KERI: 2
Universal DKMI

Samuel M. Smith Ph.D.
IIW Fall 2019 2019/10/01
https://github.com/SmithSamuelM/Papers
sam@samuelsmith.org

Background Reading

Girault, M., “Self-certified public keys,” EUROCRYPT 1991: Advances in Cryptology, pp. 490-497, 1991

https://link.springer.com/content/pdf/10.1007%2F3-540-46416-6_42.pdf

Kaminsky, M. and Banks, E., “SFS-HTTP: Securing the Web with Self-Certifying URLs,” MIT, 1999

https://pdos.csail.mit.edu/~kaminsky/sfs-http.ps

Mazieres, D. and Kaashoek, M. F., “Escaping the Evils of Centralized Control with self-certifying pathnames,” MIT Laboratory for Computer Science, 2000

http://www.sigops.org/ew-history/1998/papers/mazieres.ps

Mazieres, D., “Self-certifying File System,” MIT Ph.D. Dissertation, 2000/06/01

https://pdos.csail.mit.edu/~ericp/doc/sfs-thesis.ps

Smith, S. M., “Open Reputation Framework,” vol. Version 1.2, 2015/05/13

https://github.com/SmithSamuelM/Papers/blob/master/whitepapers/open-reputation-low-level-whitepaper.pdf

Smith, S. M. and Khovratovich, D., “Identity System Essentials,” 2016/03/29

https://github.com/SmithSamuelM/Papers/blob/master/whitepapers/Identity-System-Essentials.pdf

Smith, S. M., “Decentralized Autonomic Data (DAD) and the three R’s of Key Management,” Rebooting the Web of Trust RWOT 6, Spring 2018

https://github.com/SmithSamuelM/Papers/blob/master/whitepapers/DecentralizedAutonomicData.pdf

Smith, S. M., “Key Event Receipt Infrastructure (KERI) Design and Build,” arXiv, 2019/07/03

https://arxiv.org/abs/1907.02143

Human Basis of Trust

I know you

or

I know of you

therefore

I trust you

On the internet I can't really know who you are.
therefore
I can't trust you

Decentralized Control Authority

Assymetric PKI: Public-Private Key Pairs with Digital Signatures

CSPRNG Cryptographic Strength Pseudo Random Number Generator

Collision Resistant Random Seed (Entropy) Available to Anyone

Seed -> Private Key -> One Way Function -> Public Key

Authority comes from collision resistance

Inherently decentralized

Sole Sovereign over random seed

Only one who can make verifiable signed statements associated with the Public Key

Cryptographic Root-of-Trust

Trust *who said it* not *what was said*

Consistent attribution is the root-of-trust (integral non-repudiable statements)

Duplicity detection

I trust that controller of private key made a set of statements

Build trust in *what was said* via consistent history of consistently attributable statements.

Self-Certifying Identifier/Namespace

Use public key in identifier

Use public key as prefix in namespace

Decentralized Root of Trust

Signed Statements that include self-certifying identifier

Self-Certifying Namespace

Provenanced chain of transformations with verifiable control over transformation

End-wise Verifiable (primary root-of-trust)

Other roots of trust may add to but not replace self-certification

All decentralized infrastructure has self-certifying identifiers as primary root-of-trust

Decentralized key management is therefore essential to protecting infrastructure

AUTONOMIC NAMESPACE

Self-Certifying

Self-Managing

Self-Administering

Extensible

KEY MANAGEMENT

Rotation

Reproduction

Recovery

KEY MANAGEMENT

Best Practice:

One-use: One-time One-place One-way

Key Rotation

Key rotation is useful when the controller of a self-certifying identifier needs to maintain persistent control over that identifier indefinitely despite exploits of the private key(s).

Otherwise in the event of exploit, the controller could just abandon the exploited identifier and create a new identifier with a new (public, private) key pair.

Key Rotation

After rotation the public key associated with the identifier is not changed, but merely the private key(s) that is authoritative is changed. Otherwise the identifier loses its value as an identifier.

After rotation the original private key has been revoked and replaced with the new private key(s) specified in the rotation operation.

The initial rotation operation must be signed at the very least with the original private key.

The new public key is then included in the identifier's key rotation history.

The latest rotation entry provides the current authoritative key pair.

In order to verify an attestation belonging to a self-certifying identifier the verifier must lookup and validate the key rotation history for that identifier.

