

Why do we need a SKEX WG?

Mattia Montagna

Challenges with Public-Key Cryptography & Public Key Infrastructure

- Vulnerability to quantum computers
 - Traditional public-key cryptography (PKC) methods (RSA, ECC) rely on mathematical problems that Shor's algorithm can solve efficiently, making them breakable by quantum computers.
- Computational and resource overhead
 - PKC requires high CPU power and memory, making it inefficient for IoT and constrained devices.
 - Resource-heavy cryptographic operations slow down authentication processes, affecting scalability and performance.
- Complexity & operational overhead
 - Public Key Infrastructure (PKI) requires:
 - Certificate issuance, validation, and revocation.
 - Centralized trust models that create single points of failure.
 - Certification chains create several single-points of failure for any certificate.
 - Heavy administrative and security burdens.

Challenges with Post-Quantum Cryptography (PQC)

Security Uncertainty

- PQC relies on computational hardness assumptions, which could be broken by future algorithmic advances.
- No proven long-term guarantees. NIST's PQC competition has already seen multiple proposals broken.

Performance and Implementation Risks

- Larger key sizes and increased computational overhead make PQC impractical for:
 - IoT, embedded systems, and resource-constrained devices.
- More complex algorithms increase the attack surface $\Rightarrow$ higher risk of implementation flaws and side-channel attacks.

Challenges with Kerberos

Why Kerberos is not a counterexample:

- Kerberos is a key management/exchange protocol, not a symmetric key establishment protocol: It uses a trusted third-party Key Distribution Center (KDC) to provide session keys but does not establish long-term symmetric keys between entities.
- Scalability challenges: Kerberos relies on centralized KDCs, introducing single points of failure or operational complexities and operational overhead when scaling to large, distributed, or cross-domain environments.
- Limited applicability: Kerberos is primarily used for user authentication in enterprise environments, not for general symmetric key distribution across networks, especially for scenarios like machine-to-machine or post-quantum-secure key exchange.
- Cross-domain trust issues: Extending Kerberos beyond a single administrative domain requires complex trust relationships, limiting its usability for broader symmetric key distribution needs.

The case for Symmetric Key Establishment protocols

Overcome asymmetric cryptography challenges

- Resistant to both quantum and classical attacks $\Rightarrow$ based on information-theoretic principles.

Lightweight & Efficient

- Works on low-power devices, IoT sensors, routers, and constrained environments.
- Minimal computational and memory footprint compared to PKC & PQC.

Scalable & Automated Key Distribution

- Replaces manual PSK provisioning with a secure, dynamic key establishment mechanism.

Seamless Integration

- Plugs into existing security protocols (IPsec, TLS, MACsec, TCP-AO).

No prior standardisation has been done

- Symmetric key distribution (Kerberos) is standardized and in use; however no existing standardized solution addresses scalable symmetric key establishment across networks without a central “all-knowing” authority.

Symmetric Key Establishment protocols offer a quantum-safe, efficient, and scalable solution to the key distribution problem that PKC and PQC fail to address.

Use Cases for Symmetric Key Distribution

Diego R. Lopez / Hooman Bidgoli

Symmetric key based protocols

- Some protocols are only symmetric key based (e.g., TCP-AO)
- PSK and symmetric key distribution are preferentially used in other protocols (e.g., MACsec, IPsec in managed security environments)
- Issue(s)
 - Huge number of symmetric key to generate/manage (e.g., existing scenario of around 1500 symmetric keys) => need of a centralized symmetric key provisioning/management

