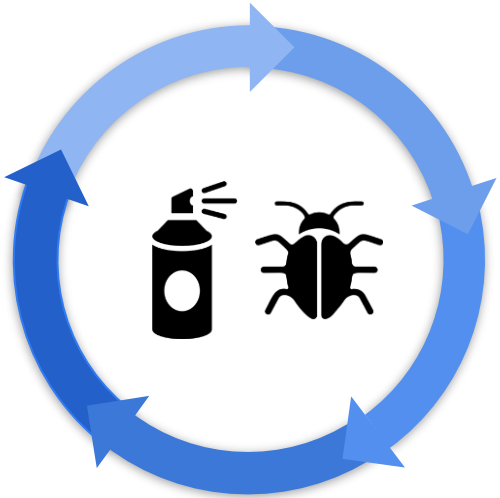
Implicit Attestation at Scale

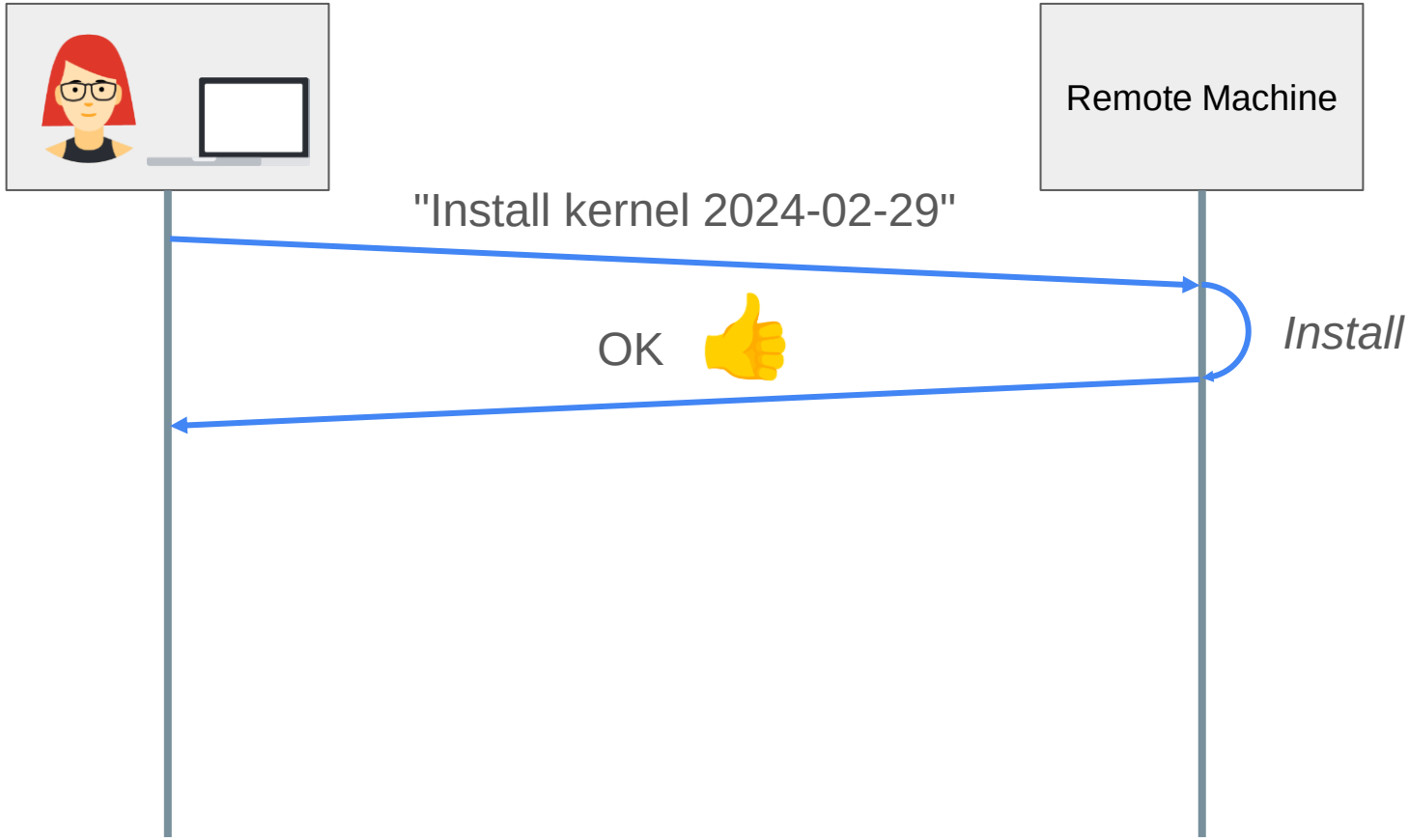
Chris Fenner, Google

Remote Attestation Gives Us Recoverability

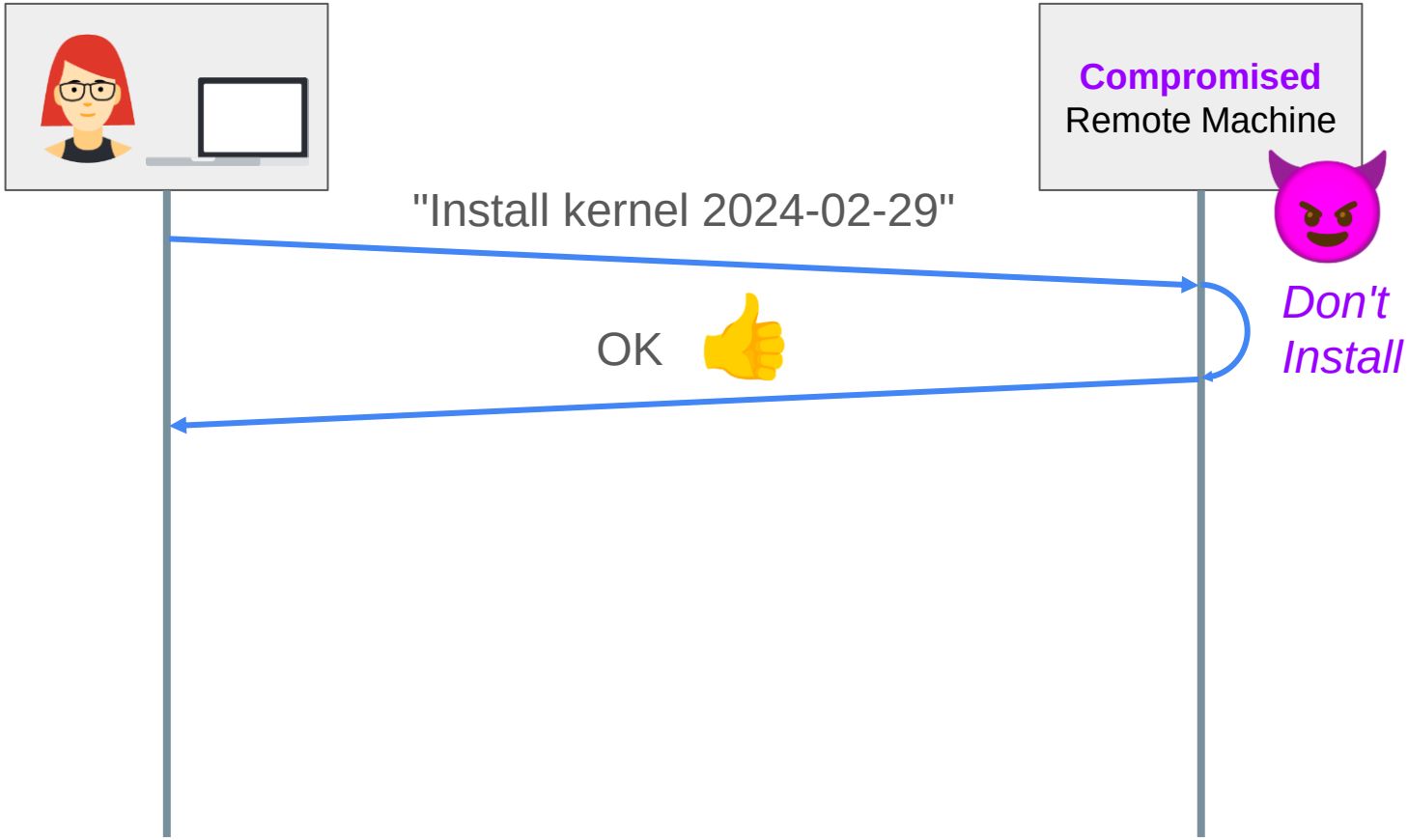
- Step 1: Old code is running in the fleet
- Step 2: Find bugs in the old code
- Step 3: Deploy new code with fixes
- Step 4: **Verify the fixed code got deployed**
- Step 5: GOTO step 1