Nomenclature

self-certifying identifier: includes public key

digital signature: unique non-repudiable (cypher suite known)

digest: collision resistant hash of content

signed digest: commitment to content

controller: controlling entity of identifier

message: serialized data structure

event: actionable message

key event: key management operation

More Nomenclature

inception event: unique self-signed event that creates identifier and controlling key(s)

rotation event: self-signed uniquely ordered event from a sequence that changes the set of controlling keys

verifier: cryptographically verifies signature(s) on an event message.

witness: entity that may receive, verify, and store key events for an identifier. Each witness controls its own identifier used to sign key event messages, *controller* is a special case of witness.

receipt: event message or reference with one or more witness signatures

Even More Nomenclature

key event log: ordered record of all self-signed key event messages

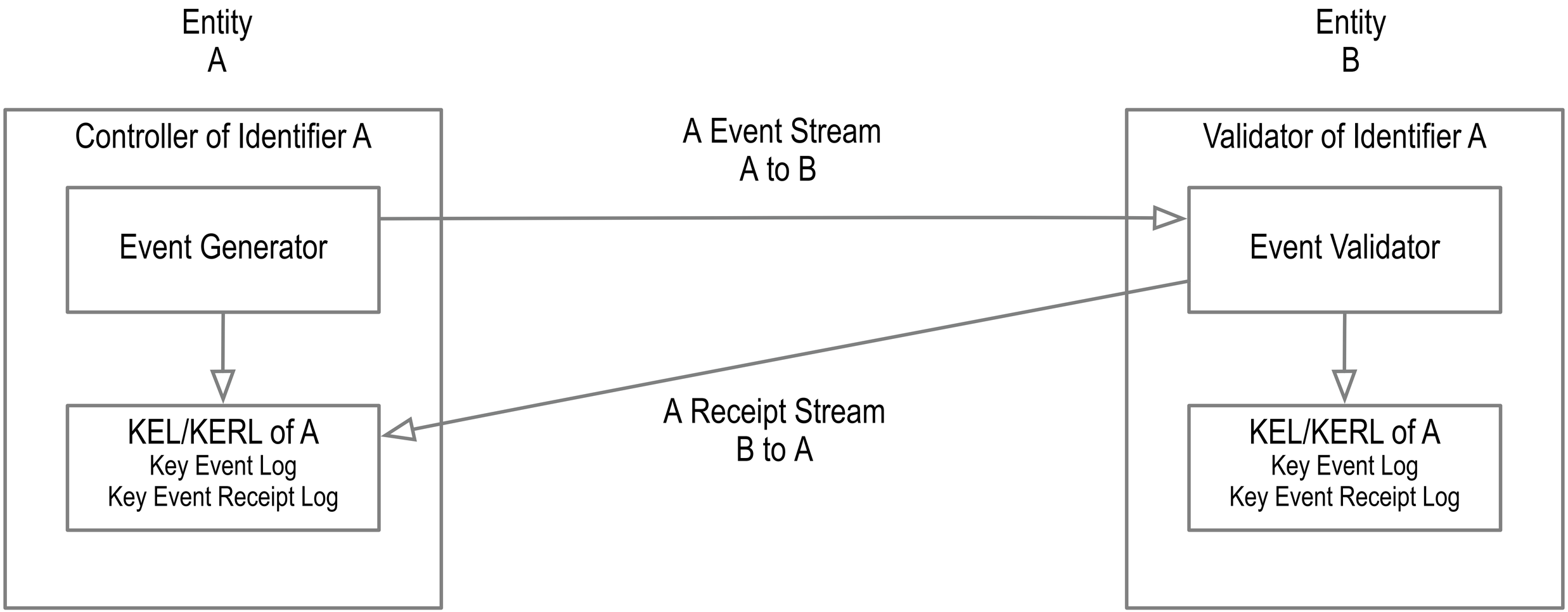
key event receipt log: ordered record of all key event receipts for a given set of witnesses

validator: determines current authoritative key set for identifier from at least one key event (receipt) log.