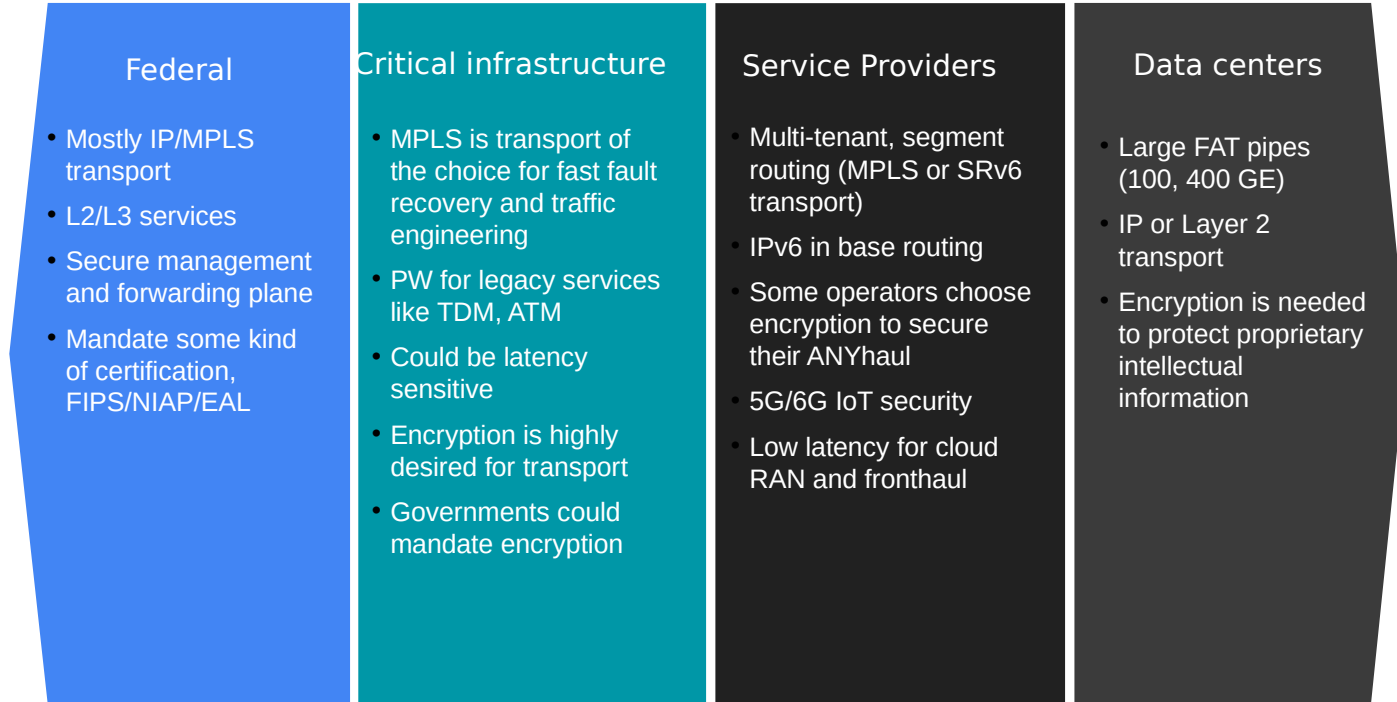
Post-quantum transition - authentication

- Key exchange mechanisms need to be authenticated, with either certs or Pre-Shared Key (PSK), to avoid Man-in-the-Middle attacks
- Issue(s)
 - Even if Post-Quantum Cryptography (PQC) algos will be “quickly” integrated inside protocols using traditional cryptography (e.g., IPsec/IKEv2, TLS), transition from a traditional PKI use to a PQC based PKI use could take time => need to use PSK for authentication
 - This is especially applicable to legacy environments, related to device-to-device communications

Post-quantum transition - IoT

- IoT devices prefer light crypto algos (e.g., ECC) when using asymmetric crypto
 - Avoid PKI additional processing implications
 - Managed environments, with (very) limited personal data involved
- Issue(s)
 - PQC algorithms, either standardized or candidate for a standardization, are not light (i.e., public key/signature size, CPU time) => need to do a transition to symmetric crypto (cf. first use case) ?