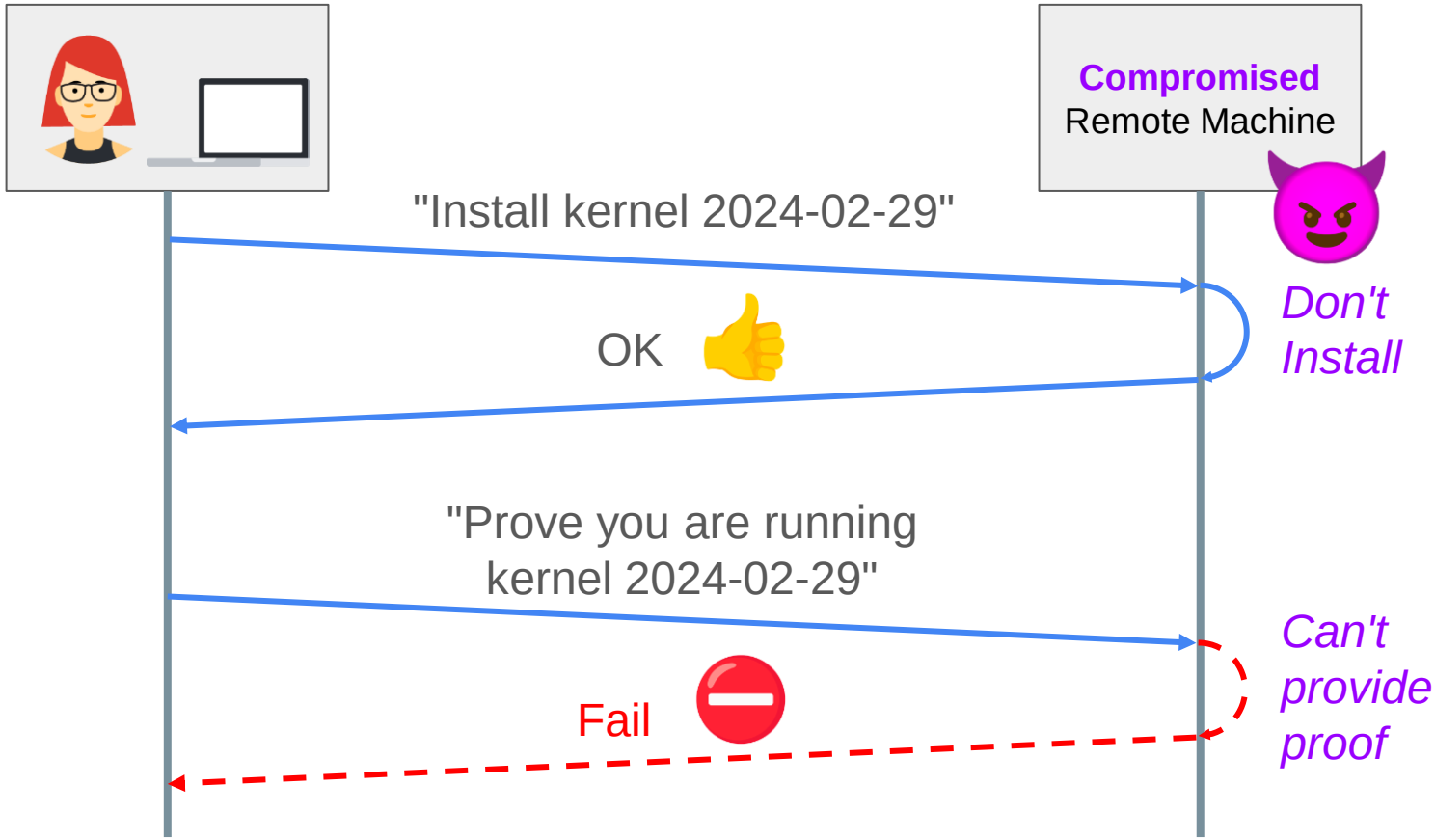
Remote Attestation Needs Cryptographic Evidence



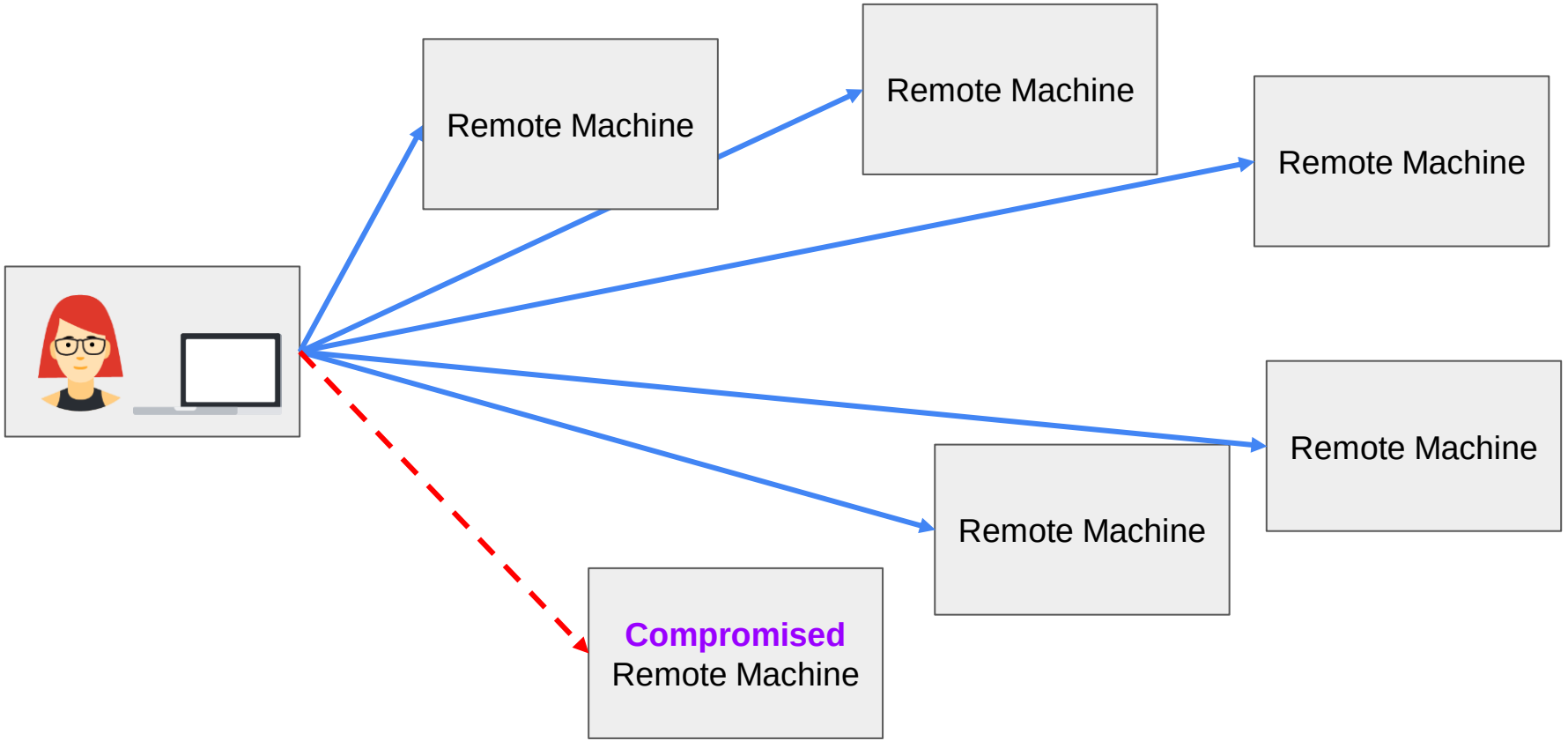
Remote Attestation Needs Cryptographic Evidence



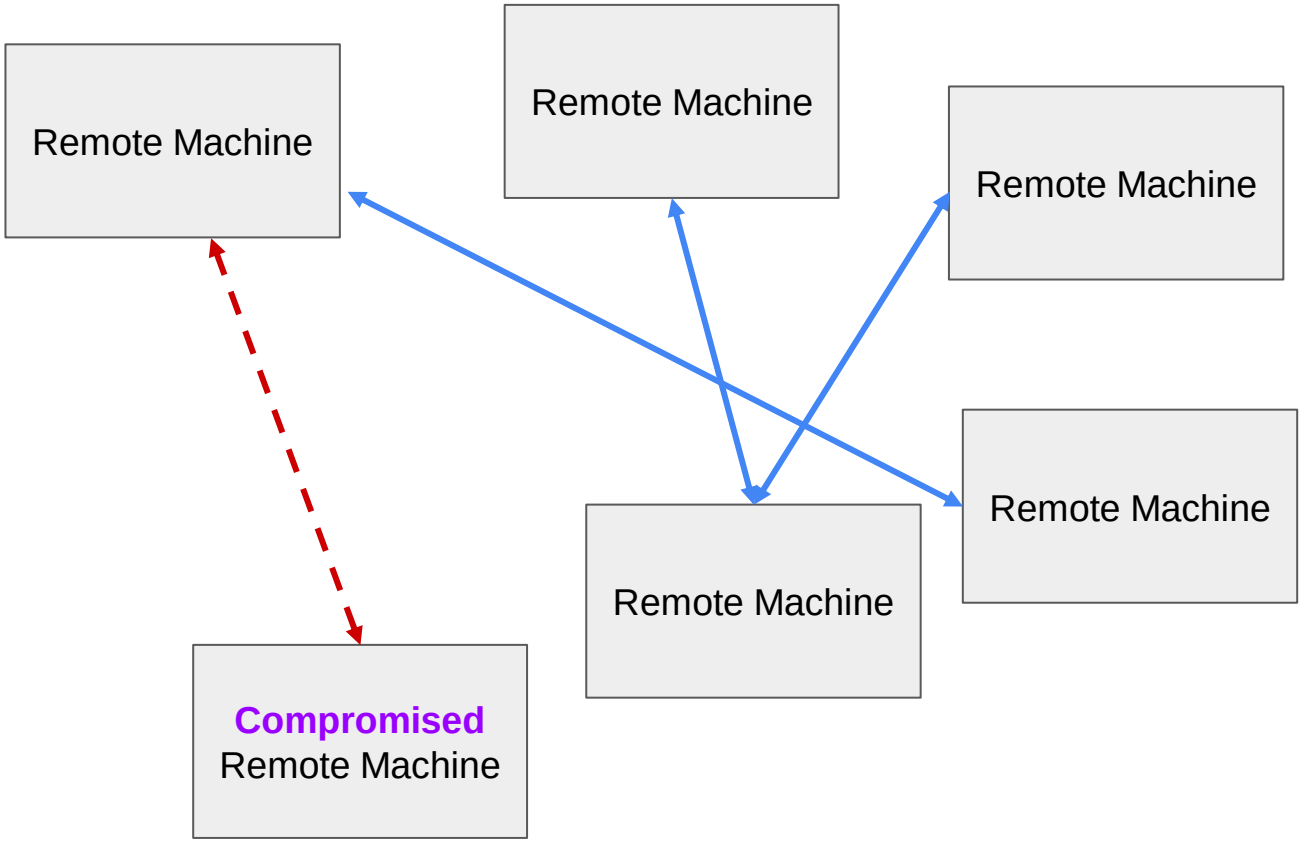
Remote Attestation Needs Cryptographic Evidence