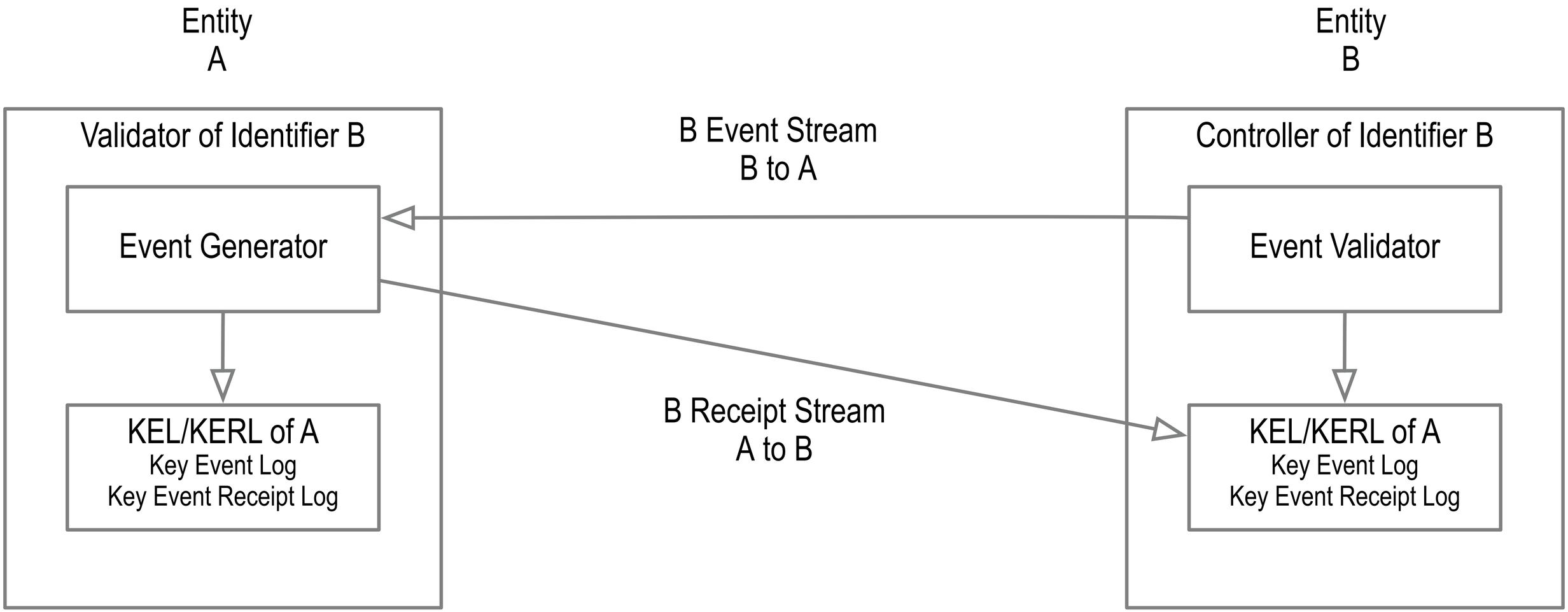
judge: determines current authoritative key set for identifier from the key event receipt logs from a set of witnesses.

pre-rotation: commitment to next rotated key set in previous rotation or inception event

