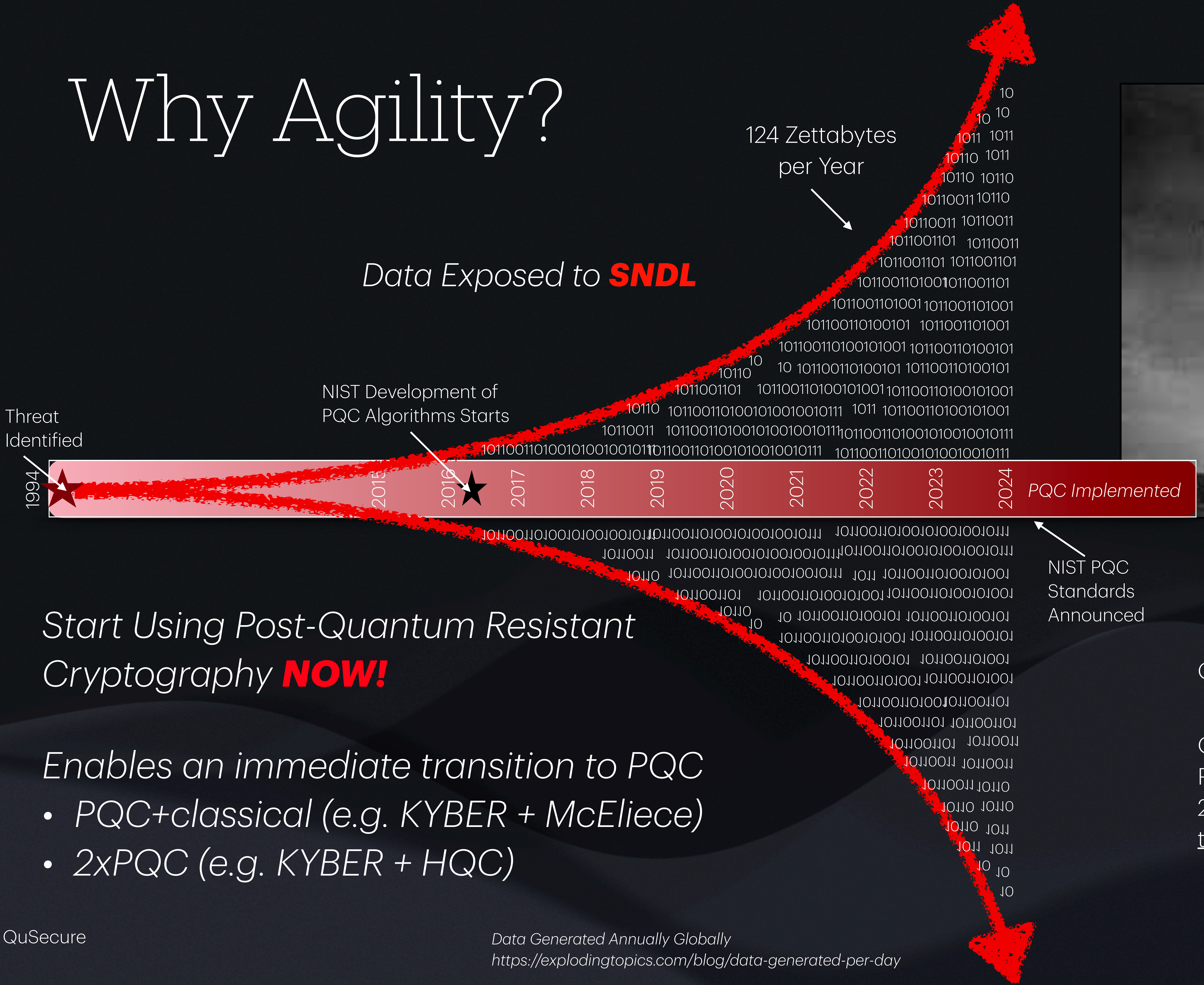
Crypto-Agility

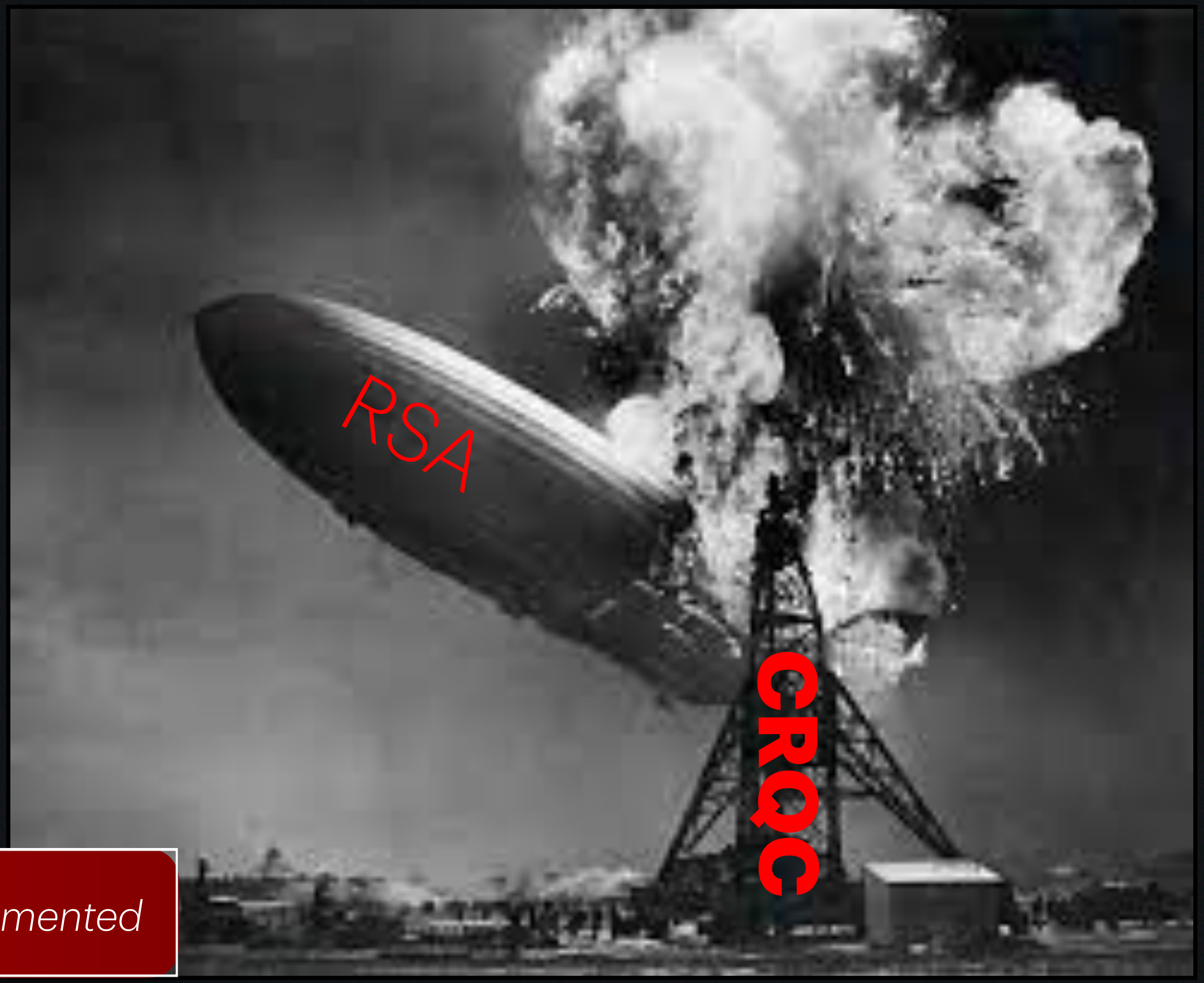
Aaron Moore, CTO, QuSecure, aaron@qusecure.com: 3/12/2024

Crypto-agility is the ability to rapidly modify the cryptography used by a protocol, application, or service in order to establish and preserve a desired security posture with minimal disruption to the system.