Challenge: Scale

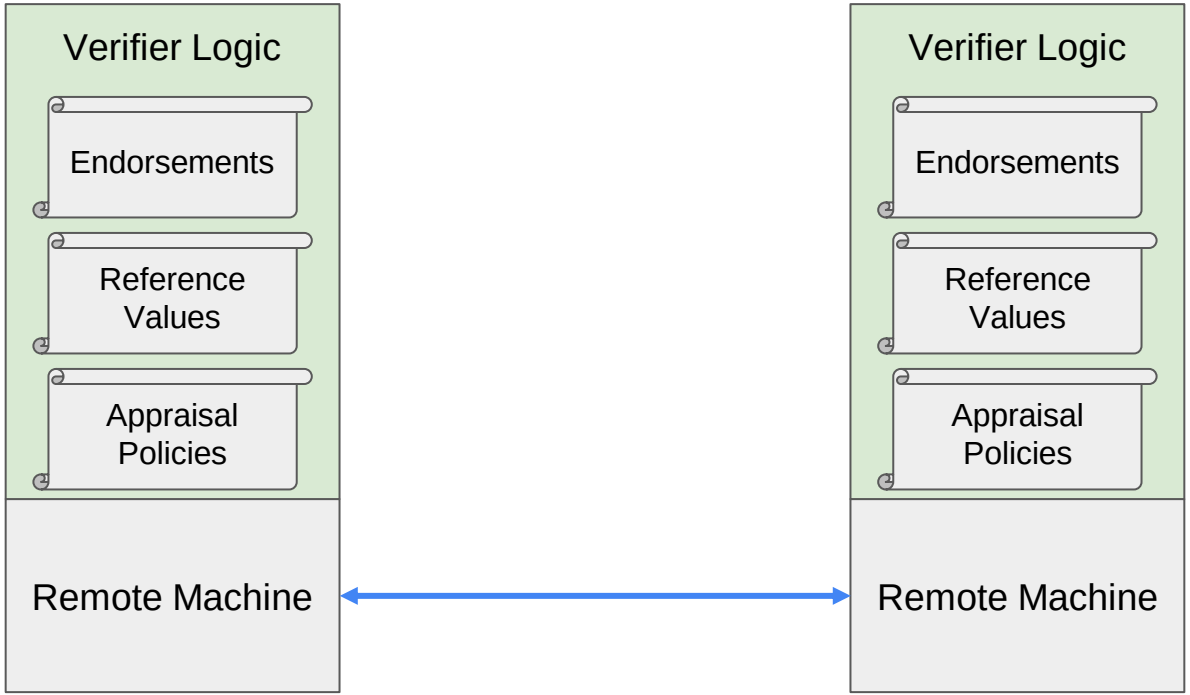


Solution: Make Machines Verify Each Other

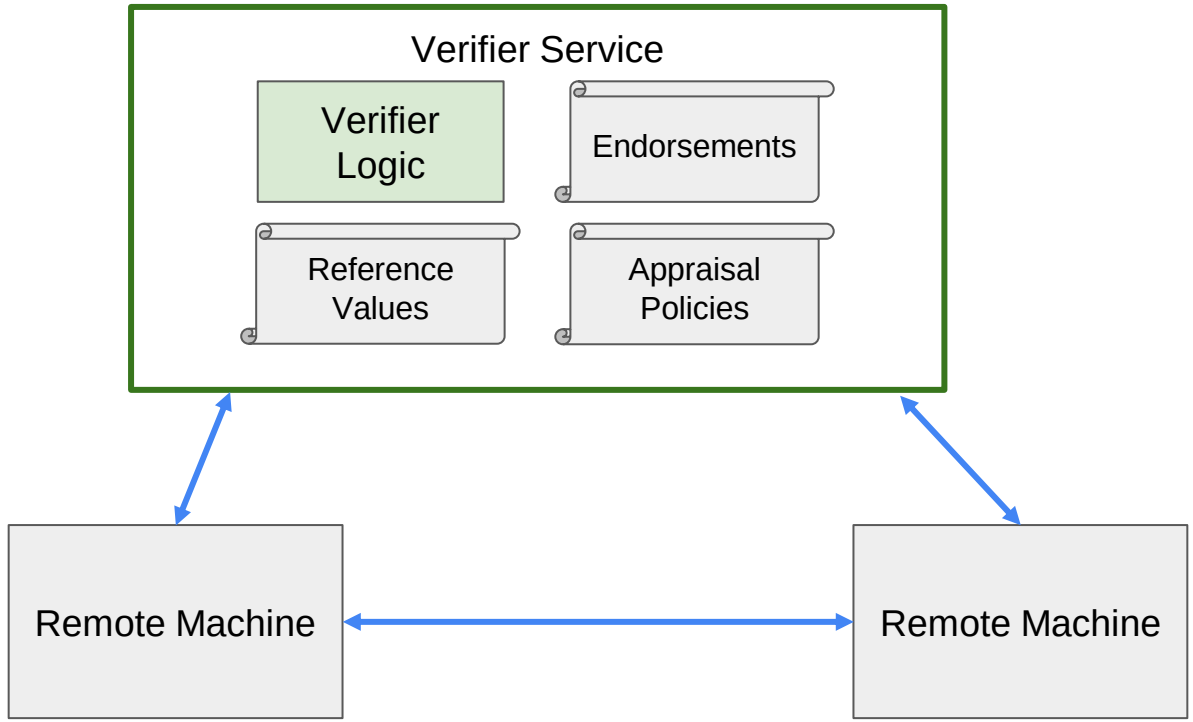


Where to Put Attestation Verifier Logic?

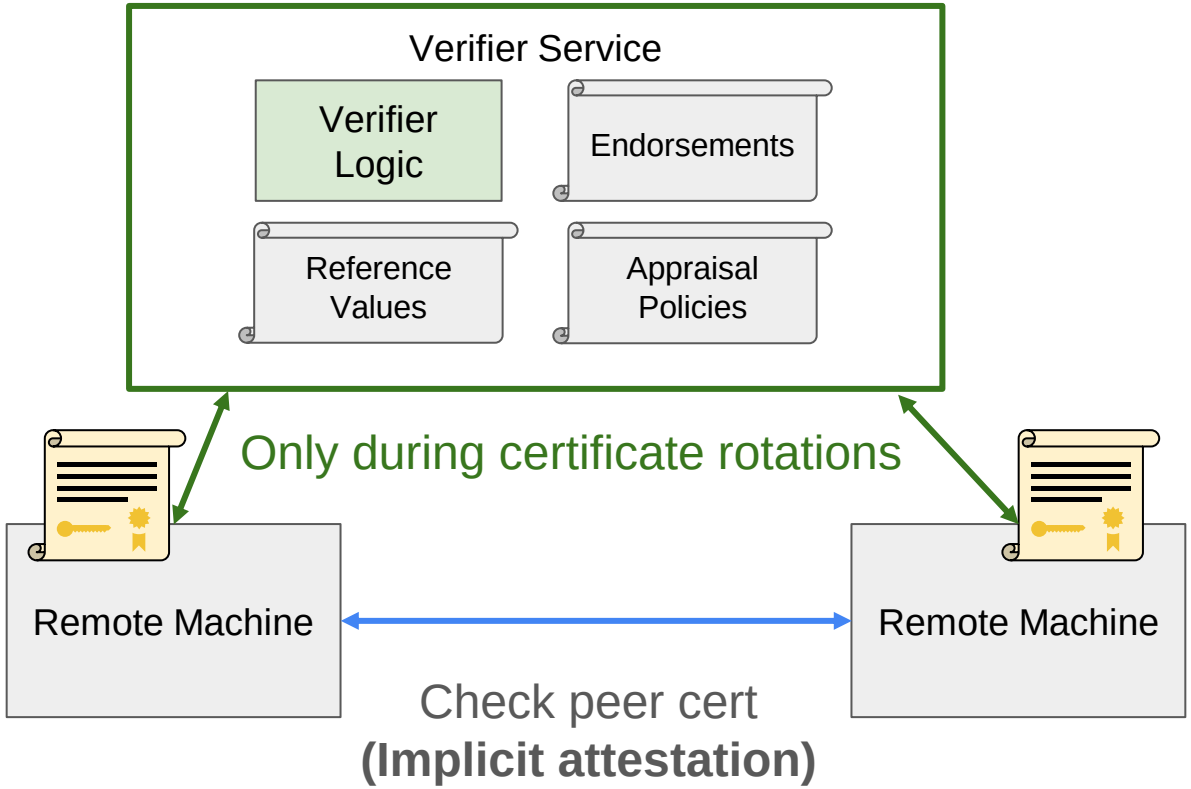
Where to Put Attestation Verifier Logic?