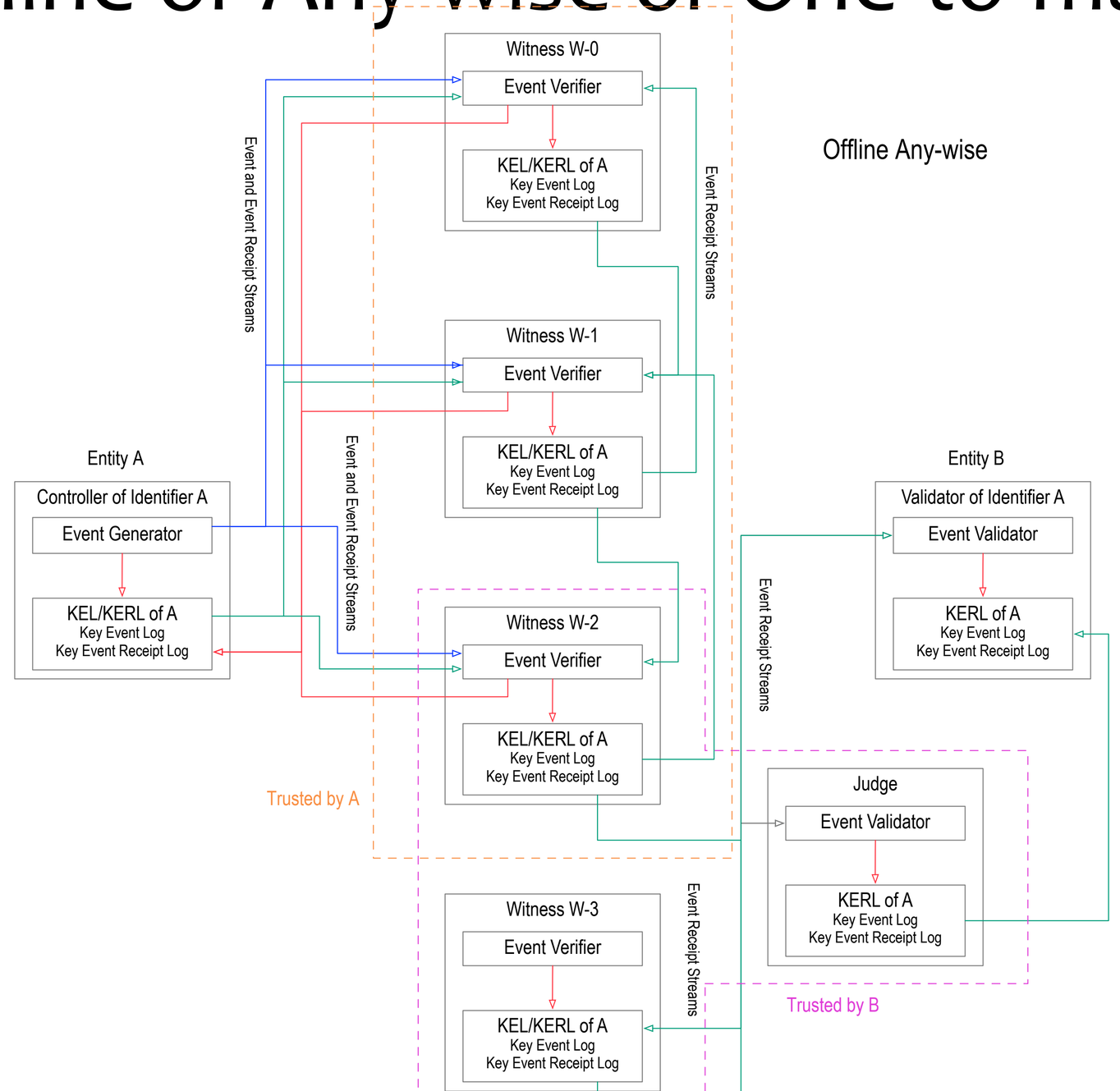
Online or Pair-wise or One-to-One: A to B



Online or Pair-wise or One-to-one: B to A



Offline or Any-wise or One-to-many



Offline Trustworthy Service

Service provided by *N* designated *witnesses* and at least one *judge*.

Service provides a *correct* or at least a *complete* KERL to any validator that requests it.

Service availability means at least one *correct* or *complete* KERL is extant

key state validation mechanism not distributed consensus

Complete or Correct

Witnesses only signs and stores first received and verified version of an event at a given location in full event sequence.

Witnesses provide an immutable KERL of all receipts from any witnesses for the version of each event in its log.

M of N Designated Witnesses

Complete if KERL has M+1 receipts for event

Correct if M is a majority of N

Complete or Correct

Witnesses only signs and stores first received and verified version of an event at a given location in full event sequence.

Witnesses provide an immutable KERL of all receipts from all witnesses for the version of each event in its log.