Why Agility?



(>700 ZB at Risk Since 1994)



Cryptographically Relevant Quantum Computer (CRQC) Realized

*Start Using Post-Quantum Resistant Cryptography **NOW!***

Enables an immediate transition to PQC

- *PQC+classical (e.g. KYBER + McEliece)*
- *2xPQC (e.g. KYBER + HQC)*

OWASP #2: **Cryptographic Failures**

Open Worldwide Application Security Project (OWASP) Top 10 Security Risks for 2023 (https://www.sitelock.com/blog/top-10-owasp-vulnerabilities/)

Data Generated Annually Globally
https://explodingtopics.com/blog/data-generated-per-day

NB: 1 ZB = 1 Billion TB

Attributes of Agility

Agility allows us to future-proof systems against novel cryptanalysis, evolving adversaries, and implementation errors.

- Shortens the time between the demonstration of a vulnerability in an algorithm, implementation, or protocol and the patching or upgrading of all applications and services affected by the vulnerability
- Enables the transition to more efficient algorithms or implementations
- Reduces reaction time through the simplicity of action
- Increases the depth of security through the layering of cryptography throughout the OSI stack
- Synchronizes cryptographic security in time, space, and purpose to produce the greatest possible security at any point in the system.

Algorithmic Agility

"The de facto methodology, in modern work, is to then show that the resulting scheme (*algorithm*), when attacked in a specific cryptographic model, is secure assuming the underlying assumption on the primitive holds." https://www.ecrypt.eu.org/csa/documents/D5.4-FinalAlgKeySizeProt.pdf

Primitives and Parameters (i.e., Key Length):

- Symmetric: block ciphers, hash functions and stream ciphers
- Asymmetric: factoring, discrete logarithms, pairings, lattices, error-correcting codes

Algorithms: RSA, AES, CRYSTALS-KYBER, McEliece, etc...

Implementation Agility

Quickly patching vulnerable algorithm implementations, the ability to access different implementation libraries for the same algorithm, and enabling “fall back” and switching to other algorithms.

For example, a software library may implement an algorithm in a way which is vulnerable (e.g. KyberSlash1 and KyberSlash2)

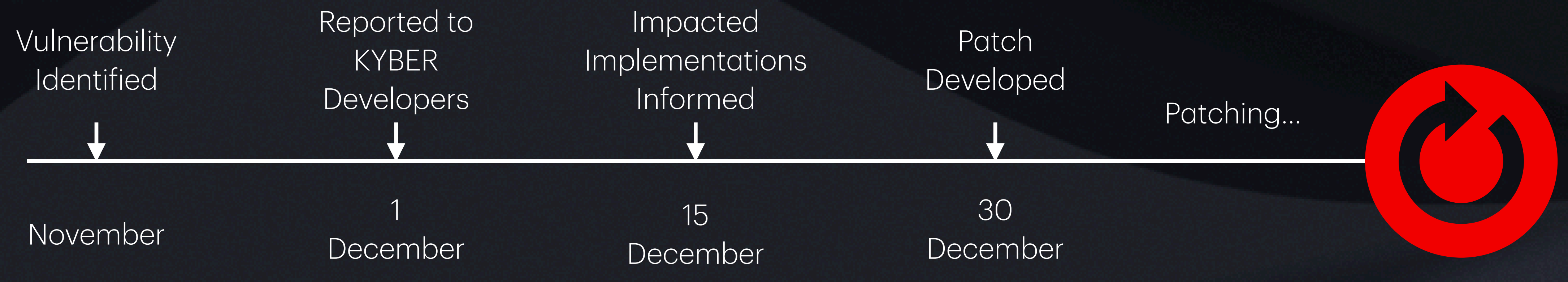
Impacted Implementations and Status as of 10 January 2024:

- pq-crystals/kyber/ref – fully patched
- symbolicsoft/kyber-k2so – fully patched
- aws/aws-lc/crypto/kyber, main branch – fully patched
- zig/lib/std/crypto/kyber_d00.zig – fully patched
- kudelskisecurity/crystals-go – patched on January 10
- liboqs/src/kem/kyber – patched only for KyberSlash1
- aws/aws-lc/crypto/kyber, fips-2022-11-02 branch – patched only for KyberSlash1
- randombit/botan – patched only for KyberSlash1
- mupq/pqm4/crypto_kem/kyber – patched only for KyberSlash1
- antontutoveanu/crystals-kyber-javascript – unpatched
- Argyle-Software/kyber – unpatched
- debian/src/liboqs/unstable/src/kem/kyber – unpatched
- PQClean/PQClean/crypto_kem/kyber/aarch64 – unpatched
- PQClean/PQClean/crypto_kem/kyber/clean – unpatched
- rustpq/pqcrypto/pqcrypto-kyber (used in Signal) – unpatched

https://www.flickr.com/photos/jonnyb558/7875495274

Implementations NOT Impacted:

- boringssl/crypto/kyber
- filippo.io/mlkem768
- formosa-crypto/libjade/tree/main/src/crypto_kem/kyber/common/amd64/avx2
- formosa-crypto/libjade/tree/main/src/crypto_kem/kyber/common/amd64/ref
- pq-crystals/kyber/avx2
- pqclean/crypto_kem/kyber/avx2