Where to Put Attestation Verifier Logic?



Where to Put Attestation Verifier Logic?



Q&A

What to Expect about Post-Quantum Computing

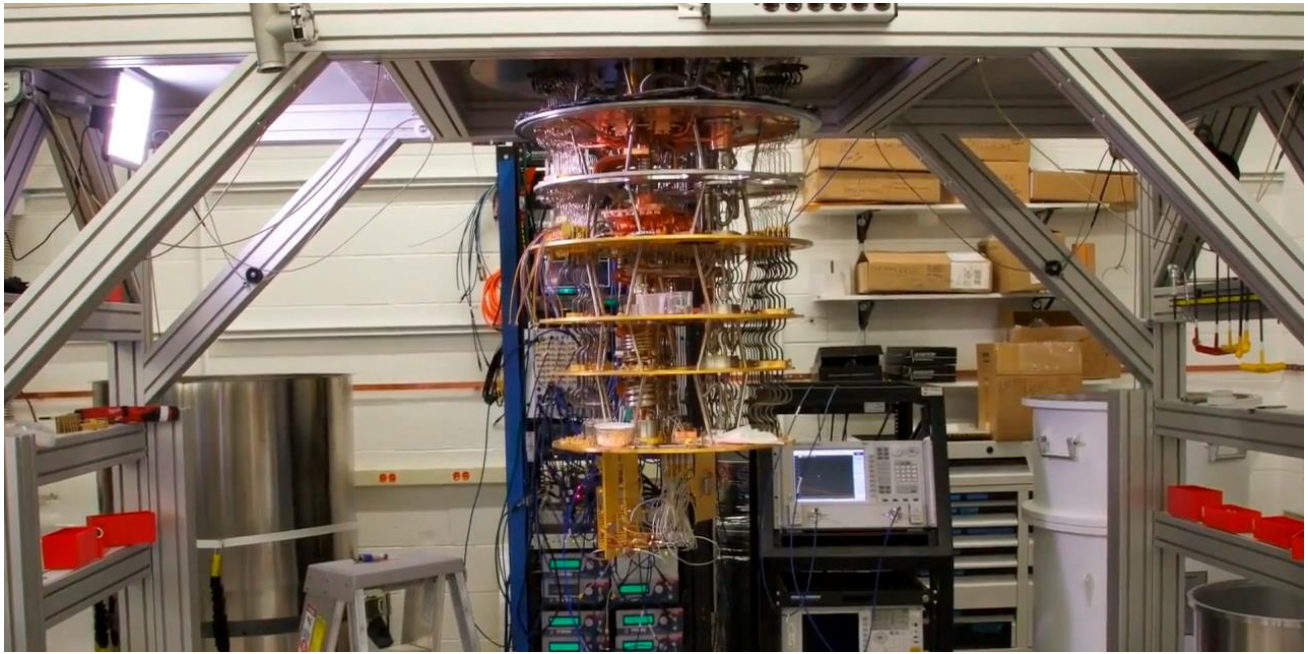
Chris Fenner, Google

Agenda

- Why care about post-quantum cryptography
- Synopsis of new algorithms
- How to prepare

Why care about post-quantum cryptography?

Quantum computers will solve new kinds of problems…



Source: https://quantumai.google/hardware

…including the ones assumed Hard by cryptographers

Shor's algorithm

- solves discrete log/factor
- breaks RSA and ECC
- can be parallelized
- can't fix by changing key sizes
- fix by changing algorithms
  - e.g., lattice crypto algorithms (encapsulation and signing), hash-based algorithms (signing)

Grover's algorithm

- unstructured search in √N time
- birthday attacks in $\sqrt[3]{N}$ time
- slightly weakens symmetric crypto
- can't be parallelized
- if paranoid, fix by increasing key/hash sizes
  - e.g., AES-256, SHA-384

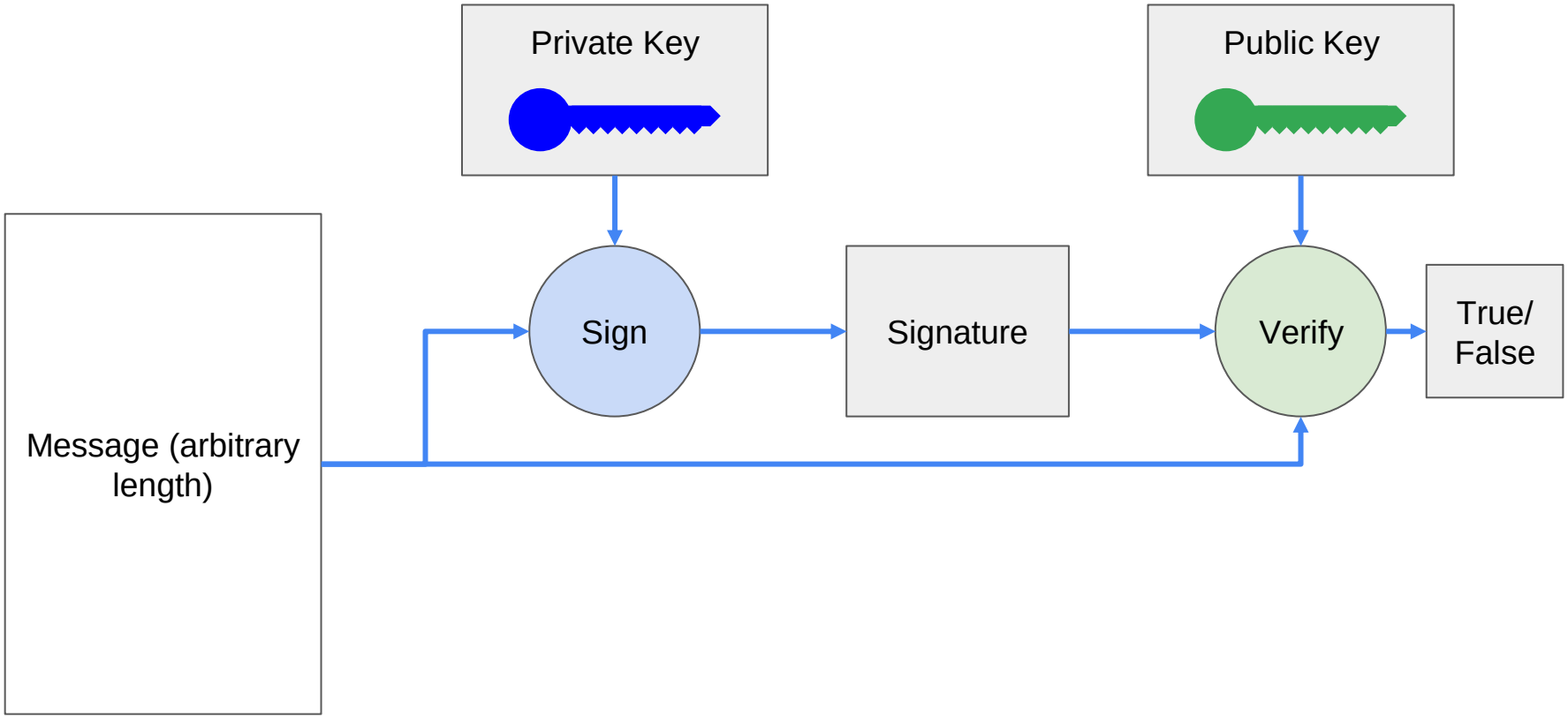
When will we have RSA-2048-scale quantum computers?

