Preparing for Post-Quantum: The DNSSEC Case

Burt Kaliski, Verisign

6th International Symposium on Cyber Security, Cryptology and Machine Learning (CSCML 2022)

June 30 – July 1, 2022

VERISIGN®

Agenda: Preparing for Post-Quantum DNSSEC

1. DNS and DNSSEC
2. Post-Quantum DNSSEC
3. Merkle Tree Public Keys
4. Merkle Tree Ladders
5. PQ DNSSEC Next Steps

1. DNS and DNSSEC

DNS and DNSSEC: Key Message

DNS is core protocol for **internet naming**

DNSSEC is extension for **authenticating DNS records**

The Domain Name System

341.7 Million Domain Name Registrations[1]

example.com, cscml.org, bgu.ac.il, etc.

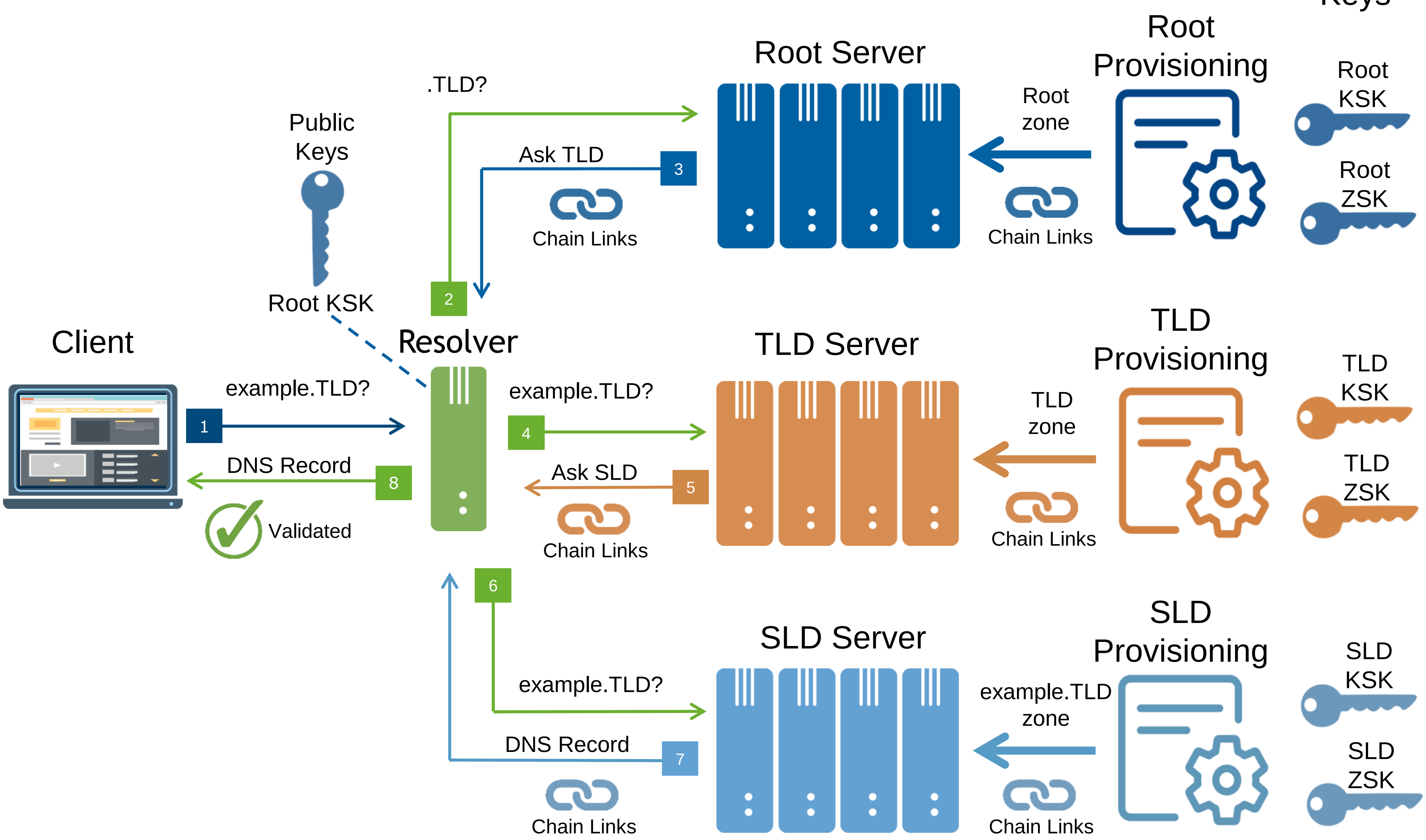
1591 Top-Level Domains[2]

.com, .org, .il, etc.