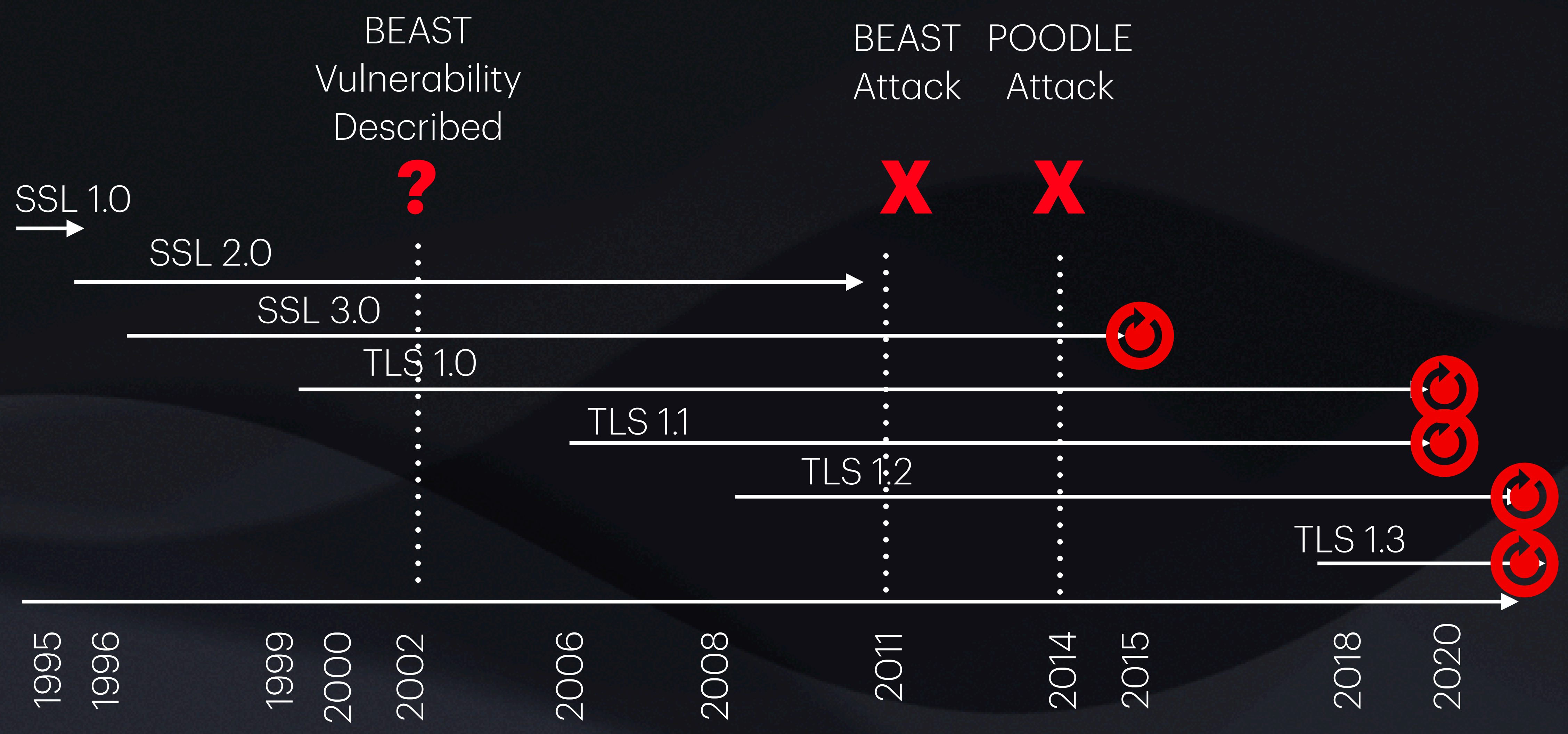


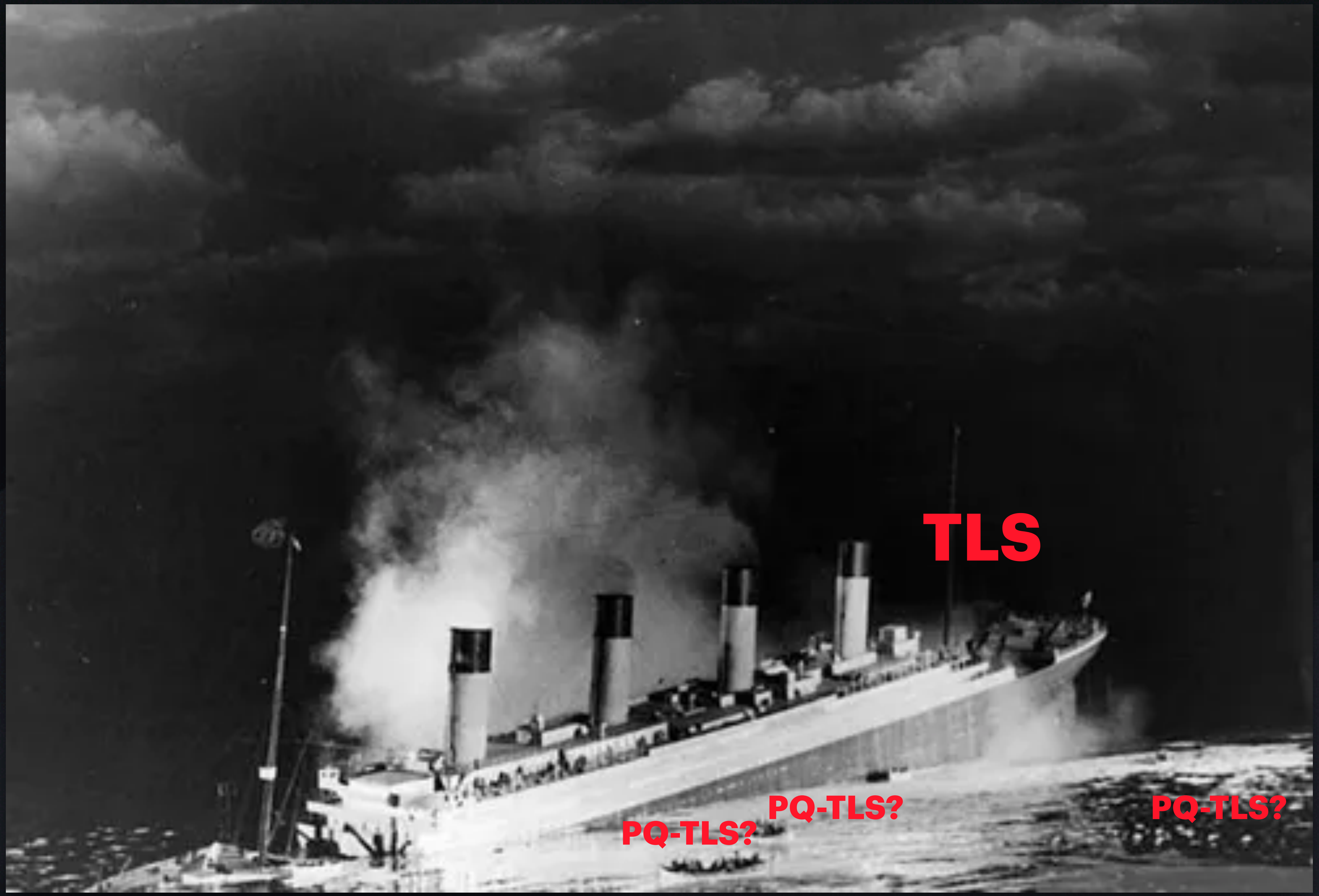
>90 Days Exposure

Protocol Agility

Protocols: TLS 1.x, SSL x.0, etc...

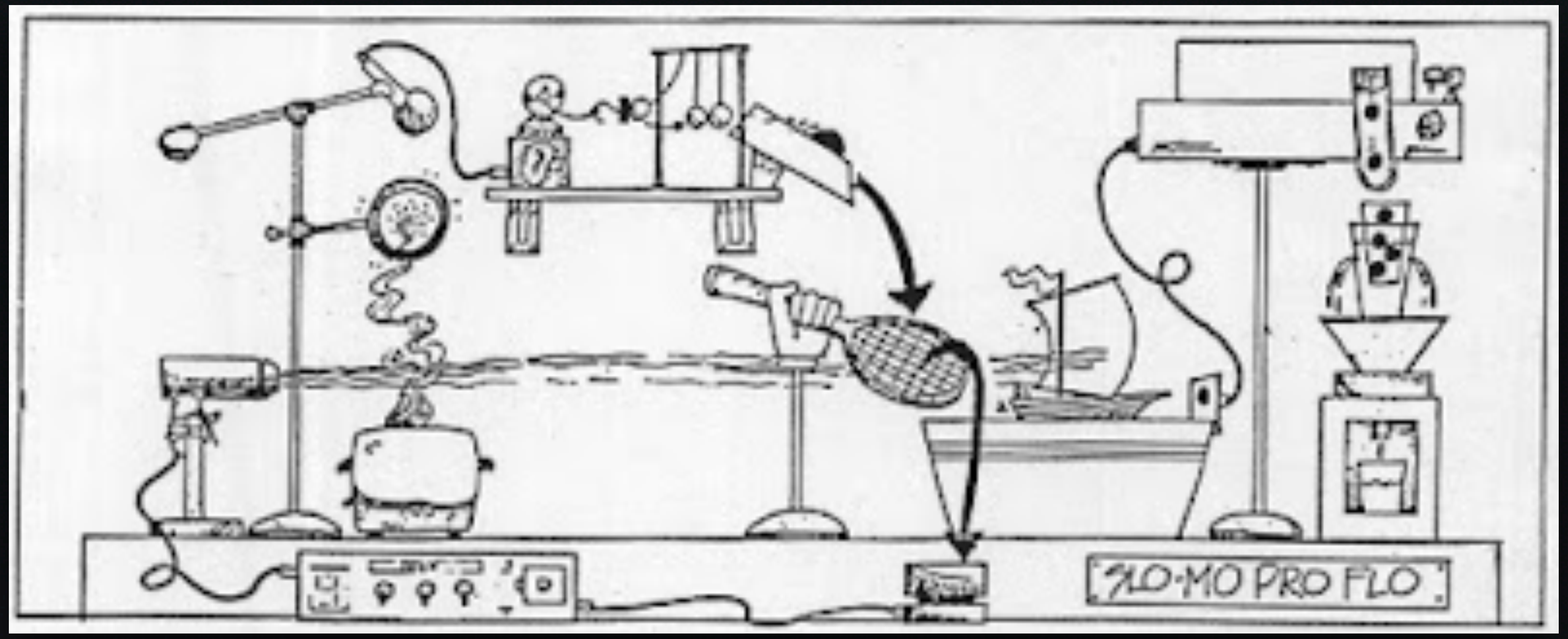
- Initially, the RC4 cipher was recommended for use to mitigate BEAST attacks (because it is a stream cipher, not a block cipher). However, RC4 itself was later found to be unsafe for use and is now prohibited in the PCI DSS (Payment Card Industry Data Security Standard).
- Quickly upgrade from a vulnerable protocol version, such as migrating from TLS 1.0 to TLS 1.3.
- Facilitate Protocol Tunneling