Timeframe	5 years	10 years	15 years	20 years	30 years
Experts' Estimated Likelihood	4-11%	17-31%	33-54%	56-78%	75-92%

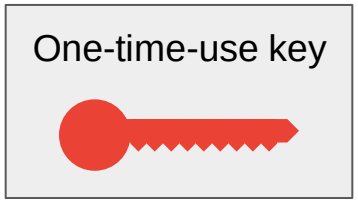
Data source: https://globalriskinstitute.org/publication/2023-quantum-threat-timeline-report/

Synopsis of new algorithms

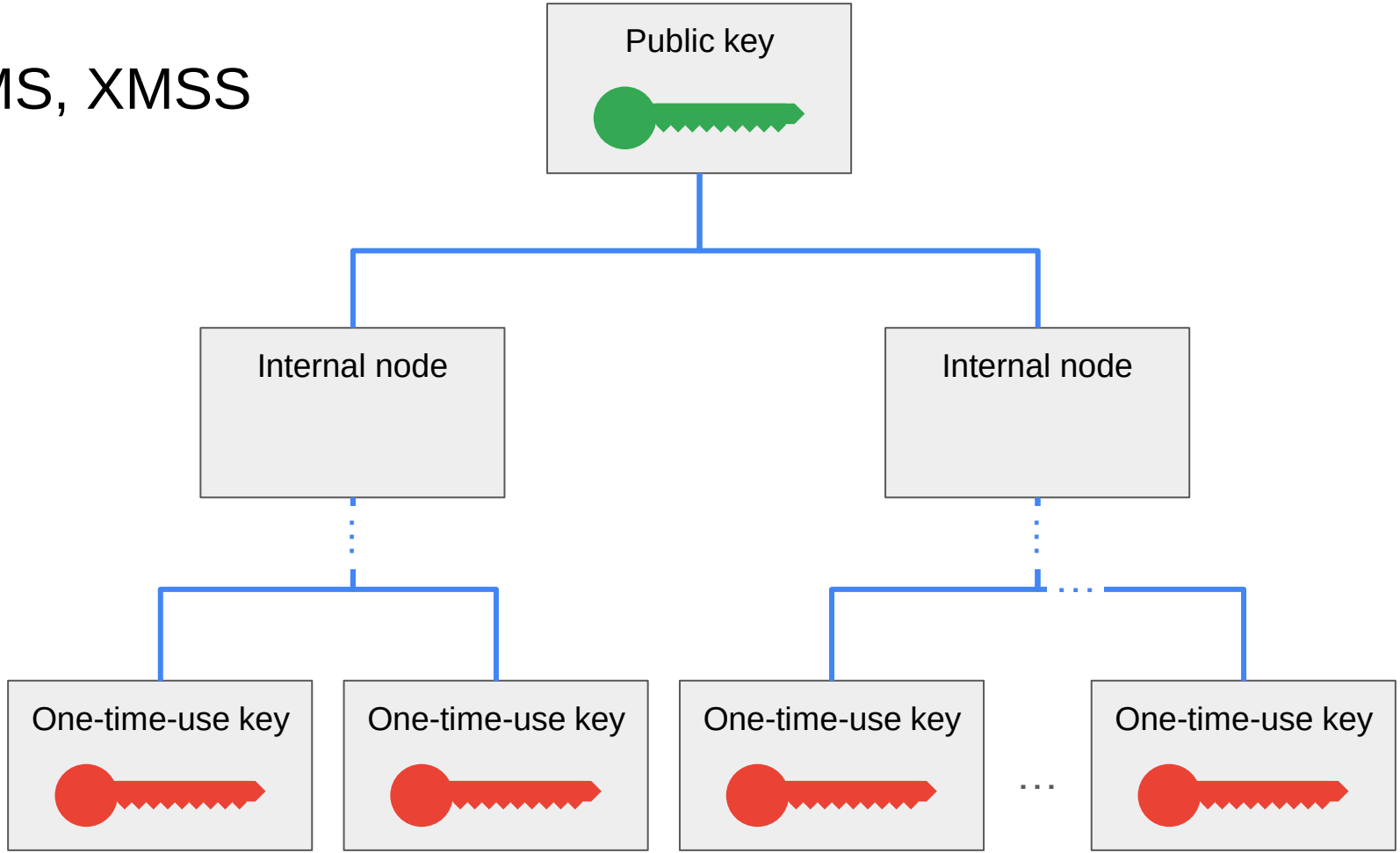
All PQC signature algorithms



LMS, XMSS



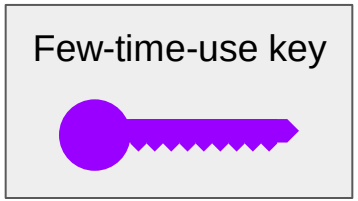
LMS, XMSS


Public key
Internal node
Internal node
One-time-use key
One-time-use key
One-time-use key
One-time-use key

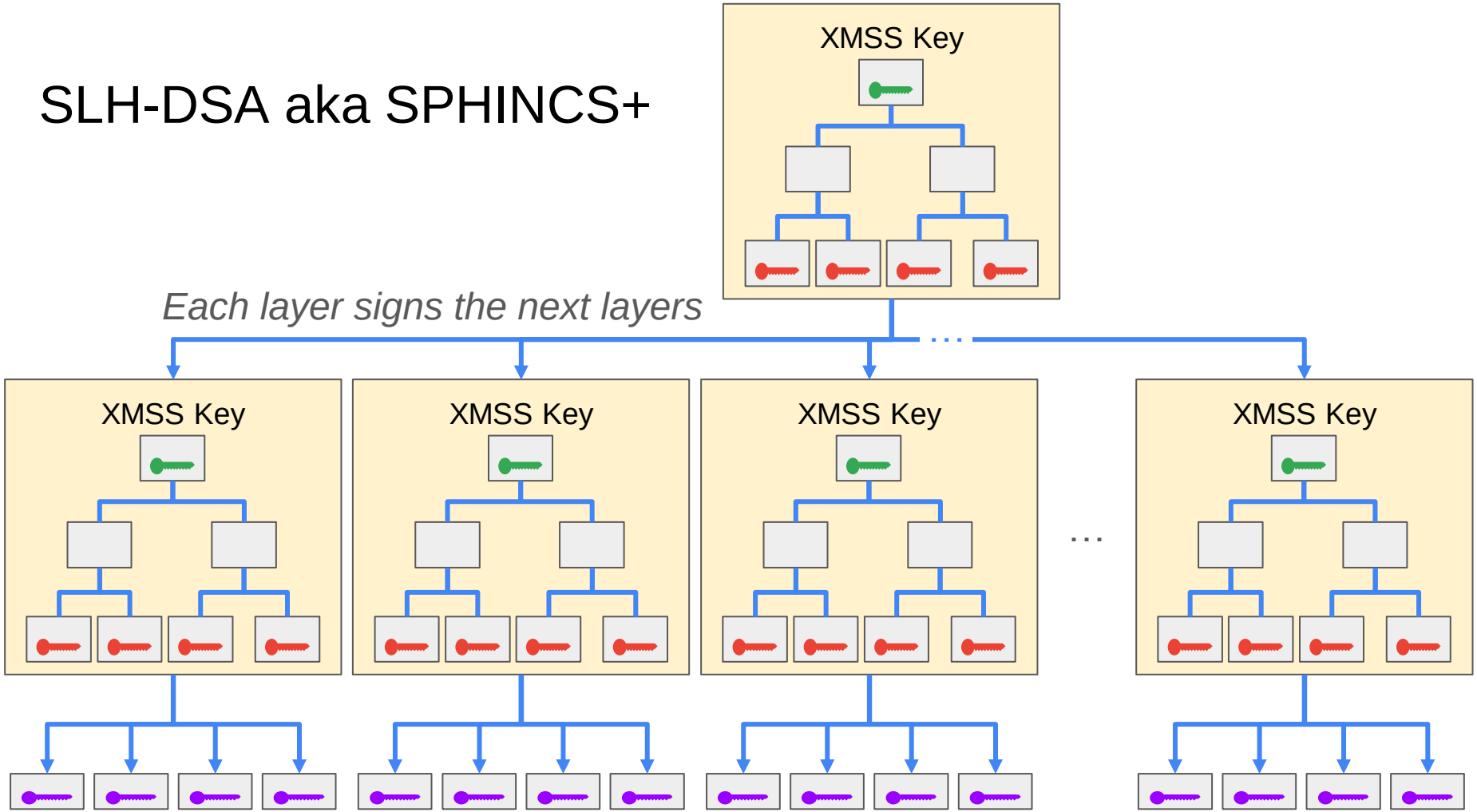
LMS, XMSS

NIST parameter set	# of sigs	hash size	w	Public size	Signature size
LMS_M24_H5_W8	2^5 (32)	24 bytes	2^8 (256)	40 bytes	780 bytes
LMS_M24_H5_W1	2^5 (32)	24 bytes	2^1 (2)	40 bytes	8684 bytes
LMS_M32_H25_W1	2^{25} (33M)	32 bytes	2^1 (2)	48 bytes	9324 bytes
XMSS_10_192	2^{10} (1024)	24 bytes	2^4 (16)	48 bytes	1492 bytes
XMSS_20_256	2^{20} (1M)	32 bytes	2^4 (16)	64 bytes	2820 bytes
RSA-2048	∞	--	--	~256 bytes	256 bytes
ECDSA-P256	2^{128} (~∞)	--	--	~64 bytes	64 bytes