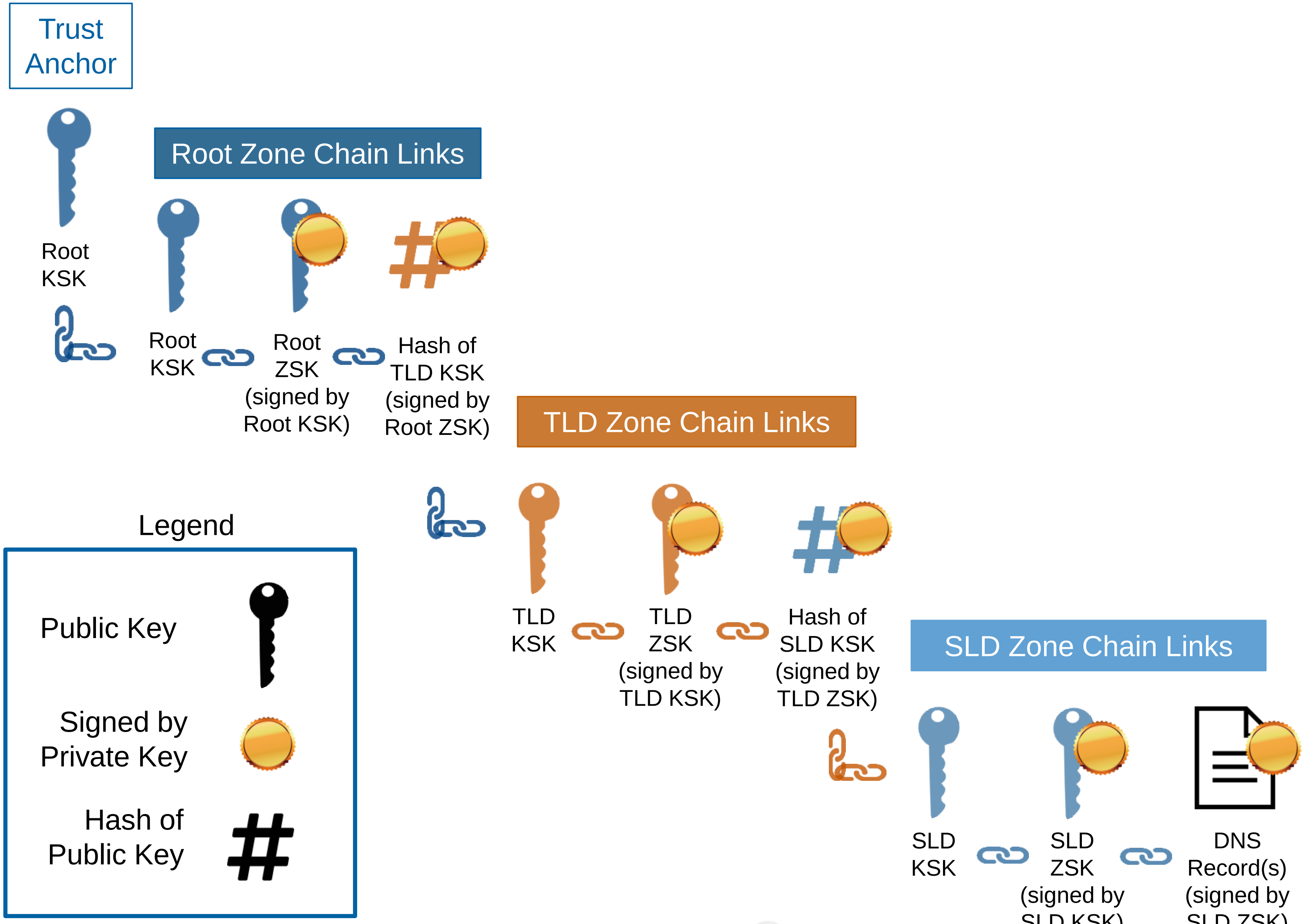
1 Global Root

[1] Verisign, The Domain Name Industry Brief, April 2022. [2] IANA, Root Zone Database, accessed May 19, 2022.

DNSSEC: Ecosystem View (Simplified)



DNSSEC Trust Chain (Simplified)



2. Post-Quantum DNSSEC

Post-Quantum DNSSEC: Key Message

DNSSEC use case has **unique priorities** for **practical long-term cryptographic resiliency**

Some DNSSEC Distinctives

Practical Considerations Differ from Other Use Cases

Small response sizes (e.g., $\leq$ 1,220 bytes) preferred for UDP transport

Sign-once, verify-many model

Ceremonial / offline signing

Public key lookups built into protocol

Highly decentralized deployment

Primary Classical DNSSEC Algorithms & Sizes

Mandatory or Recommended for Signing Implementations[1]

Algorithm	Public Key Size (bytes)[2]	Signature Size (bytes)[2]	Notes
RSASHA256[3,4]	260	256	Mandatory
ECDSAP256SHA256[5]	32	64	Mandatory
ED25519[6]	32	64	Recommended

All Are Vulnerable to Quantum Cryptanalysis

[1] RFC 8624. [2] Algorithm-specific portion, excludes protocol overhead. [3] RFC 5702. [4] Assumes 2048-bit keys, public exponent *e* = $2^{16}+1$. [5] RFC 6605. [6] RFC 8080.

Leading NIST PQC Project Signature Algorithms[1]

Algorithm	Public Key Size (bytes)[2]	Signature Size (bytes)[2]	Notes
Falcon[3]	897	666	Lattice-based NIST Level I
Dilithium[4]	1,312	2,240	Lattice-based NIST Level II
SPHINCS+[5]	32	7,856	Alternate Stateless hash-based NIST Level I

[1] D. Moody, The Beginning of the End: The First NIST PQC Standards, PKC 2022, March 2022.
[2] Algorithm-specific portion, excludes protocol overhead. [3] T. Prest et al., Falcon. [4] V. Lyubashevsky et al., CRYSTALS – Dilithium. [5] A. Hülsing et al., SPHINCS+. Refs. 3-5 all from NIST 3rd PQC Standardization Conference, June 2021.

Stateful Hash-Based Signature Algorithm Sizes[1]

Algorithm	Public Key Size (bytes)[2]	Signature Size (bytes)[2]	Notes
HSS-LMS with params L=2, LMS_SHA256_M32_H10, LMOTS_SHA256_N32_W8[3]	60	2,836	Max. 2^{20} signatures
XMSSMT-SHA2_20/2_256[4]	68	4,963	Max. 2^{20} signatures

[1] A. Fregly and R. van Rijswijk-Deij, Stateful Hash-Based Signatures for DNSSEC, Internet-Draft, 2022. [2] Algorithm-specific portion, excludes protocol overhead. [3] RFC 8554. [4] RFC 8391.

Key Priority: Diversity of Cryptographic Families

<u>Solution Goal:</u> Deploy Post-Quantum Techniques That Fit DNSSEC from Two or More Families

Pre-Quantum

- Integer Factoring
- Finite Field Discrete Logarithms
- Elliptic Curve Discrete Logarithms

→

Post-Quantum?

- Lattice
- Hash-Based
- Multivariate
- Zero-Knowledge Proofs

Long-Term Resiliency: If One Technique Becomes at Risk, Switch to Alternate until Replacement Can Be Deployed

3. Merkle Tree Public Keys

(aka Synthesized Signing Keys)

