronized terminology with I-D.draft-driscoll-pqt-hybrid-terminology-01.
- Good on-list discussion already about whether those are the right combinations.

The New K of N draft

- The ‘K of N’ and ‘OR’ use-cases have been removed from the original proposal because of the indication from the WG that this mode could be more problematic
- We have worked on a new solution that is separated from the Composite Crypto draft that leverages the same structure but a public-key parameter
- The optional public-key parameter provides the indication of the required number of successful component signatures’ validations for the composite signature to be considered valid
  - K = 1 provides the pure “OR” validation policy (alternative algorithms)
  - K = n provides the default composite ”AND” validation policy (all required)
- Initial Version Available:
  - https://github.com/EntrustCorporation/draft-klaussner-pala-composite-kofn

ISARA Catalyst Hybrid Cert announcement



ISARA Dedicates Four Hybrid Certificate Patents to the Public, Easing Path to Quantum-Safe Security

Making critical digital certificate methodology widely available expands crypto-agile, quantum-safe security solutions, with industry support from leading companies including Crypto4A, DigiCert, Entrust, Keyfactor, PKI Solutions, Sectigo, and Venafi

ISARA Catalyst Hybrid Cert announcement

- This was announced on Oct 26, so have not had time for full implications analysis.
- Is the subject of now-expired *draft-truskovsky-lamps-pq-hybrid-x509*
  - Do we want to revive it?
  - We believe it complements, rather than replaces, composite and multi-cert, but a healthy debate is probably warranted!
- This mechanism has already been standardized by ITU-T
  - T-REC-X.509-201910 section 7.2.2
  - WG TODO: analyse; do we have technical feedback that we would want addressed in an IETF version?

Implementation Updates

Hackaton Results

Slides By: John Gray

Join Work (see next slide)

Thanks to All Participants!

- We would like to thank all the contributors for their work and sustained support
- Agreed on Monthly Meetings to continue progress
  - Monday Dec 5 @ 12:00 UTC
  - Expand artifacts and add X.509 based protocols
- The Amazing Hackaton Team:

  Mike Ounsworth, John Gray, Felipe Ventura, Jake Massimo, Cory Bonnell, Michael Baentsch, Kris Kwiatkowski, Alexander Railean, Pat Kelsey, Britta Hale, Tomofumi Okubo, Carl Wallace, Max Pala, Markku-Juhani O.Saarinen, David Hook, @bblfish

PQ Keys and Signatures Hackathon

Goals:

- Production and validation of X.509 keys, certificates, PKCS10, CRLs and other X.509 structures with the new NIST algorithms (Dilithium, Falcon, SPHINCS+, Kyber) alone and in composite combinations with traditional crypto
- Solving ASN.1 encoding issues to help clarify specifications in the new drafts
- Obtain experience with practical use of the new NIST algorithms in X.509
- Provide an artifact repository for interoperability testing

RFC Drafts:

- https://datatracker.ietf.org/doc/html/draft-uni-qsckeys-00.html
- https://datatracker.ietf.org/doc/draft-ietf-lamps-dilithium-certificates/
- https://datatracker.ietf.org/doc/draft-ounsworth-pq-composite-keys/
- https://datatracker.ietf.org/doc/draft-ounsworth-pq-composite-sigs/
- RFC 5280, 5208, 5958, 2986 (Public and Private key formats, Certificate Request, others)
- https://www.secg.org/sec1-v2.pdf - section section 2.3.1/2.3.2

Hackaton Git Repo

- The **GitHub Hackaton repository** provides a set of generated examples for X.509 data structures that make use of post-quantum and composite algorithms
- The artifacts archive format is defined as follows (artifacts.zip)

```
artifacts/
 |
 +-> oid_number/            // Ex. 1.3.6. ..
      |
      +-> ta/               // Trust Anchor (RootCA)
      +-> ca/               // Intermediate CA
      +-> ee/               // End Entity
      +-> crl/              // Issued CRL
      +-> ocsp/             // Issued OCSP response
```

- 7 different providers already available in the archive

What got done

- Formed new Hackathon team with about 16 participants!
  - Created Github artifact repository https://github.com/IETF-Hackathon/pqc-certificates
  - Defined a .zip file structure for X.509 artifacts to make interoperability testing easier. These include pure PQ artifacts as well as composites.
  - Agreed on public and private key ASN.1 encodings (See what we learned).
  - 7 different implementations (Java, C, Python, Rust).
    - Open Source (OpenSSL, Bouncy Castle, Python, LibPKI)
    - 4 Vendor implementations

What we learned

- Public Keys – OCTET STRING can be mapped to BIT STRING from RFC 5208
  - https://www.secg.org/sec1-v2.pdf section section 2.3.1/2.3.2
  - “**treat BIT STRING and OCTET STRING identically** (doing sensible 0-bit-padding if BIT STRING is no multiple of 8)“
  - Thus, no need for wrapping/adding another TLV layer for implementations that internally operate on octet strings (and tag BIT STRINGS only where the standard mandates it)”

What we learned

- Private Keys - No need to have an OCTET_STRING wrapping another OCTET_STRING.   We will use a Single OCTET_STRING representation as per 5958.
- OIDS – Object Ids need to be flexible at this point, and we are suggesting they be version controlled as there are still tweaks being made to the NIST competition winners (Dilithium, SPHINCS+, Falcon, Kyber)
- Suggest <Arc>.Version.SecurityLevel
- Most issues found are not related to PQ algorithms
    - Setting the right extensions when testing certs (e.g., CA:TRUE)

What’s Next ?

- We think that the work on Composite Crypto for Keys and Signatures is in a very good status, and we would like for the KEYS and SIGS documents ***to be adopted by the WG***
- We are continuing the development of the `K of N` initial draft and we will post the -00 version as soon as it is possible to do so, but we are not ready for adoption at this time
- We are hosting regular meeting to discuss implementations details related to composite and PQC certificates
  - (question for the chairs) Shall the discussion be on the LAMPS mailing list or shall we ask for a separate mailing list ?