M non-faulty of N Designated Witnesses

Complete if KERL has M+1 receipts for event

Complete is Correct if M is a majority of N

Scales with 2N vs BA distributed consensus that scales with N^2

Variants

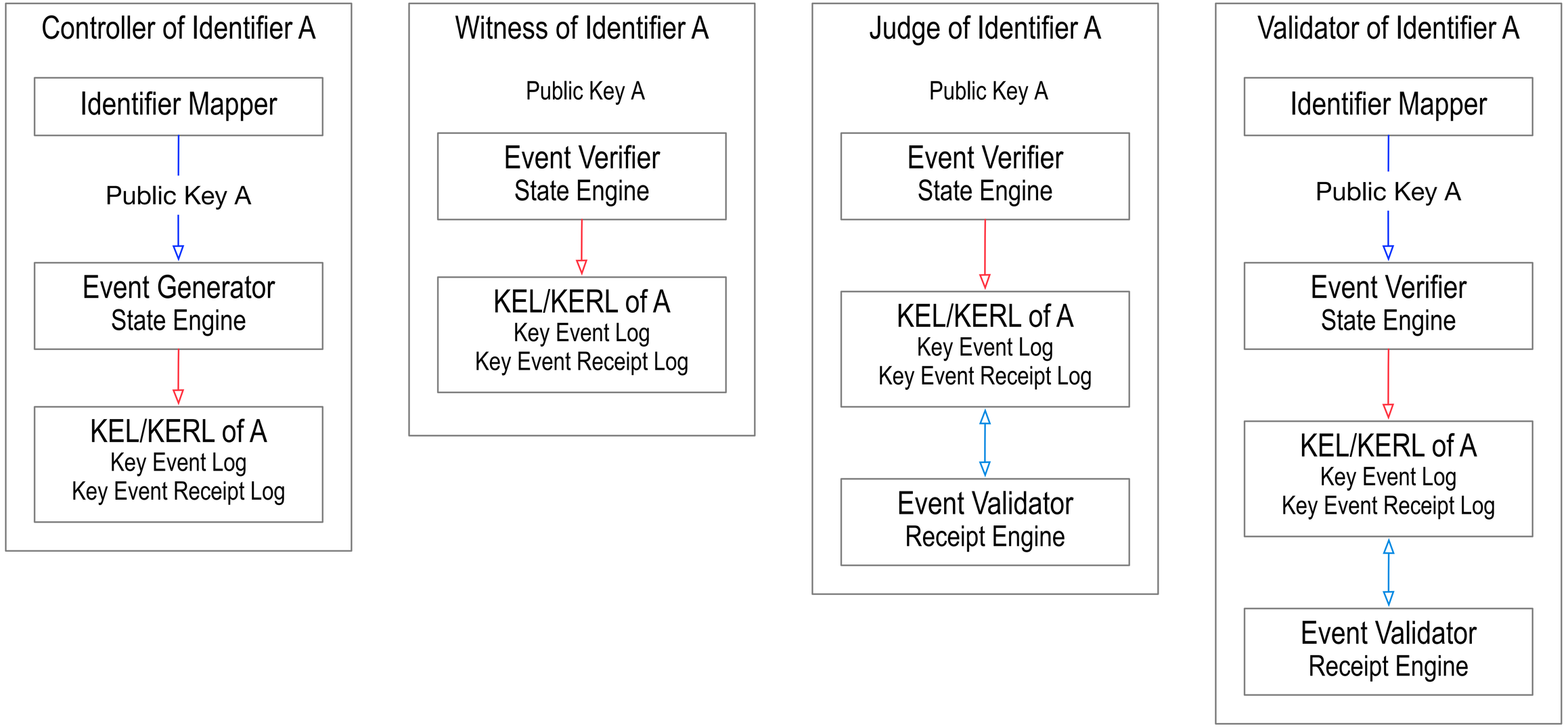