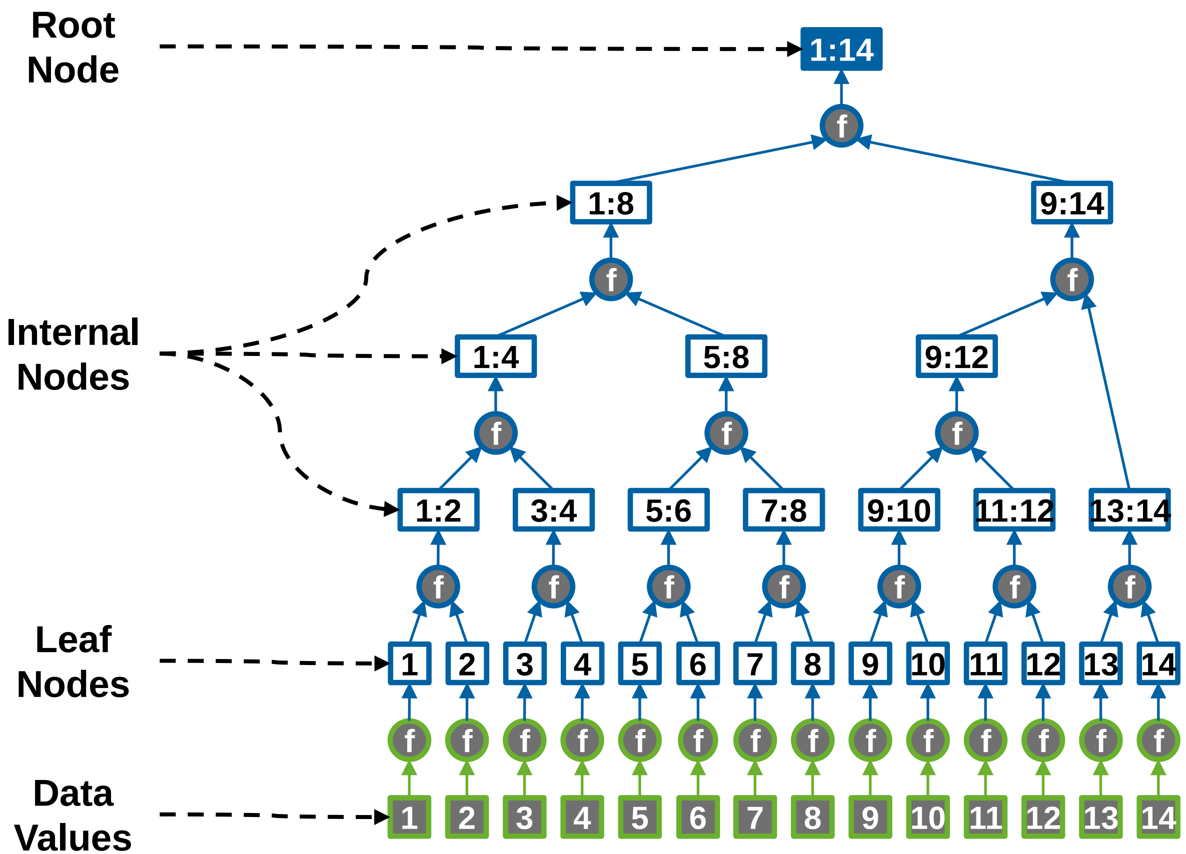
Merkle Tree Public Keys:[1] Key Message

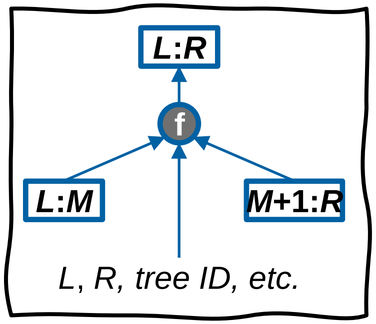
Merkle Tree Public Keys can help provide **long-term cryptographic resiliency** for DNSSEC with **relatively short signatures**

[1] B. Kaliski, Securing the DNS in a Post-Quantum World: Hash-Based Signatures and Synthesized Zone Signing Keys, Verisign blog, Jan. 2021.

Merkle Tree

Root Node Recursively Authenticates All Data Values

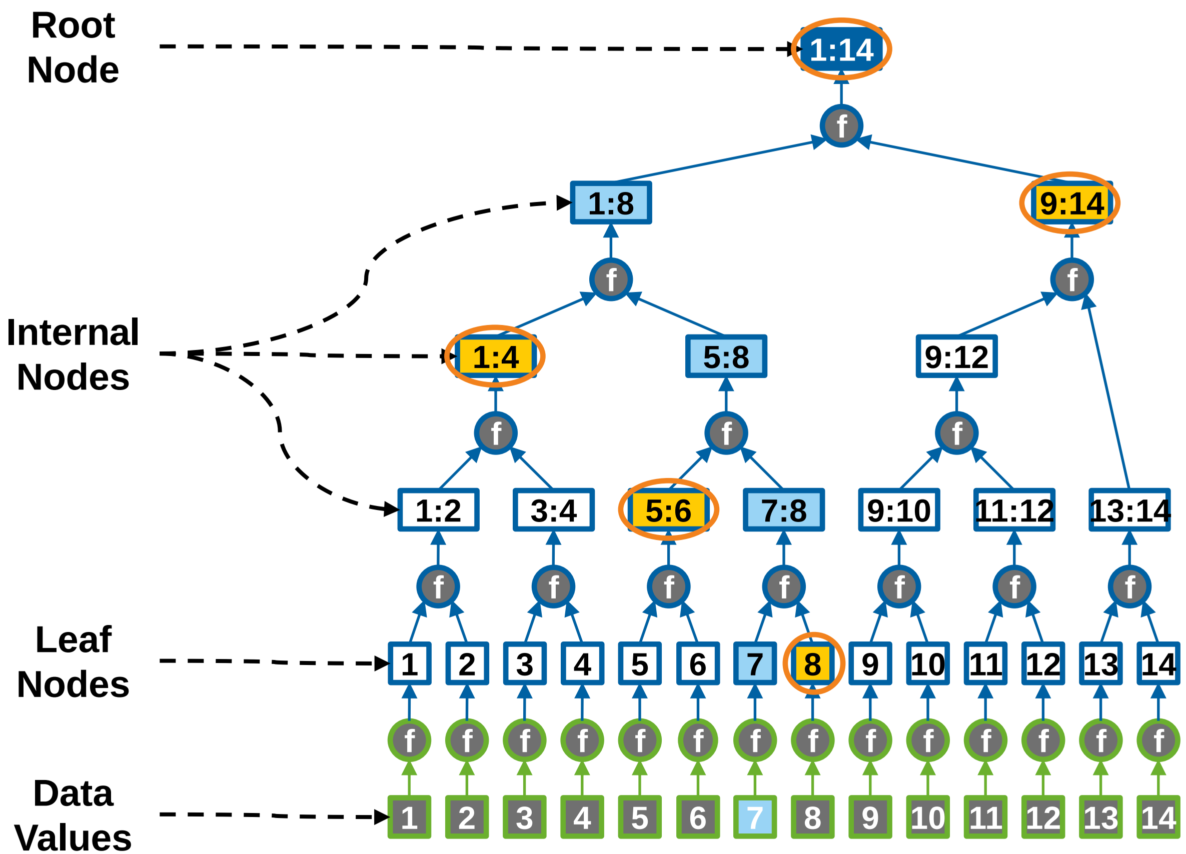




- **Parent node value is hash of child node values, "context" info**
- **"Canonical aggregation" used for example trees**