Networks in Need of Strong Security

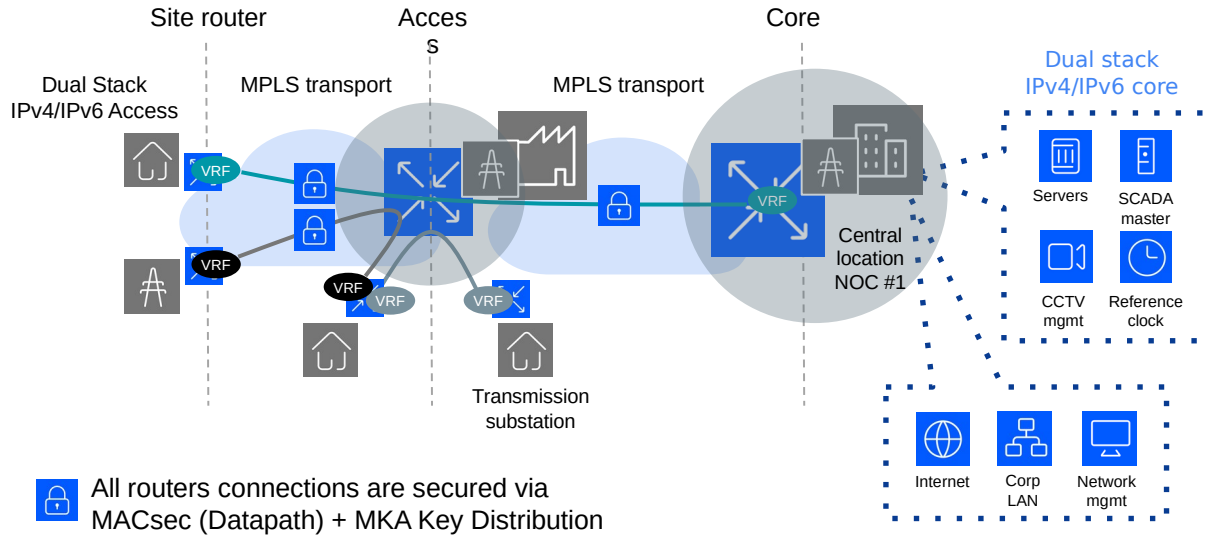


All Interested in strong encryption for EXISTING Transport and SIMPLE Key Distribution and Management
PKI and Certificates are complicated for these operators

Simplicity Wins the Race for Security

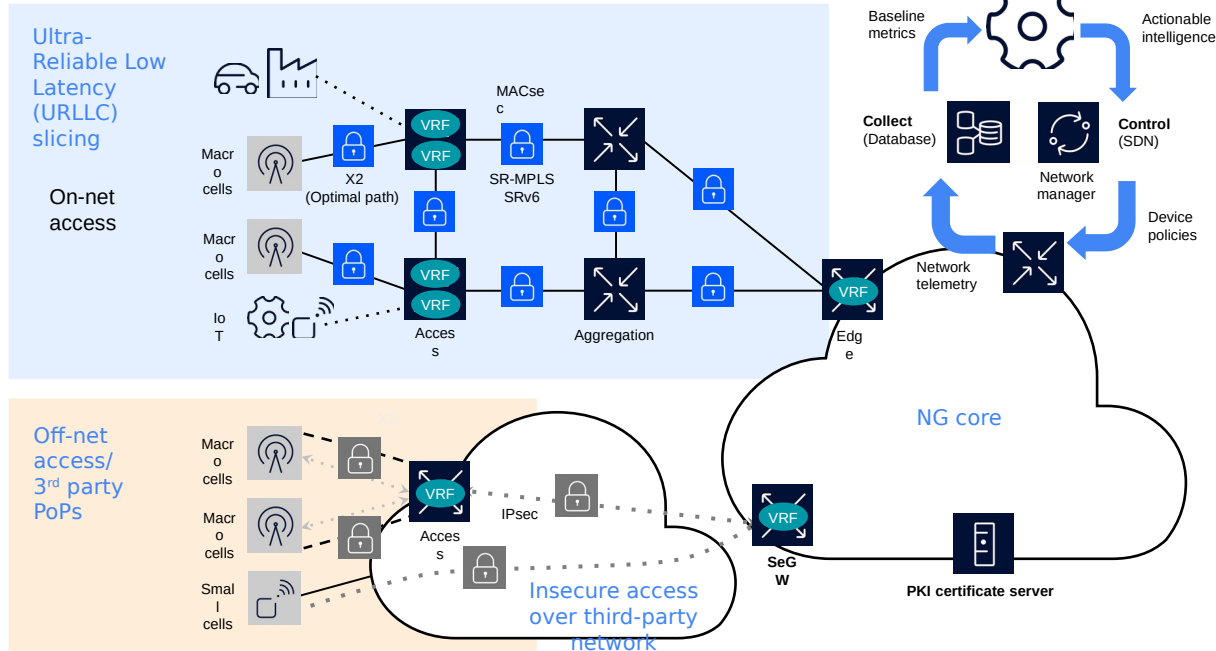
Critical Infrastructure

MPLS is the transport of the choice



- Simple Encryption protocol and Key Management is preferred by most operators
- Example, MACsec with MACsec Key Agreement. Uses PSK and AES-CMAC to secure the Datapath Key (SAK)