Single Key Sequence for Rotation and Signing

Single Interleaved Key Sequence for Double Rotation and Signing

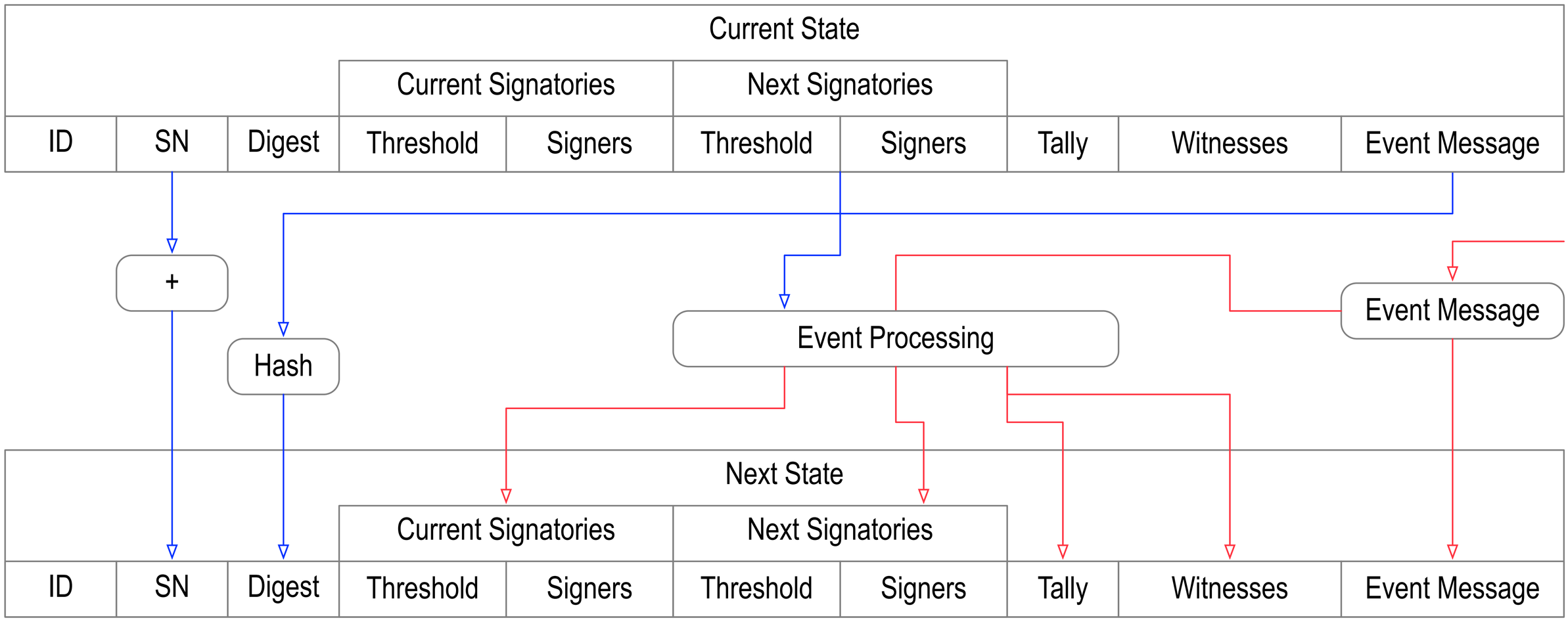
Single Key Sequence for Rotation with multiple Delegated Key Sequence(s) for Signing