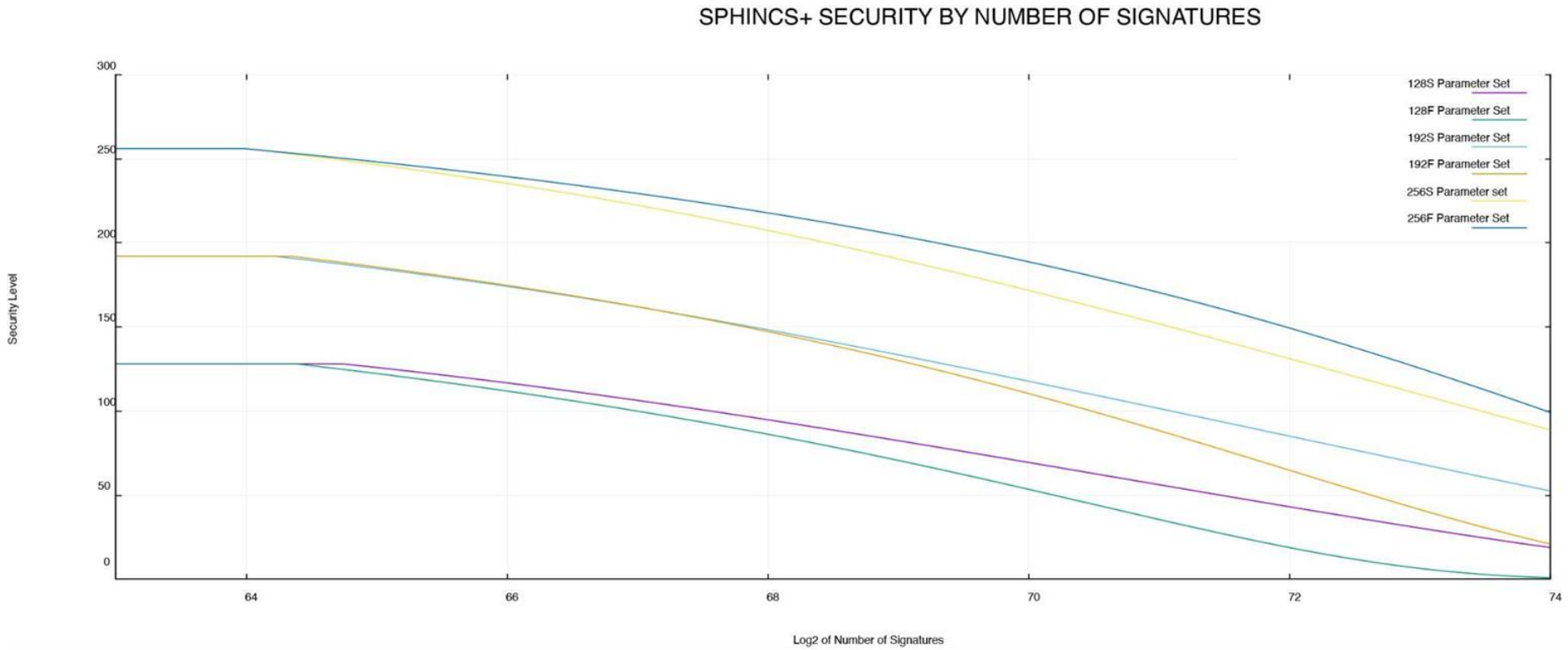
SLH-DSA aka SPHINCS+



SLH-DSA aka SPHINCS+



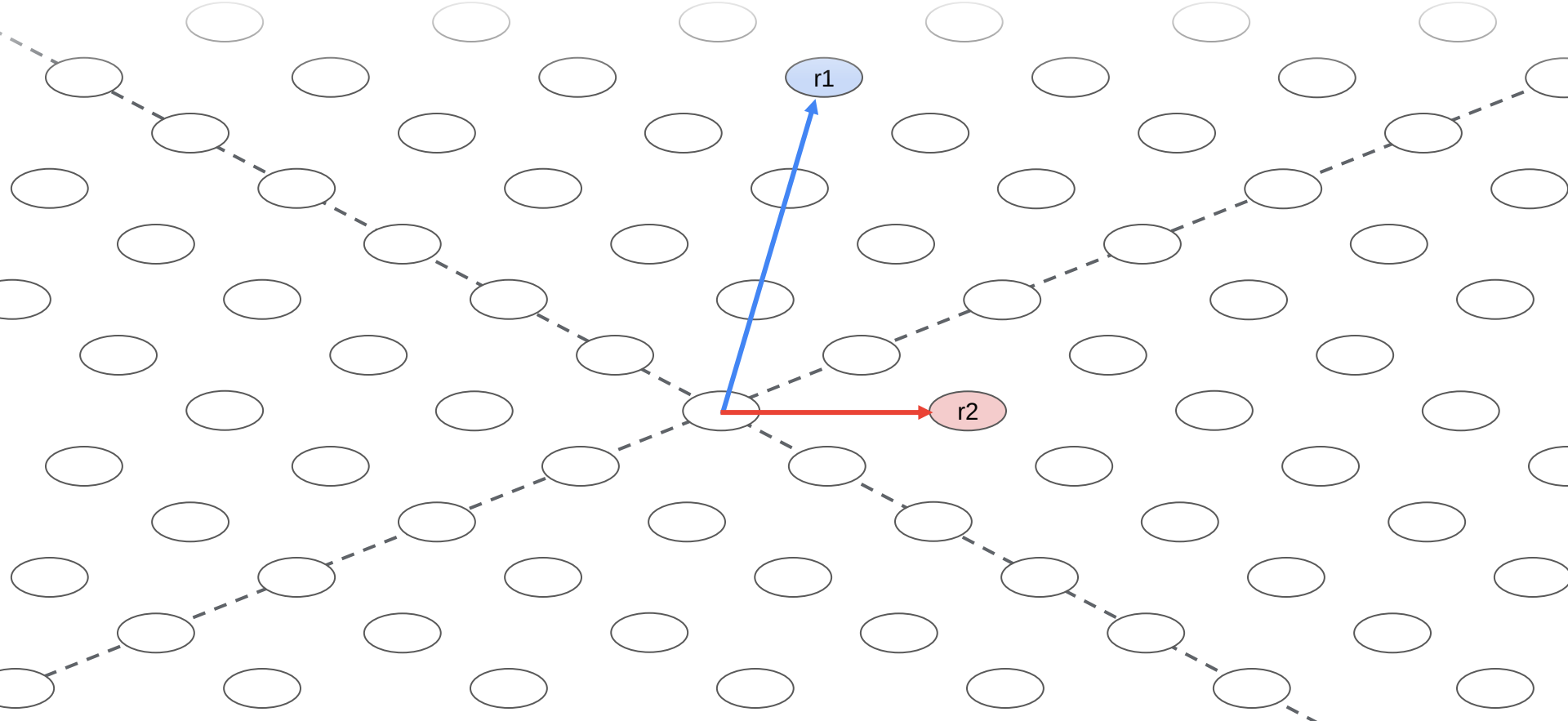
SLH-DSA aka SPHINCS+



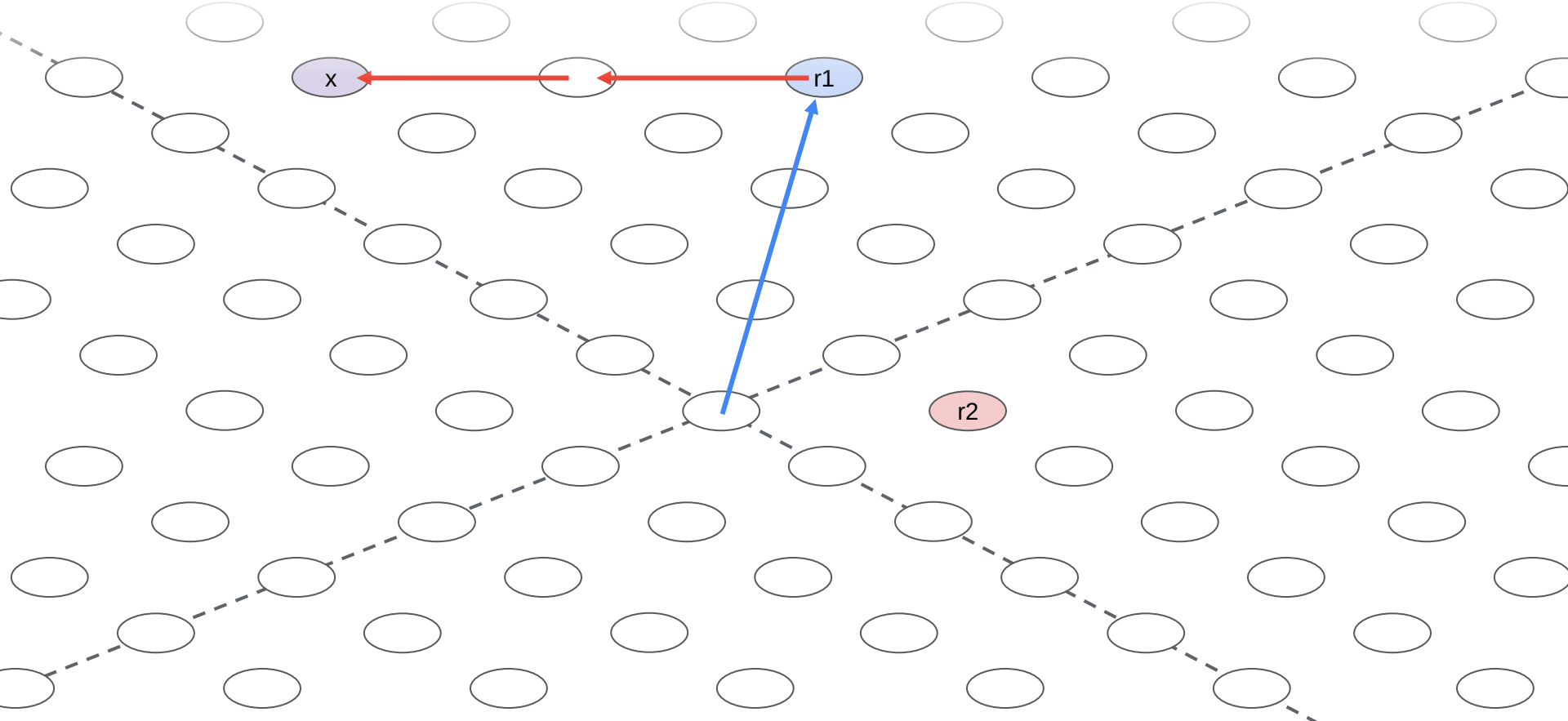
SLH-DSA aka SPHINCS+

Parameter set	# of sigs	Private size	Public size	Signature size
SPHINCS+-128s	2^{64}	64 bytes	32 bytes	7856 bytes
SPHINCS+-128s-q20	2^{20}	64 bytes	32 bytes	3264 bytes
SPHINCS+-192s	2^{64}	96 bytes	48 bytes	16224 bytes
SPHINCS+-192s-q20	2^{20}	96 bytes	48 bytes	7008 bytes
SPHINCS+-256s	2^{64}	128 bytes	64 bytes	29792 bytes
SPHINCS+-256s-q20	2^{20}	128 bytes	64 bytes	12640 bytes
RSA-2048	∞	~256 bytes	~256 bytes	256 bytes
ECDSA-P256	2^{128} (~∞)	~32 bytes	~64 bytes	64 bytes