Service providers

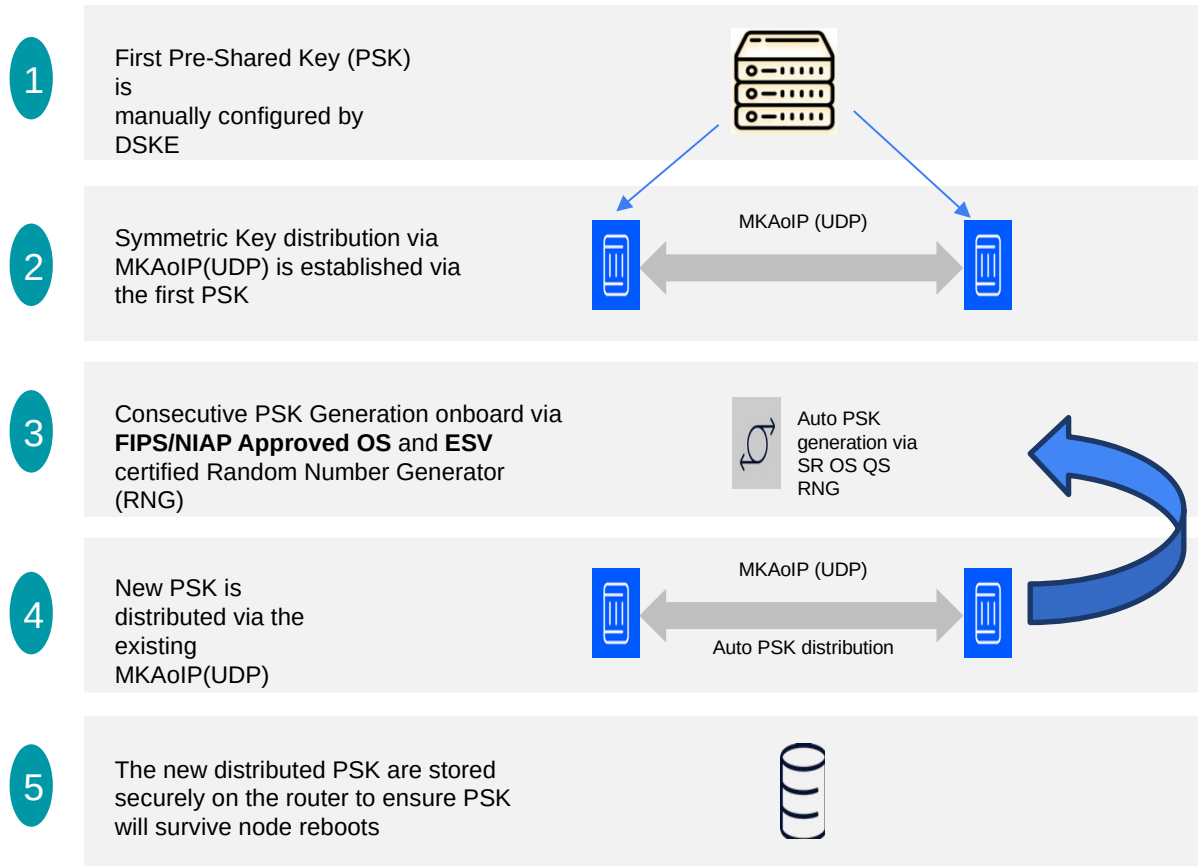


- Mandates to secure the Mobile Backhaul because of Critical Services like IoTs
- Some vendors use IPsec with PKI, but they ask for simplified solutions
- Others prefer MACsec because of simplified Key Distribution