Dual Key Sequence for Separate Rotation and Signing

Architecture



State Verifier Engine



Generic Inception

$$\varepsilon_0 = \left\langle C, t_0, \texttt{icp}, C^0, K_1, \widehat{C}_1, M_0, \widehat{W}_0 \right\rangle \sigma_{C^0}$$

$$\widehat{C}_1 = \left[C^1, \ldots, C^{L_1} \right]_1$$

$$\widehat{W}_0 = \left[W_0, \ldots, W_{N-1} \right]_0$$

Generic Rotation

$$\varepsilon_k = \left\langle C, t_k, \eta\left(\varepsilon_{k-1}\right), \texttt{rot}, K_{l+1}, \widehat{C}_{l+1}, M_l, \widehat{X}_l, \widehat{Y}_l, \{data\}, \widehat{e}_l, \widehat{s}_l \right\rangle \widehat{\sigma}_l$$

$$\widehat{C}_{l+1} = \left[C^{r_{l+1}}, \ldots, C^{r_{l+1}+L_{l+1}-1} \right]_{l+1}$$

$$\widehat{X}_l = \left[X_0, \ldots, X_{O_l-1} \right]_l$$

$$\widehat{Y}_l = \left[Y_0, \ldots, Y_{P_l-1} \right]_l$$

$$\widehat{e}_l = \left[e_0, \ldots, e_{E_l-1} \right]_l$$

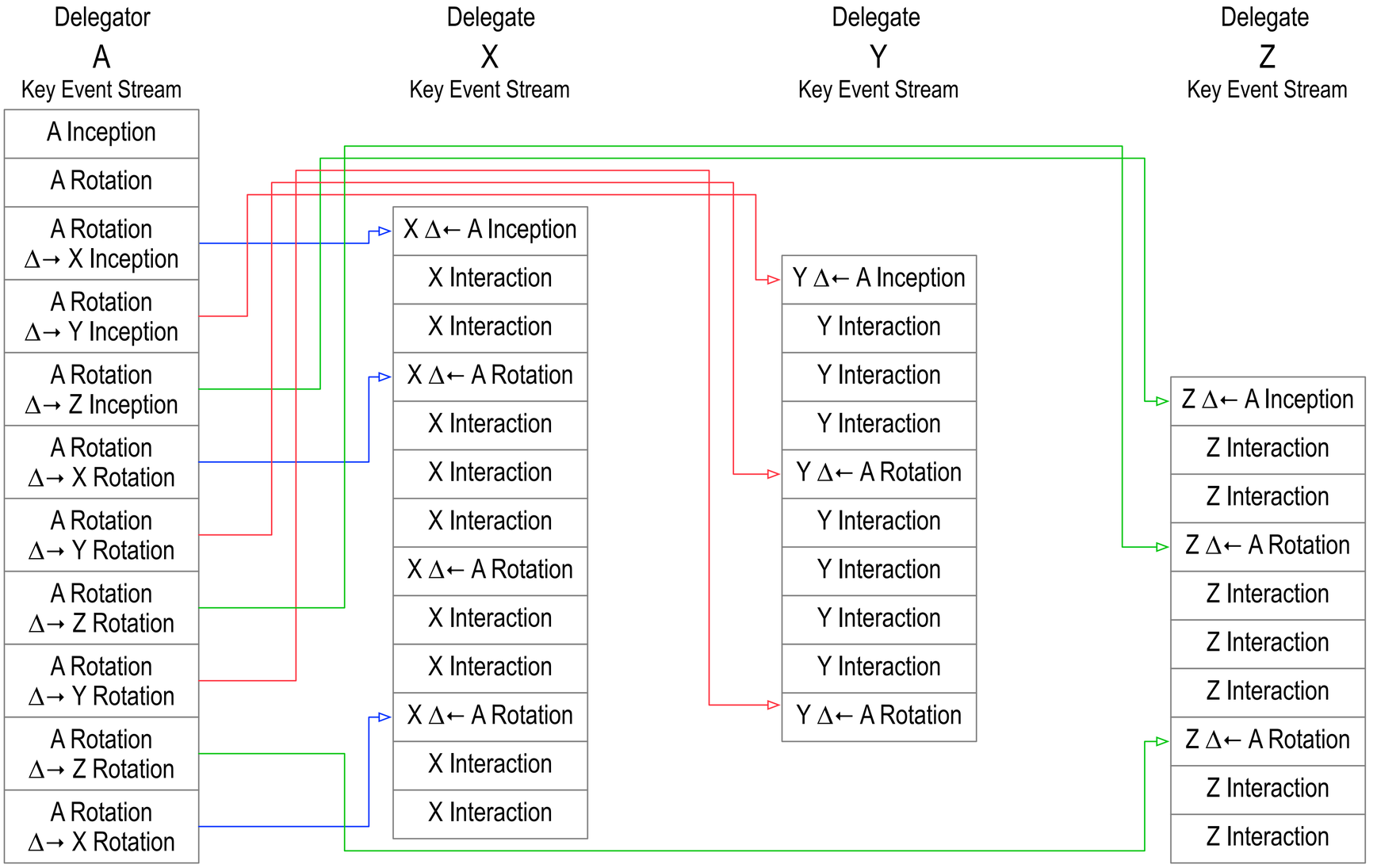
$$\widehat{s}_l = \left[s_0, \ldots, s_{S_l-1} \right]_l$$

$$\widehat{\sigma}_l = \sigma_{C^{e0+\eta-1}} \ldots \sigma_{C^{eE_l-1+\eta-1}} \sigma_{C^{s0+\eta}} \ldots \sigma_{C^{sS_l-1+\eta}}$$

Interaction Event

$$\varepsilon_k = C_{\varepsilon_k} = \left\langle C, t_k, \eta\left(\varepsilon_{k-1}\right), \texttt{itc}, \{data\}, \widehat{s}_k \right\rangle \widehat{\sigma}_k$$