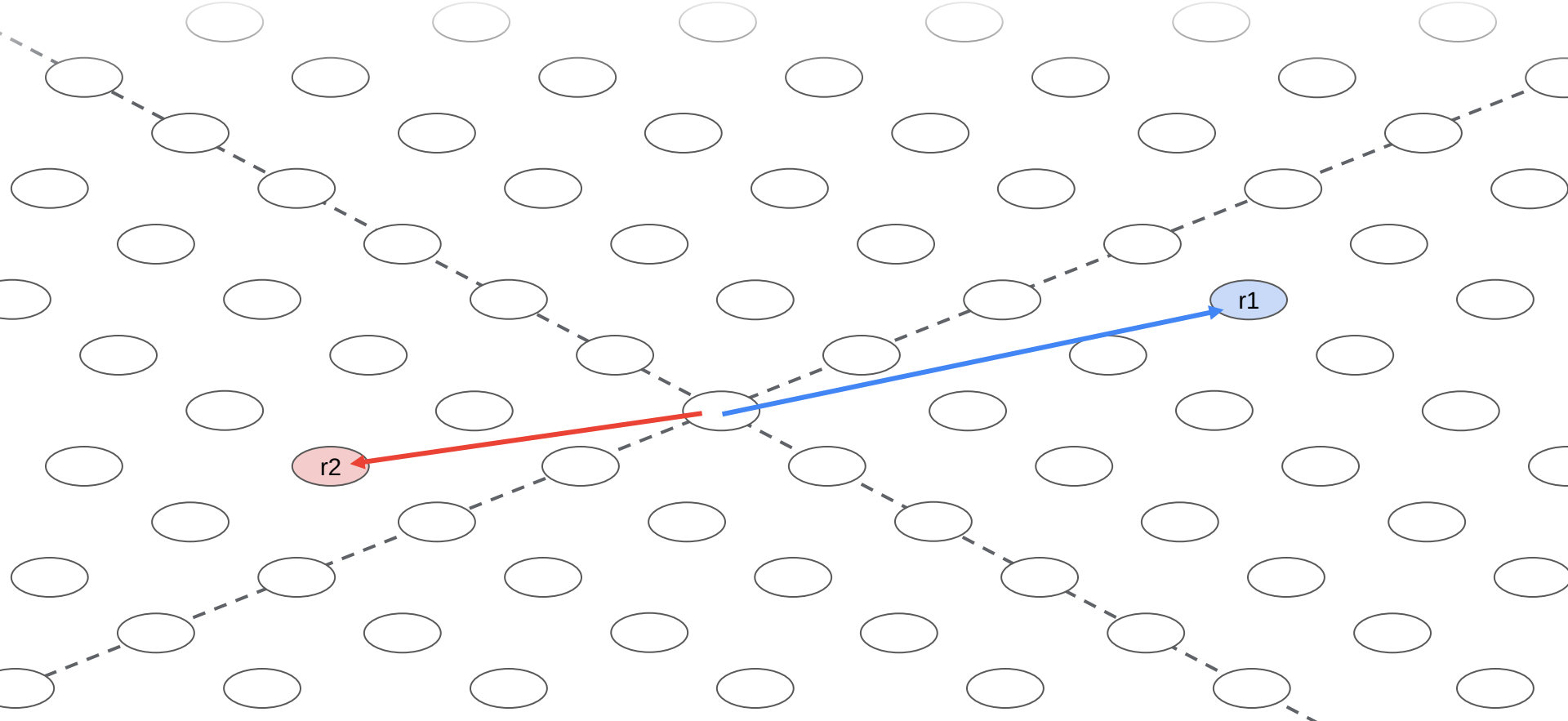
Lattice Cryptography In 2 Minutes



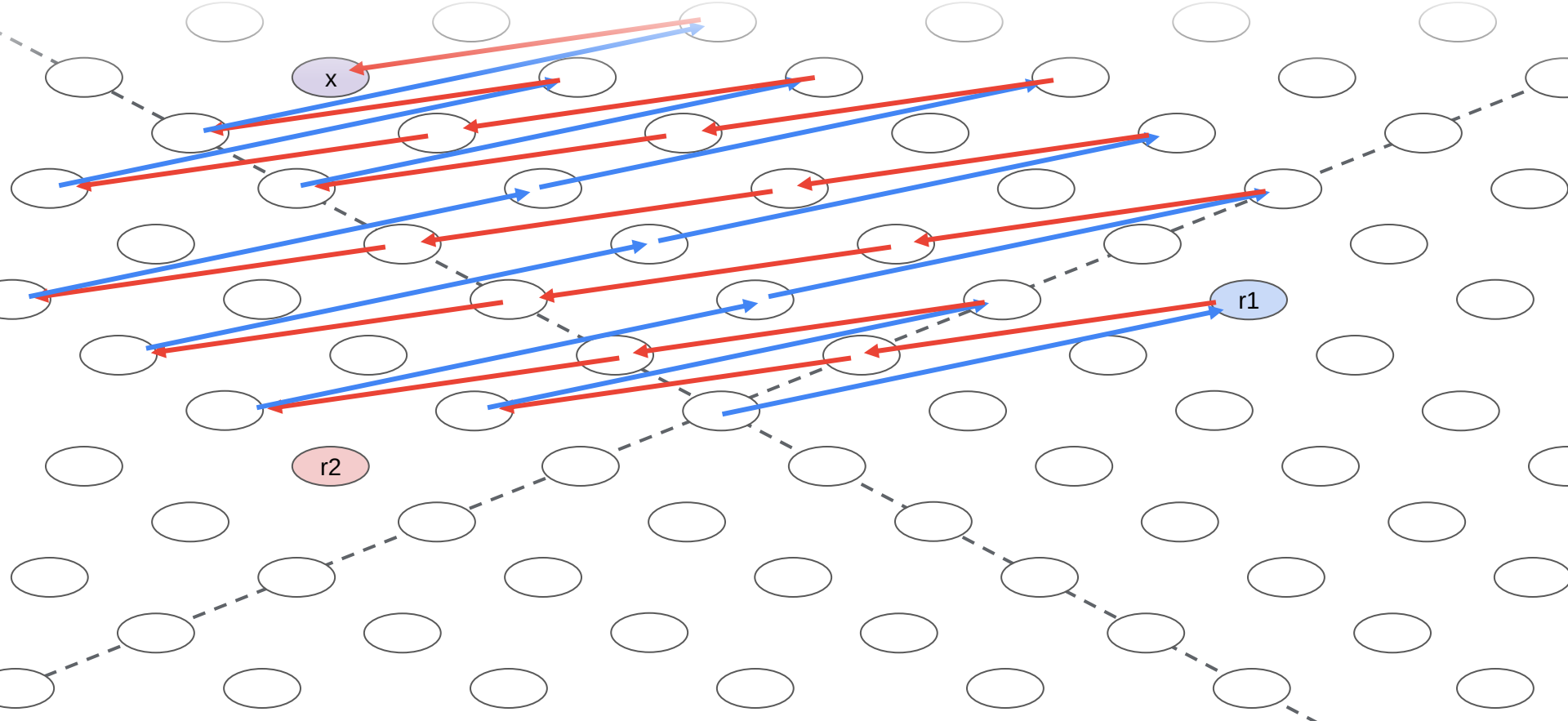
Lattice Cryptography In 2 Minutes



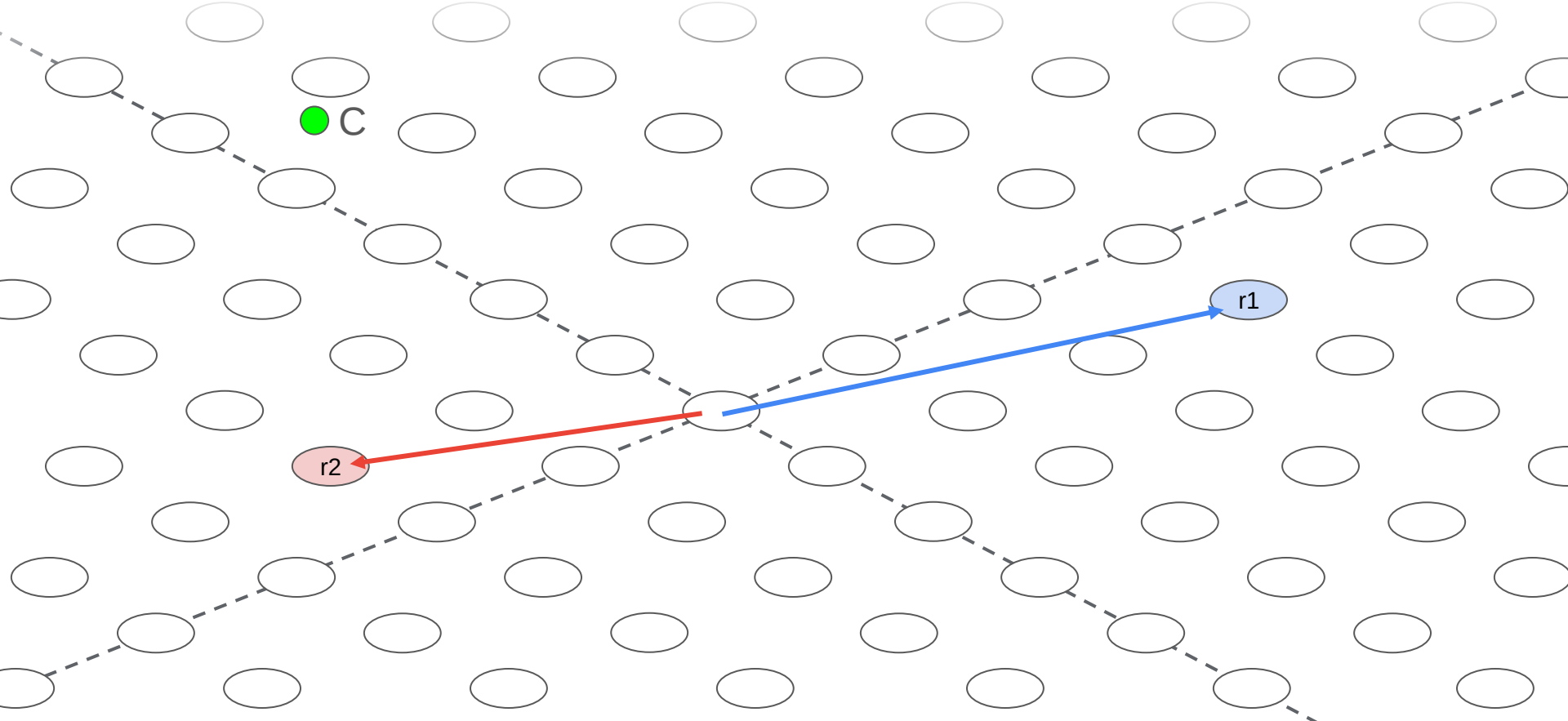
Lattice Cryptography In 2 Minutes



Lattice Cryptography In 2 Minutes



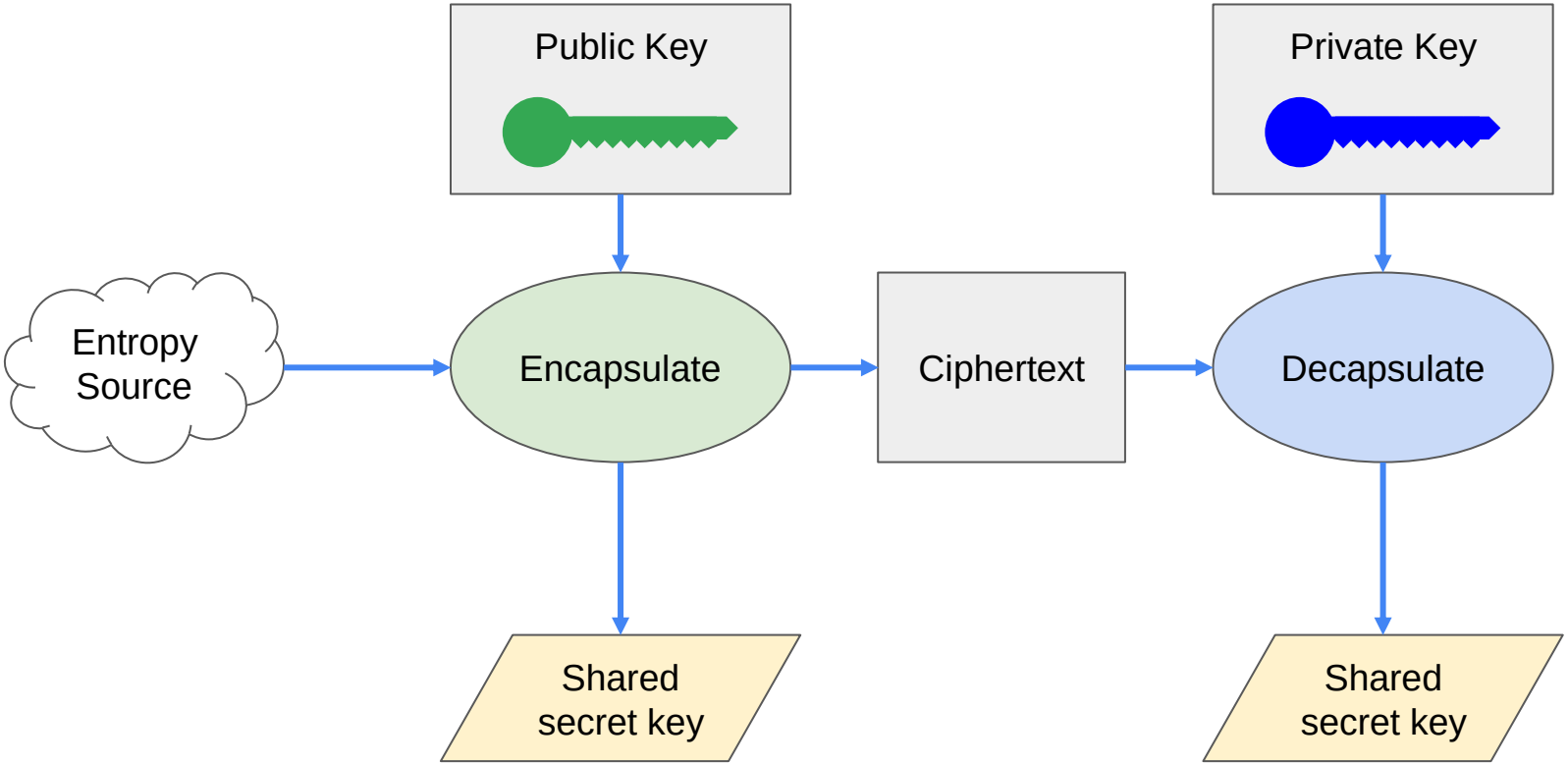
Lattice Cryptography In 2 Minutes



ML-DSA aka Dilithium

Parameter set	Private size	Public size	Signature size
ML-DSA-44	2528 bytes	1312 bytes	2420 bytes
ML-DSA-65	4000 bytes	1952 bytes	3293 bytes
ML-DSA-87	4864 bytes	2592 bytes	4595 bytes
RSA-2048	~256 bytes	~256 bytes	256 bytes
ECDSA-P256	~32 bytes	~64 bytes	64 bytes

All PQC key-encapsulation algorithms



ML-KEM aka Kyber

Parameter set	Private size	Public size	Ciphertext size
ML-KEM-512	1632 bytes	800 bytes	768 bytes
ML-KEM-768	2400 bytes	1184 bytes	1088 bytes
ML-KEM-1024	3168 bytes	1568 bytes	1568 bytes
RSA-2048	~256 bytes	~256 bytes	256 bytes
ECIES-P256	~32 bytes	~64 bytes	~32 bytes

How to prepare

In Protocol Design: Prefer Signing over Encryption

Store-Now-Decrypt-Later Attacks:

> Assume that your adversary is recording your encrypted network traffic today and plans to decrypt it in 10 or 20 years. Will they get anything of value?
>
> An adversary who steals your signing key can sign new things, but you can fight that by revoking your signing key after PQC happens

There are more options to choose from when it comes to PQC signing algorithms (LMS, XMSS, SLH-DSA, ML-DSA, etc)