 All routers connections are secured via MACsec (Datapath) + MKA Key Distribution

 IPsec with PKI

PSK Automation



- Simplicity of Key Distribution via Symmetric Key is preferred, as an example PSK and AES to secure the datapath Key.
- New protocols for Symmetric Key Distribution and Management are needed in IETF
- High Entropy and Automation of Symmetric Key Distribution is the key to success
- Perhaps reinventing the wheel is not necessary and reusing existing protocols like MKA with minor tweaks can be beneficial (MKA over IP/UDP [draft-hb-intarea-eap-mka-00](#))

Interactions and interfaces with IPsec, TLS, MACsec, etc.

Melchior Aelmans

Suggestions are there but no infra to supply the keys 1/2

Many RFCs suggest the use for symmetric keys, for example in TLS:

- [RFC9257](#) – “This document provides usage guidance for external Pre-Shared Keys (PSKs) in Transport Layer Security (TLS) 1.3”
- [RFC9258](#) – This document describes an interface for importing external Pre-Shared Keys (PSKs) into TLS 1.3.
- [draft-ietf-tls-8773bis](#) – This document specifies a TLS 1.3 extension that allows TLS clients and servers to authenticate with certificates and provide confidentiality based on encryption with a symmetric key from the usual key agreement algorithm and an external pre-shared key (PSK).

Suggestions are there but no infra to supply the keys 2/2

Use of Pre-Shared Keys in IPsec:

- RFC8784 – “This document describes an extension of IKEv2 to allow it to be resistant to a quantum computer by using preshared keys.”

Use of Pre-Shared Keys in TCP-AO:

- RFC5925 – “TCP-AO is compatible with either a static Master Key Tuple (MKT) configuration or an external, out-of-band MKT management mechanism.”

BGP Operations and Security BCP:

- RFC7454 – The drawback of TCP session protection is additional configuration and management overhead for the maintenance of authentication information (for example, MD5 passwords).

What is and isn't there

Despite the increasing number of references in RFCs for (internet) scalable symmetric key distribution, there is currently no standardized mechanism that efficiently extends beyond the n^2 scaling challenge.

The interfaces to consume Key material from Symmetric Distribution mechanisms are there and implemented:

- ETSI GS QKD 014: The document describes a communication protocol and data format to supply cryptographic keys to an application. It is intended to allow interoperability of equipment from different vendors.
- Secure Key Integration Protocol (SKIP): This document specifies SKIP, a two-party protocol that allows a client to securely obtain a key from an independent Key Provider. SKIP enables network and security operators to provide quantum-resistant keys suitable for use with quantum-resistant cryptographic algorithms such as AES-256. It can also be used to provide an additional layer of security to an already quantum-resistant secure channel protocol for a defense-in-depth strategy, and/or enforce key management policies.

SKEX protocols – deliverable 1

Unmediated Symmetric Key Establishment

Daniel Shiu

https://mailarchive.ietf.org/arch/msg/skex/8Vgb_sHVjCunIAaNS7P7PTYNfMs/

The Brief

A point-to-point symmetric key establishment with no mediation.

Assumes that the parties have previously shared a secret credential K_{AB} that is unknown to the adversary.

Some Prior Art

ISO/IEC 11770-2 IT Security techniques - Key management - Mechanisms using symmetric techniques

Describes 13 methods: 1–6 point-to-point; 7–10 key distribution; 11–13 key translation

Measure practicality: passes, key derivations, encryptions/decryptions

Notes some security properties: key control, key authentication, replay detection, key confirmation, entity authentication