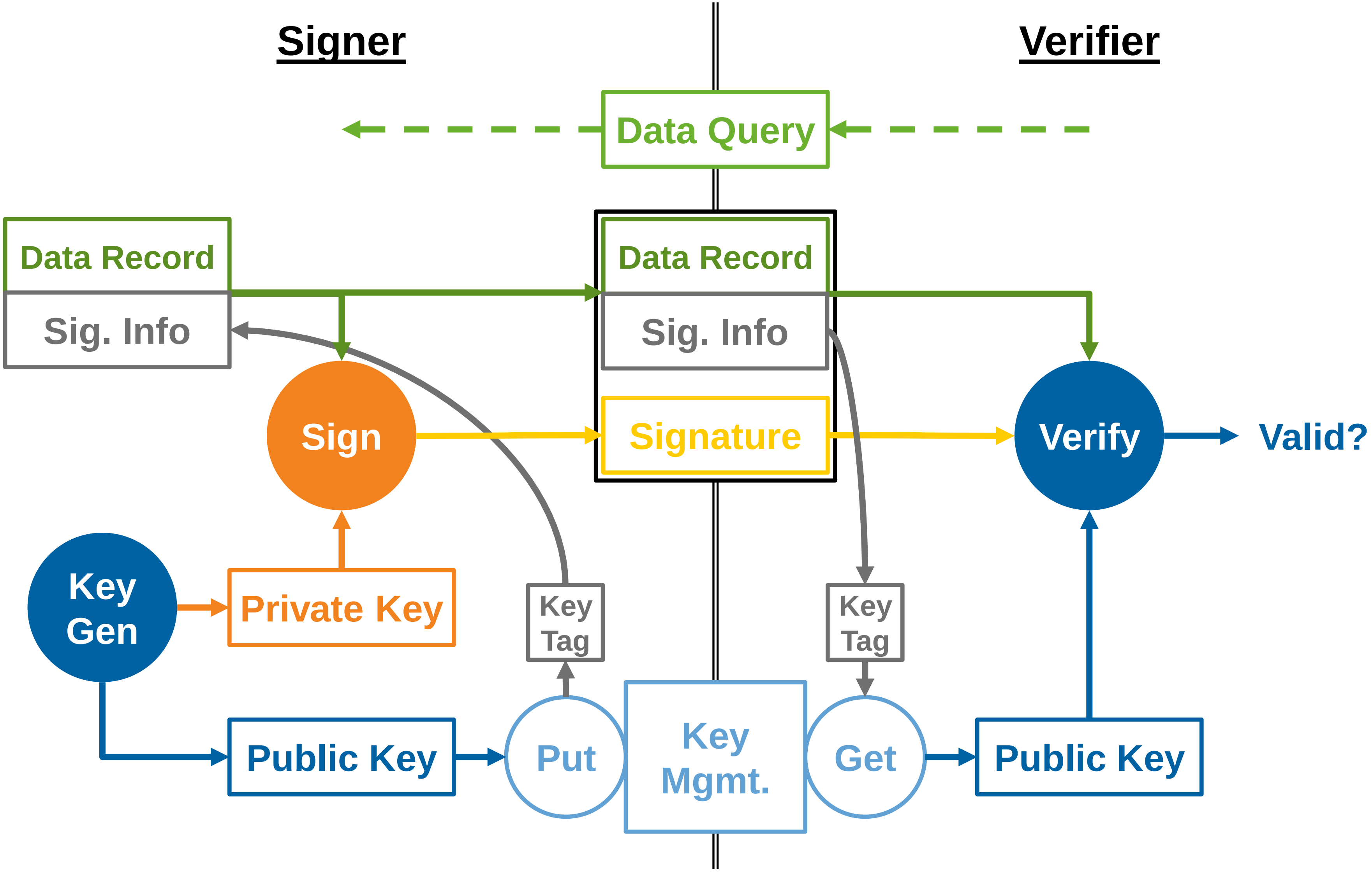
Authentication Path

Verify Data Value by Re-Hashing with Sibling Nodes



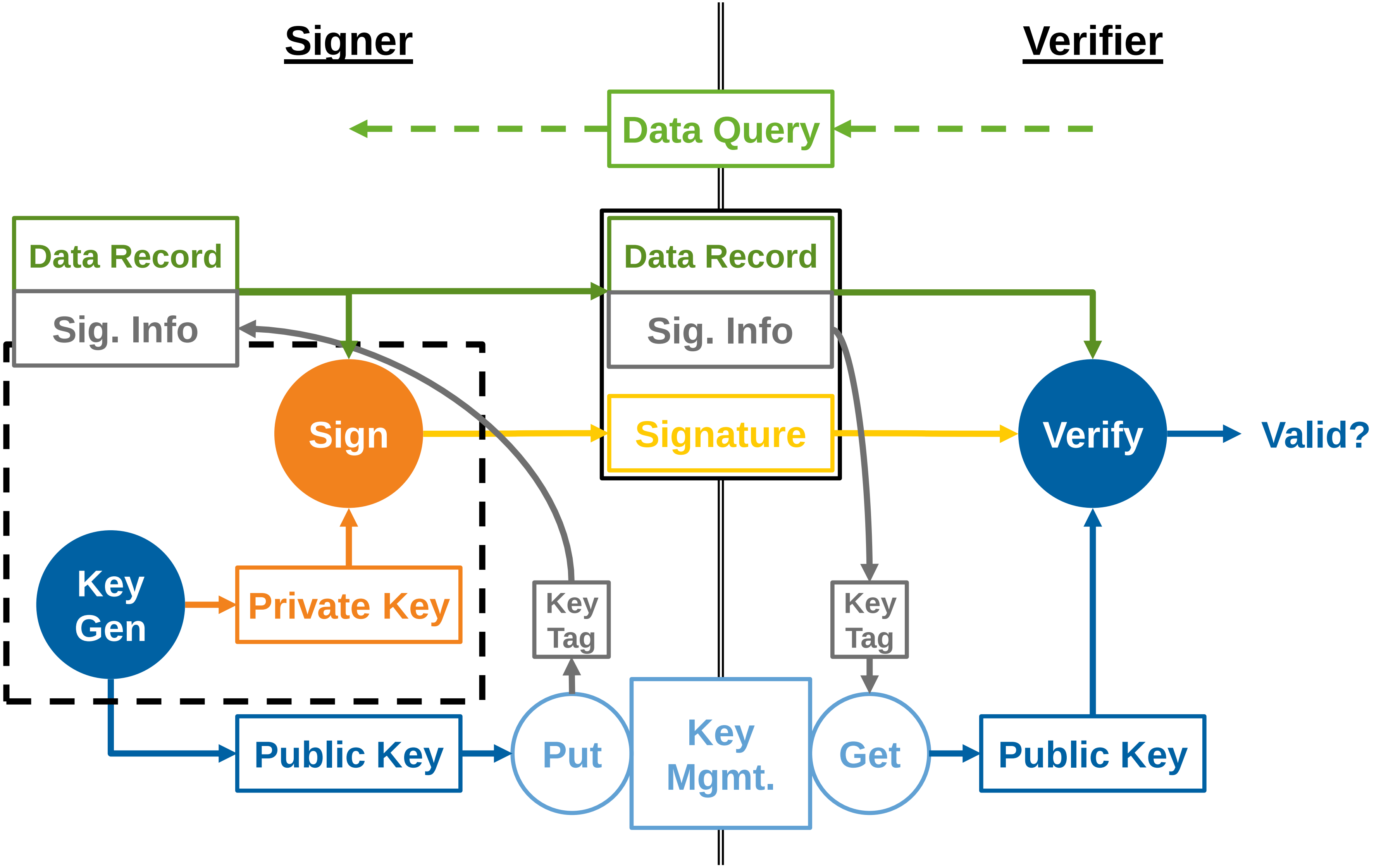
Sibling nodes = Auth. Path

DNSSEC Data Authentication Model



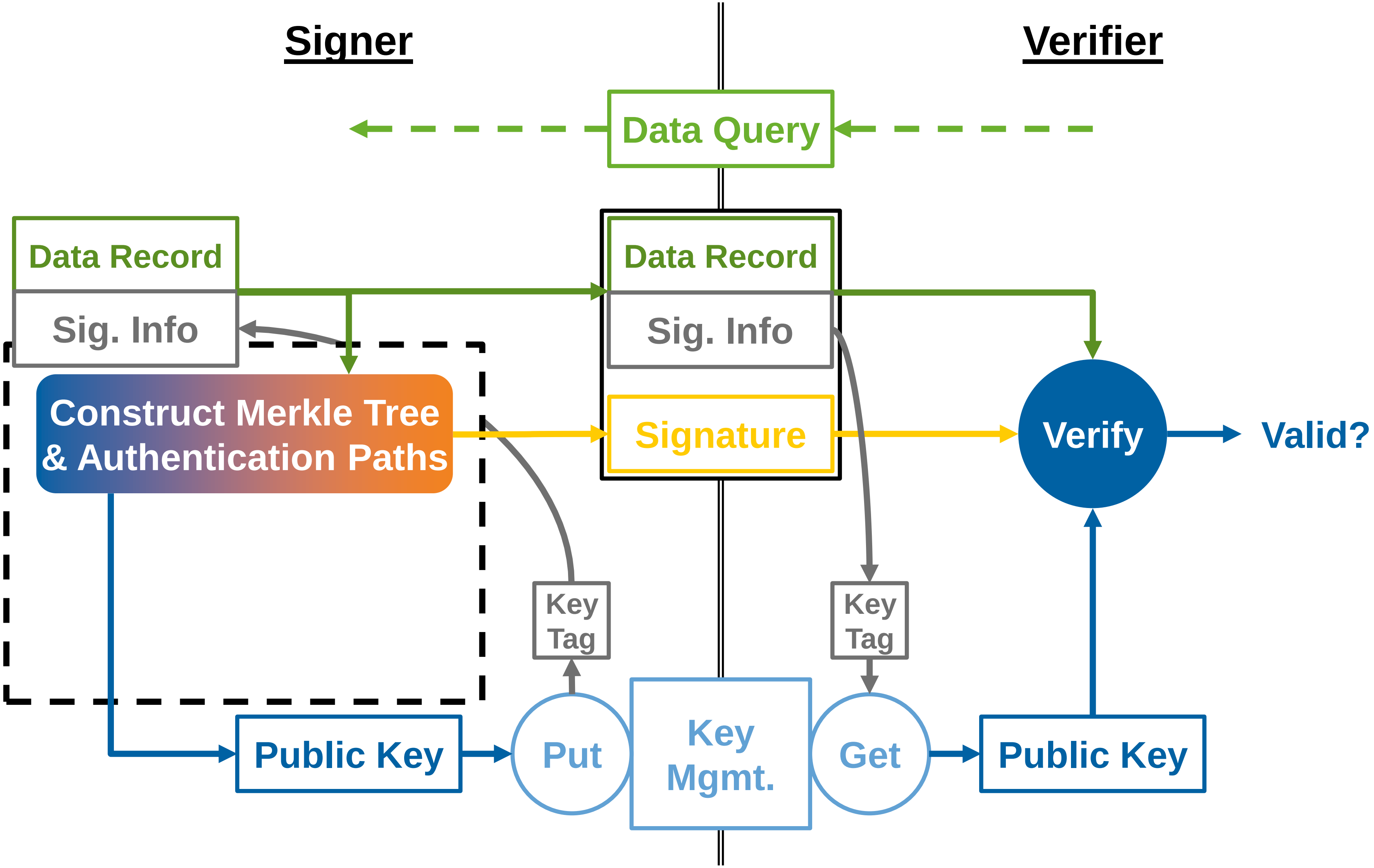
DNSSEC Data Authentication Model

Verifier's View: Signer Produces Public Key & Signature



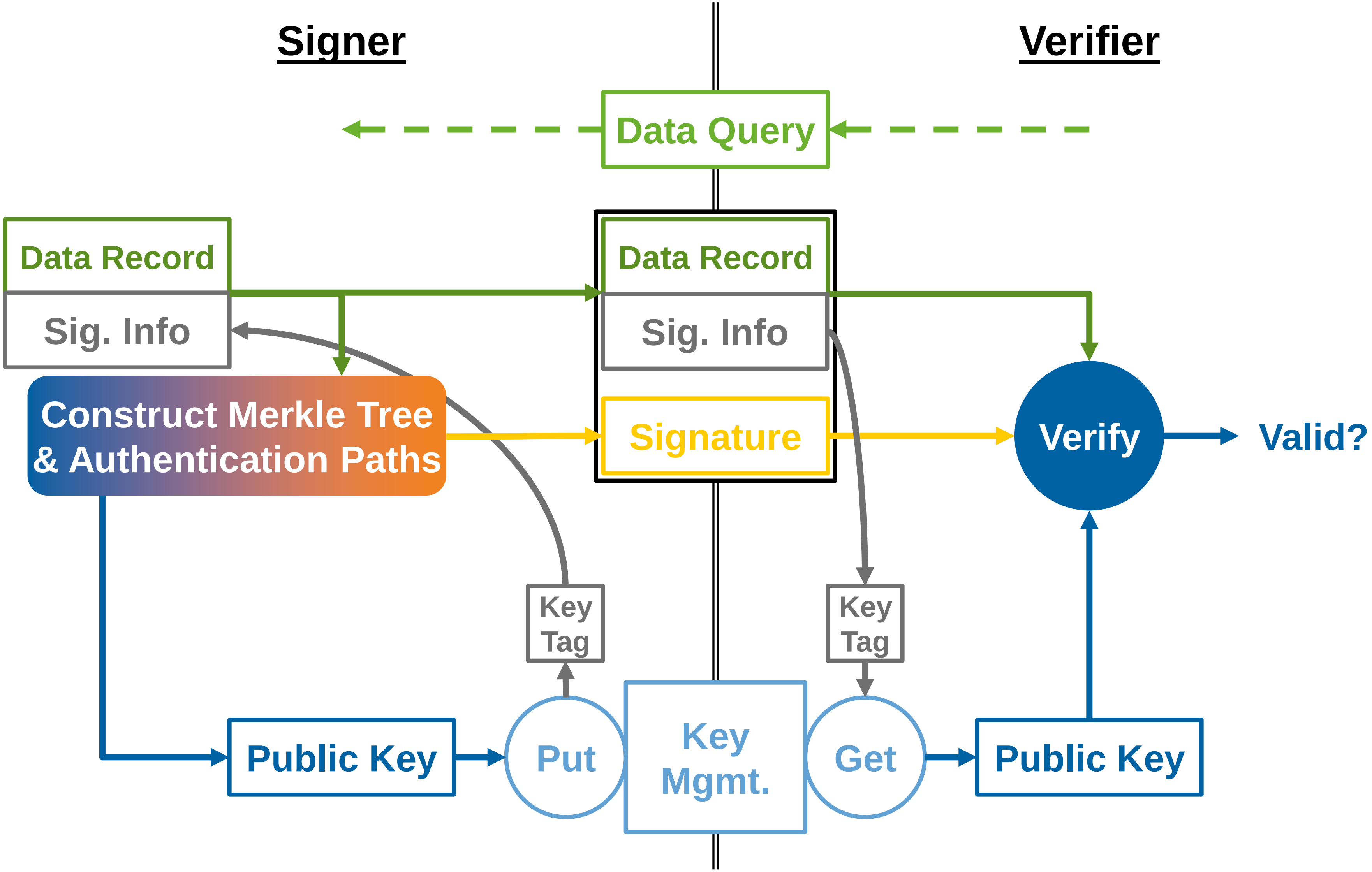
DNSSEC with Merkle Tree Public Keys

Another Way to Produce Public Key & Signature



DNSSEC with Merkle Tree Public Keys

Public Key = Tree Root; Signature = Authentication Path