In Interface Design: Plan for Big Keys and Ciphertexts

- The smallest NIST-approved PQC signature is 780 bytes (LMS_M24_H5_W8) and it's for a key that can sign only 32 times
- Prepare for public and private keys of size 1-4KB, and signatures and ciphertexts of size 1-4KB or more (especially if considering SLH-DSA aka SPHINCS+)

In Hardware Design: Implement a SHA3 Block

- Most of the effort of ML-DSA and ML-KEM is in SHA3, which is very efficient in terms of performance per die area
- SHA3 is also needed in order to implement the SHA3-variant parameter sets of the hash-based schemes
- SHA3 is also useful for more than just hashing (see SHAKE, KMAC)
    - So make sure your hardware block supports the whole SHA3 family. ML-KEM alone will use all of the variants!

In Application Design: Plan for Algorithm Changes

- Incorporate reasonable algorithm agility into your application, so that you can switch from ECDSA to ML-DSA to $FUTURE_ALGORITHM as we learn more

In Application Design: Plan for Hybrid Constructions

- Combine classical and post-quantum algorithms to ensure security is not reduced in the case of a future discovery that compromises the new quantum algorithms
- You can combine ECDSA with ML-DSA (example)
  - This example signs the the ECDSA signature again with the ML-DSA key to improve the total scheme's strong-unforgeability properties
- X-Wing: X25519 + ML-KEM-768
- A future NIST publication (SP800-227) will give more general guidance about using and combining KEMs

Q&A