Example (key distribution): Kerberos

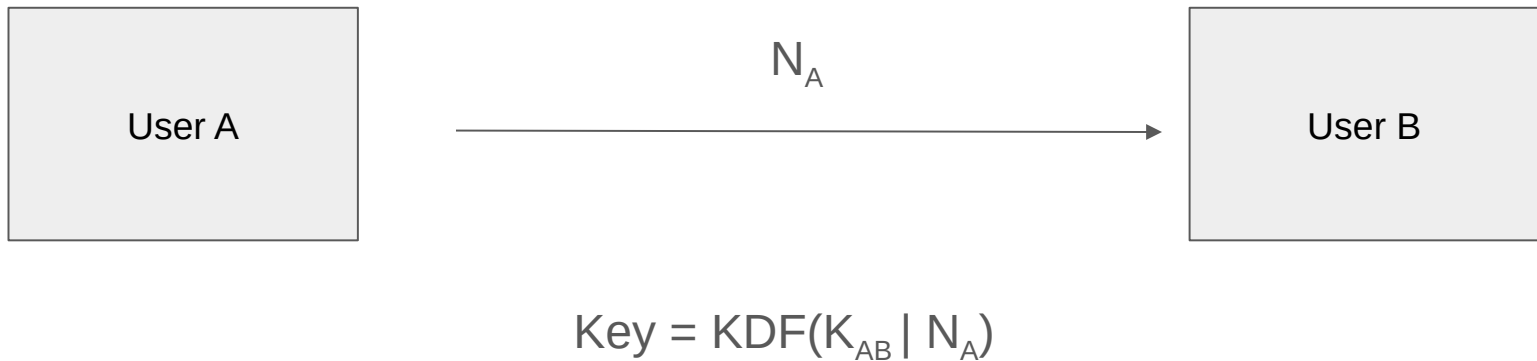
Combination of method 7 (client – authenticating server – ticketing server)

and method 8 (client – ticketing server – service server)

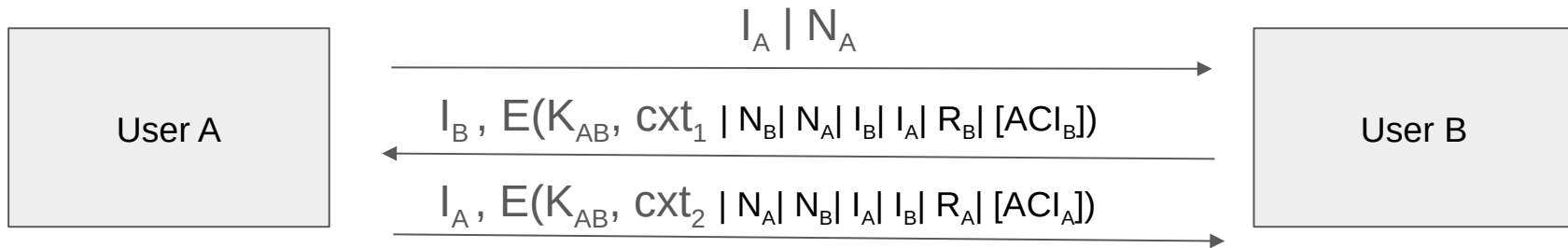
Practicality: 6 passes, 14 encryption/decryptions

Security Properties: correctness, implicit two way authentication, key confidentiality, implicit key confirmation, replay detection

Minimal point-to-point protocol



USKE draft-dkls-uske-00



$$k = \text{KDF} (\text{cxt}_3 | K_{AB} | N_A | N_B | I_A | I_B | R_A | R_B | [\text{ACI}_A] | [\text{ACI}_B])$$

$$\text{Optional: } \text{NK}_{AB} = \text{KDF} (\text{cxt}_4 | K_{AB} | N_A | N_B | I_A | I_B)$$

USKE properties

Practicality: 3 passes, 4 encryptions/decryptions, 2 derivations

Security Properties: correctness, key confidentiality, two-way identity authentication, injective key agreement, key freshness, key integrity, known key security, [optional] forward security