Paradigm Shift: Generated to Synthesized

Conventional DNSSEC	Merkle Tree Public Keys
Generated Key Pair	Synthesized Public Key
Key Gen + Sign	Construct Merkle Tree & Authentication Paths
Public Key	Tree Root (or any node)
Private Key	n/a
Signature	Authentication Path
Verify	Verify Authentication Path
1-2 Active Public Keys	Many Active Public Keys*

**Public Keys Change As Data Values Are Updated*

Merkle Tree Public Key Signature Scheme Sizes

Draft Specification in Preparation

Algorithm	Public Key Size (bytes)[1]	Signature Size (bytes)[1]	Notes
MTPKSS-SHA2_20/256	72	4 to 644	Max. 2^{20} data values. Signature size increases as data values are appended

<u>Draft Formats</u>

Public Key = $[\text{Tree ID}]_{32}$. $[\text{Left Index}]_{4}$. $[\text{Right Index}]_{4}$. $[\text{Node Value}]_{32}$

Signature = $[\text{Leaf Index}]_{4}$. (0-20) x $[\text{Sibling Value}]_{32}$

[1] Algorithm-specific portion, excludes protocol overhead

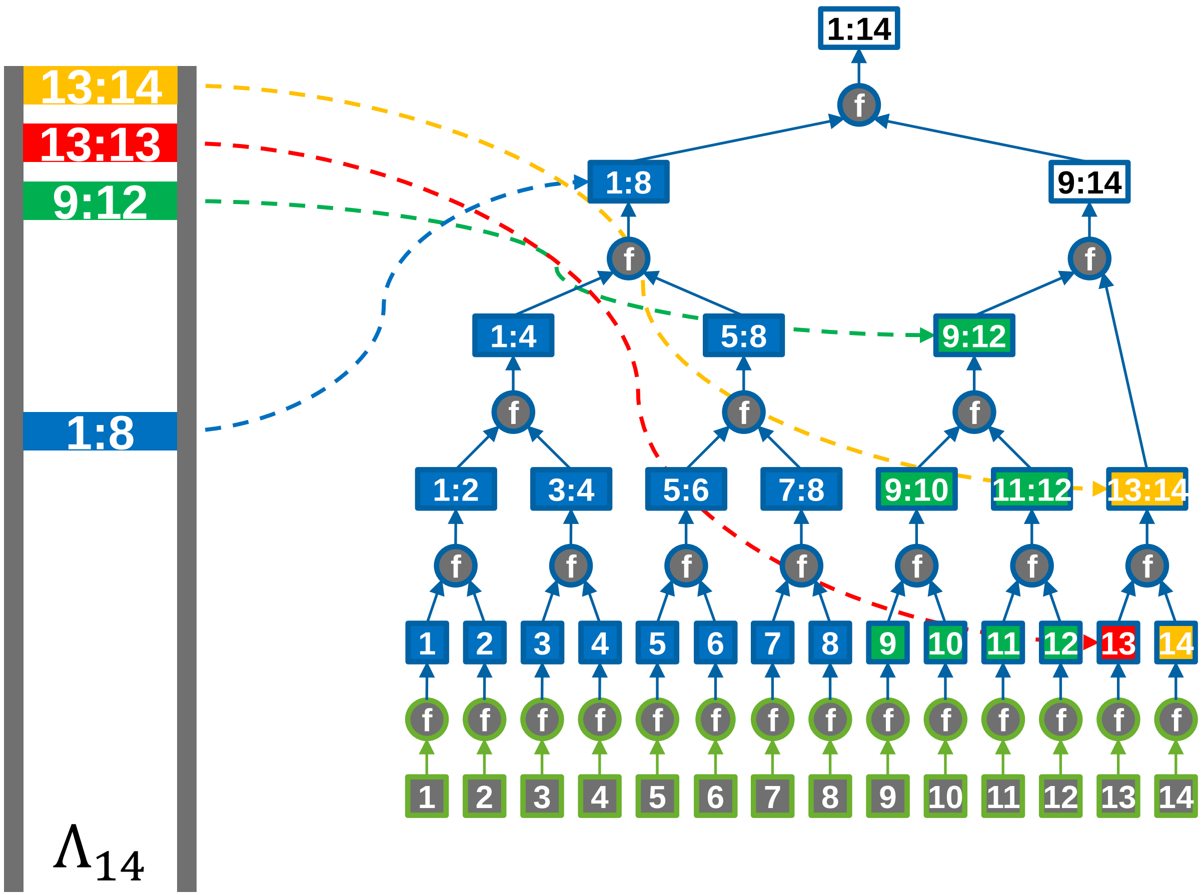
4. Merkle Tree Ladders

Merkle Tree Ladders: Key Message

Merkle Tree Ladders are a way to **model, optimize key management** for Merkle Tree Public Keys