Obstacles to Agility

- Decentralized, fragmented deployment model of cryptography.
  - Eliminating all SHA-1 TLS certificates means tracking down all certificates on every single endpoint within your network.
  - Cryptography is usually tightly coupled to applications.
    - Configuring TLS for an Apache web server with mod ssl differs from deploying NGINX with TLS, and differs from configuring TLS for Redis, MongoDB, and...etc.
    - Replacing algorithms using long-deprecated standards (e.g., RC4, MD5, DES). In the case of TLS, the deprecation period has been on the order of years.
    - Legacy applications (i.e., FORTRAN) have embedded cryptographic subroutines which are not easily discovered.
- Baked directly into a secure channel protocol by means of a negotiation phase.
  - Increases complexity in a protocol making security harder to reason about and can introduce vulnerabilities through misconfiguration.



https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.loupdargent.info%2F2013%2F04%2F5-amazing-rube-goldberg-machines.html&psig=AOvVaw1kgnUqOt0M2GCez2hl8YYc&ust=1709863131612000&source=images&cd=vfe&opi=89978449&ved=0CBMQjhxqFwoTCKiwvteG4YQDFQAAAAAdAAAAABAf

OWASP #4: **Insecure Design**
OWASP #5: **Security Misconfiguration**

Obstacles to Agility

- Dependency on providers for the timely patching of vulnerabilities or upgrading.

- Migration to new cryptography hindered by cryptographic silos across product lines

- Continuous monitoring and correction of cryptographic configurations
  - Operators or automated tools may inadvertently reconfigure devices
  - CBOM is a snapshot in time
  - Traffic bypassing IPsec tunnels
  - IPsec in a non-PFS mode (Perfect Forward Security) I.e., hidden default settings

- Updating protocols that rely on asymmetric key exchange:
  - Application Layer: MFT, SFTP, SSH, SKIP
  - Transport Layer: TLS, DTLS, QUIC
  - Internet Layer: IPsec (uses IKEv2)
  - Data Link Layer: MACSec uses 802.1x (EAP-TLS)

OWASP #6: **Vulnerable and Outdated Components**
OWASP #9: **Security Logging and Monitoring Failures**

Enabling Agility

- Abstractions and decoupling
  - Service Mesh Proxying
  - Software Defined Networking
- Continuous Monitoring of Cryptographic Configurations in Each Layer of the TCP/IP Stack

Conclusion

Agility enables the future-proofing of our technology baseline without the need to “rip and replace”

Agility expedites our ability to respond to cryptographic vulnerabilities and implementation errors

Agility facilitates the adoption of new algorithms, implementations, and protocols

Agility Accelerates Action

Thank You!

Aaron Moore, CTO
QuSecure
aaron@qusecure.com