SKEX protocols – deliverable 2

Distributed Symmetric Key Establishment draft-mwag-dske

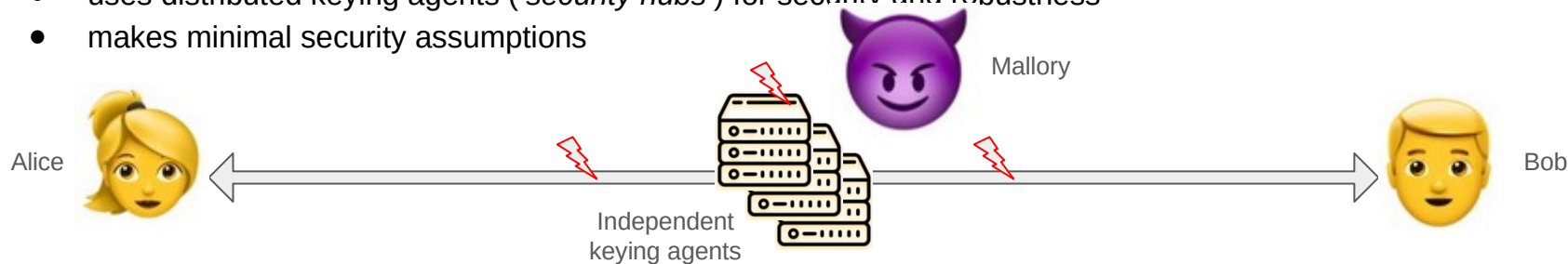
Mattia Montagna, Manfred von Willich,
Gert Grammel, Melchior Aelmans

<https://datatracker.ietf.org/doc/draft-mwag-dske/>

Overview of DSKE: Setting the scene

Distributed Symmetric Key Establishment (DSKE) is a key establishment protocol that allows Alice and Bob to establish a symmetric key between only them, and

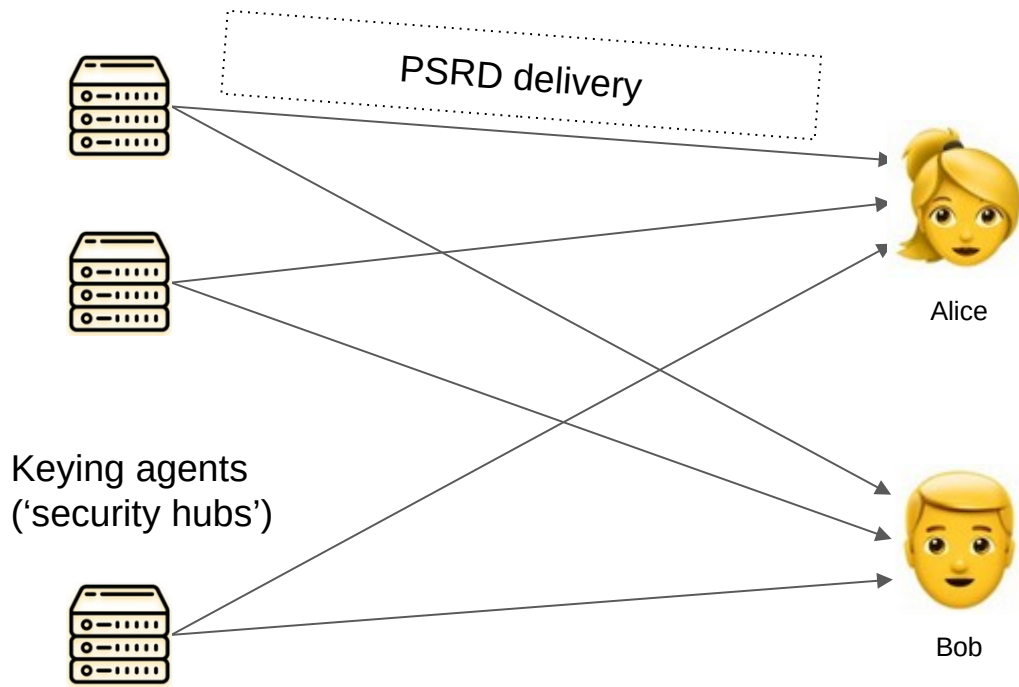
- is mathematically simple, with provable security
- uses distributed keying agents ('security hubs') for security and robustness
- makes minimal security assumptions



Properties:

- **Security:** Confidential and correct when no more than k keying agents are compromised
- **Robustness:** Completes when a lower bound on number of correctly operating paths holds
- **Scalability:** Costs and logistics are linear in the system size