Merkle Tree Ladder

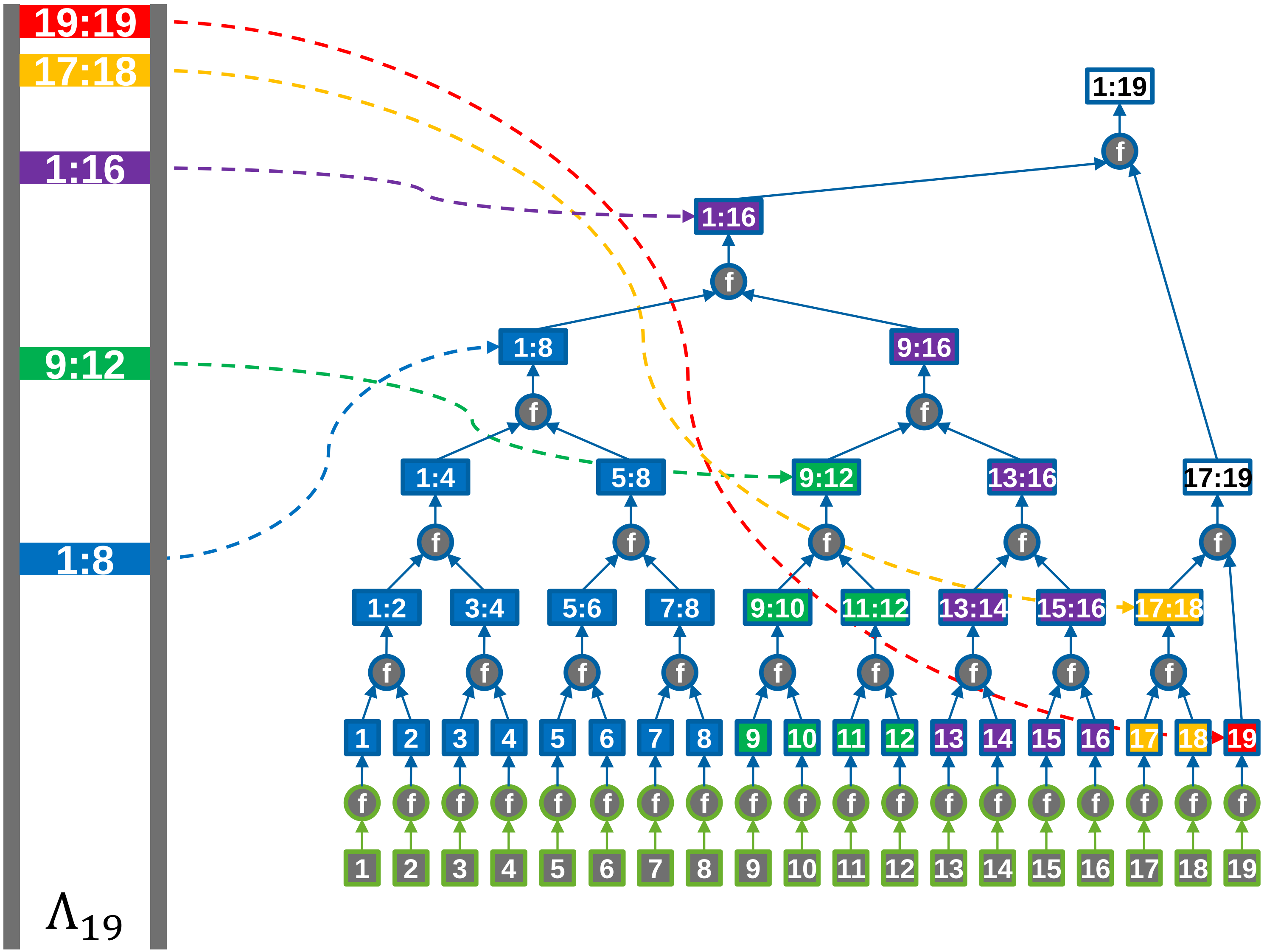
Rungs Collectively Authenticate All Data Values



- **Any node in Merkle tree can be a rung on ladder**
- **Generalization: Any node in Merkle <u>graph</u>**