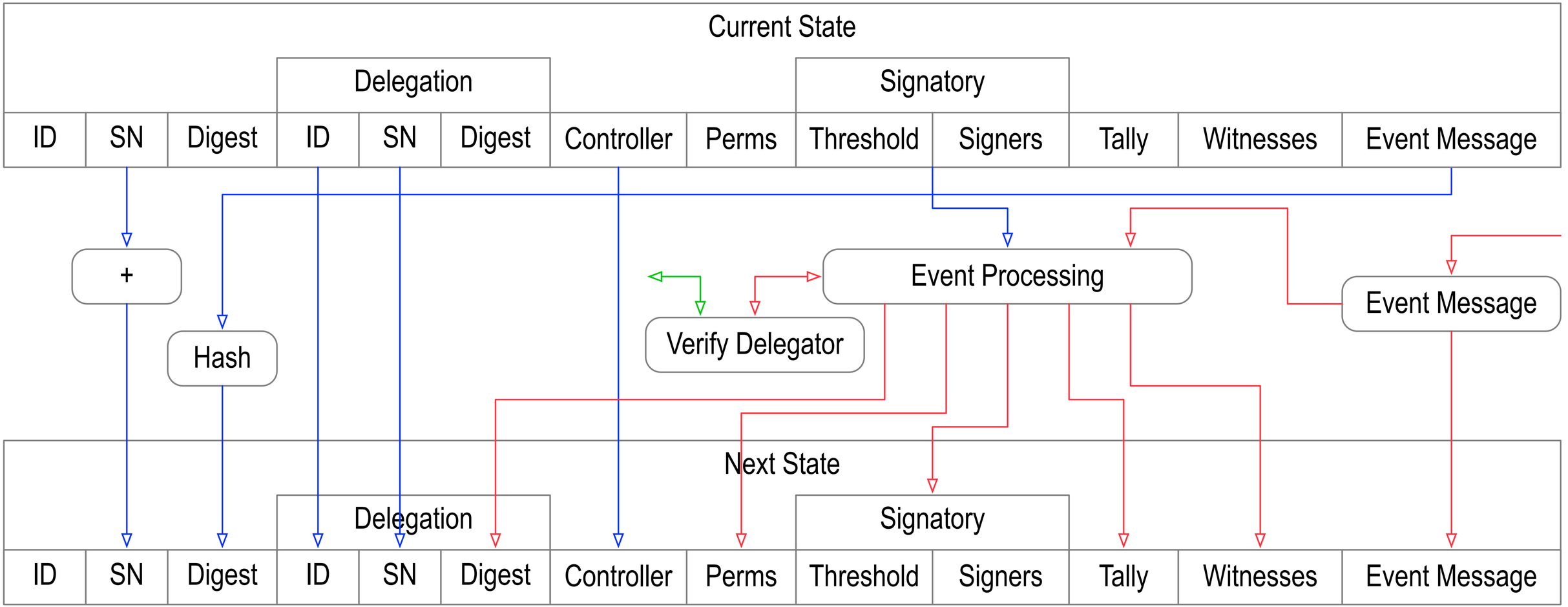
Delegation



Δ→ X : Delegation to X

Δ← A : Delegation from A

Delegated State Verifier Engine



Delegating Inception

$$\varepsilon_k = \left\langle C, t_k, \eta\left(\varepsilon_{k-1}\right), \texttt{rot}, K_{l+1}, \widehat{C}_{l+1}, M_l, \widehat{X}_l, \widehat{Y}_l, \widehat{\Delta}_0^D, \widehat{e}_l, \widehat{s}_l \right\rangle \widehat{\sigma}_l$$

$$\widehat{\Delta}_0^D = \left\{ D, \texttt{dip}, D^0, perms, K_0^D, \widehat{D}_0^D, M_0^D, \widehat{W}_0^D \right\}$$

$$\varepsilon_0 = D_{\varepsilon_0} = \left\langle D, t_0, \texttt{dip}, \widehat{\Delta}_k^C, D^0, perms, K_0^D, \widehat{D}_0^D, M_0^D, \widehat{W}_0^D \right\rangle \sigma_{D^0}$$

$$\widehat{\Delta}_k^C = \left\{ C, t_k^C, \eta\left(\varepsilon_k^C\right) \right\}$$

Delegating Rotation

$$\varepsilon_k = \left\langle C, t_k, \eta\left(\varepsilon_{k-1}\right), \texttt{rot}, K_{l+1}, \widehat{C}_{l+1}, M_l, \widehat{X}_l, \widehat{Y}_l, \widehat{\Delta}_e^D, \widehat{e}_l, \widehat{s}_l \right\rangle \widehat{\sigma}_l$$

$$\widehat{\Delta}_e^D = \left\{ D, \texttt{drt}, perms, K_e^D, \widehat{D}_e^D, M_e^D, \widehat{X}_e^D, \widehat{Y}_e^D \right\}$$

$$\varepsilon_d = D_{\varepsilon_d} = \left\langle D, t_d, \eta\left(\varepsilon_{d-1}\right), \texttt{drt}, \widehat{\Delta}_k^C, perms, K_e^D, \widehat{D}_e^D, M_e^D, \widehat{X}_e^D, \widehat{Y}_e^D \right\rangle \sigma_{D^0}$$

$$\widehat{\Delta}_k^C = \left\{ C, t_k^C, \eta\left(\varepsilon_k^C\right) \right\}$$

Delegated Interaction

$$\varepsilon_d = D_{\varepsilon_d} = \left\langle D, t_d, \eta\left(\varepsilon_{d-1}\right), \texttt{itc}, data, \widehat{s}_d^{\,D} \right\rangle \widehat{\sigma}_d$$

BACKGROUND

Variants

Single Key Sequence for Rotation and Signing

Single Interleaved Key Sequence for Double Rotation and Signing

Single Key Sequence for Rotation with multiple Delegated Key Sequence(s) for Signing

Dual Key Sequence for Separate Rotation and Signing