Review of DSKE: client onboarding



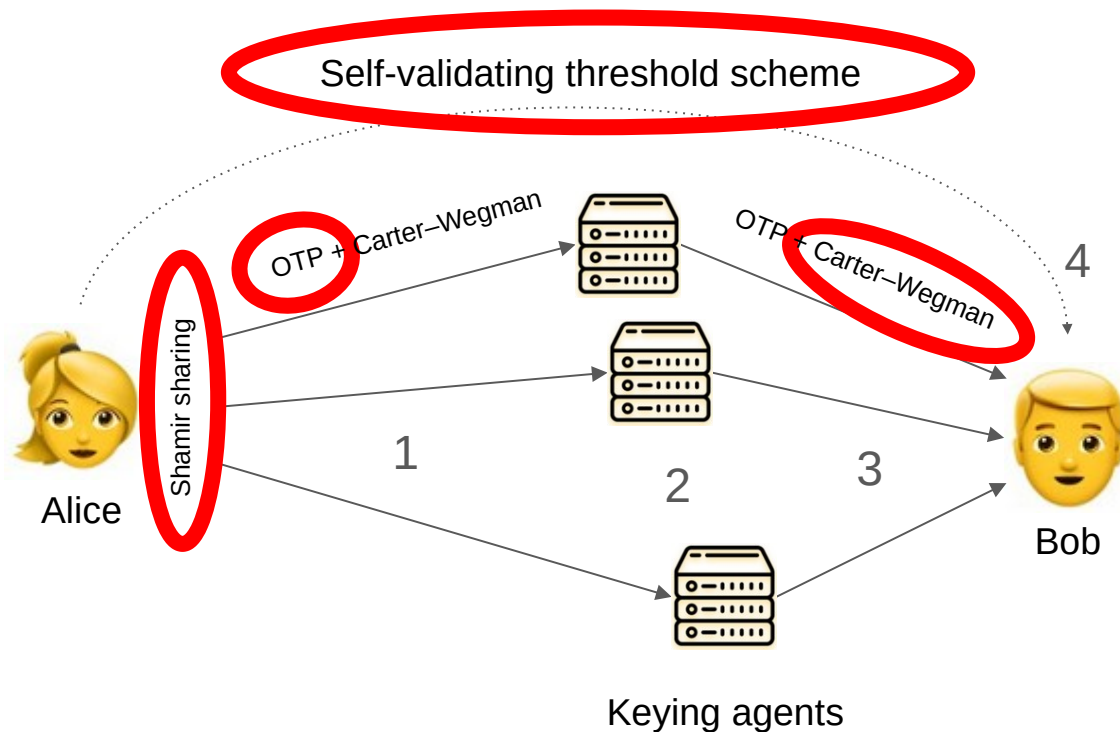
Client onboarding

- Delivery of Pre-Shared Random Data (PSRD) to clients by each independent keying agent
- Securely established client identity tied to delivery
- Activation of PSRD once the delivery is verified and acknowledged

Note:

- Onboarding is mutually independent
- Cost and complexity are linear in number of clients (per keying agent), and in the number of keying agents (per client)

Review of DSKE: key establishment



Key creation:

1. Alice and Bob need to determine the other's ID as assigned by the agents
2. Alice sends a key share to each keying agent
3. Key agents compute key instructions, forwarded to or fetched by Bob
4. Bob rebuilds same key as Alice using Shamir scheme
5. Bob validates secret and associated data

– DSKE is a composition of primitives –

DSKE: Summary

Takeaways:

- Classical and quantum-safe
 - information-theoretic security
- Very low resource requirements
- No “all knowing” central nodes
 - No single point of failure
 - Fault tolerance
- Scalable
 - PSRD distribution can be integrated in existing processes

Draft submitted to datatracker: <https://datatracker.ietf.org/doc/draft-mwag-dske/>

Summary of need for SKEX WG

- No standardized method for symmetric key distribution beyond n^2 scaling.
- Traditional pre-shared key (PSK) methods require pairwise exchanges, making them impractical for large networks (IoT, 5G, service providers).
- Existing solutions (e.g., Kerberos, KDC-based mechanisms) introduce single points of failure and potentially lack post-quantum resilience.
- Proprietary implementations exist but lack interoperability and standardization.
- SKEX BoF aims to work on scalable, standardized symmetric key exchange mechanisms for large and diverse networks.

Charter discussion

BoF Questions / Discussion