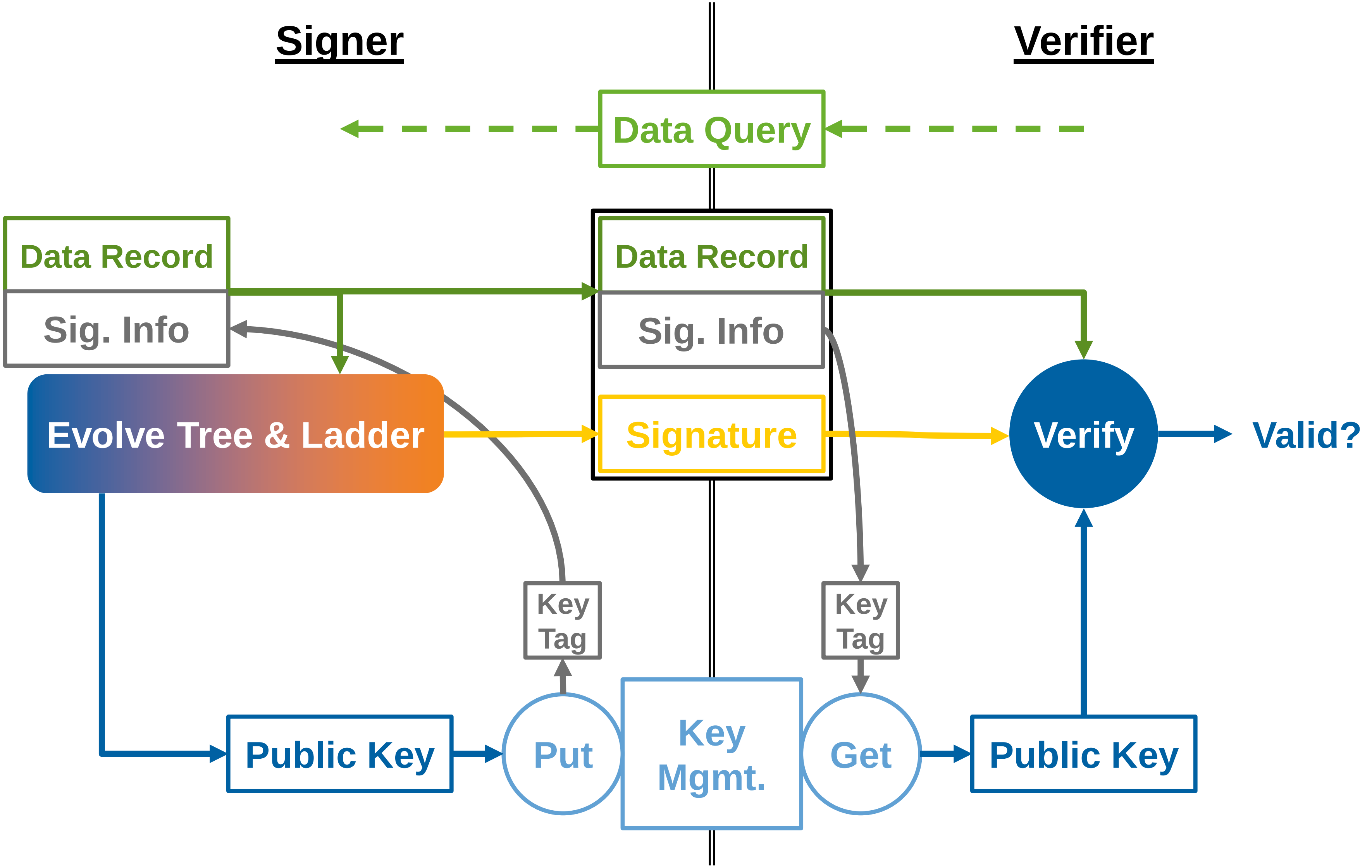
Ladder Evolution

Rungs Updated to Reflect New Data Values



DNSSEC with Merkle Tree Ladders

Public Key = Ladder Rung; Signature = Auth. Path to Rung



Definition of Endurance

Endurance (Λ_N) = maximum E such that:

Prob[E successive responses can be verified using rungs from Λ_N] $\geq \frac{1}{2}$

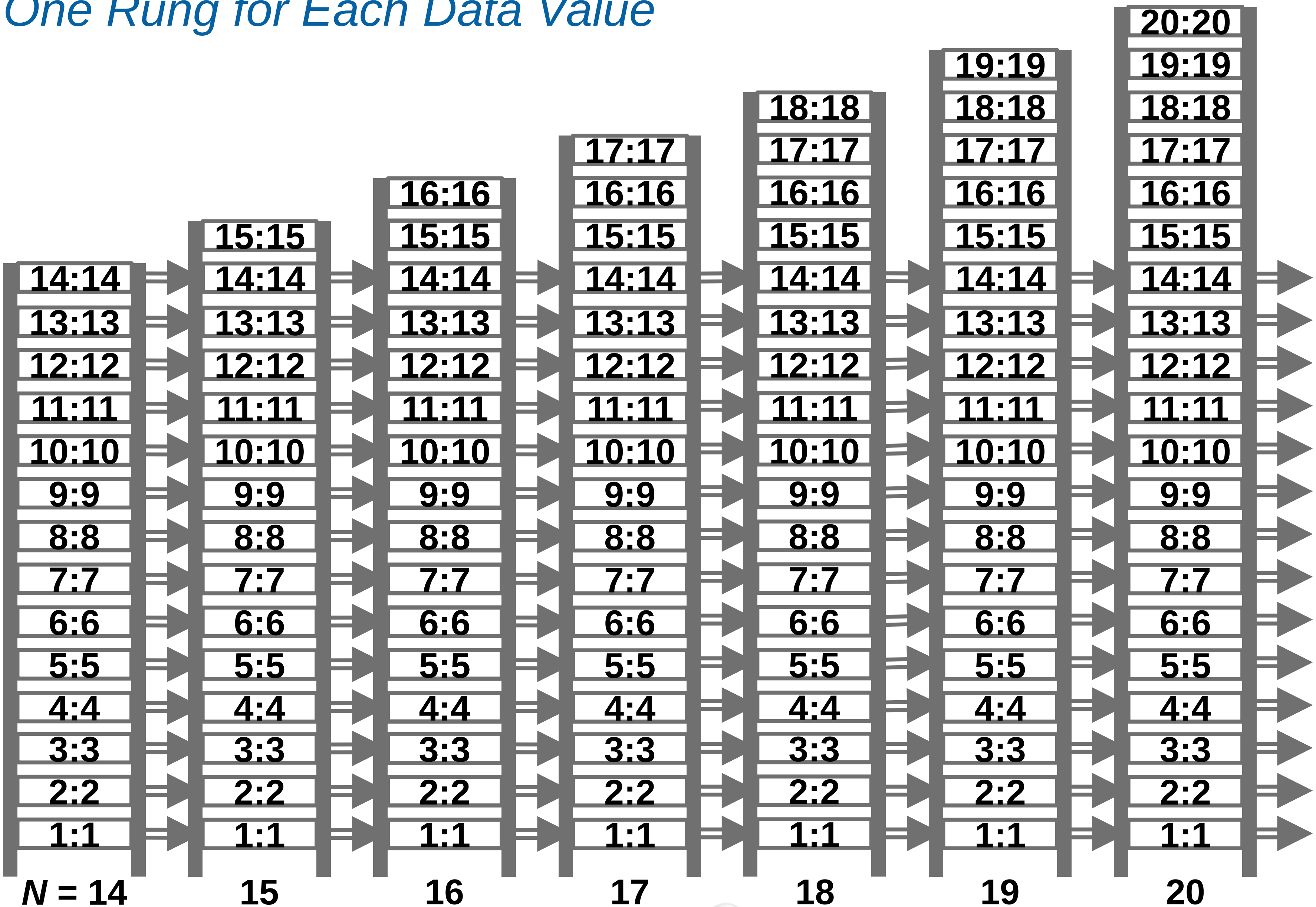
- **Endurance depends on rung "strategy"**
- **May also depend on *N*, signer's update pattern, verifier's query pattern and response indexes**

Initial Model: 1 Append & 1 Query / Iteration

- 1 new data value appended per iteration
- 1 query per iteration, for a random data value
- Response = data value with random index

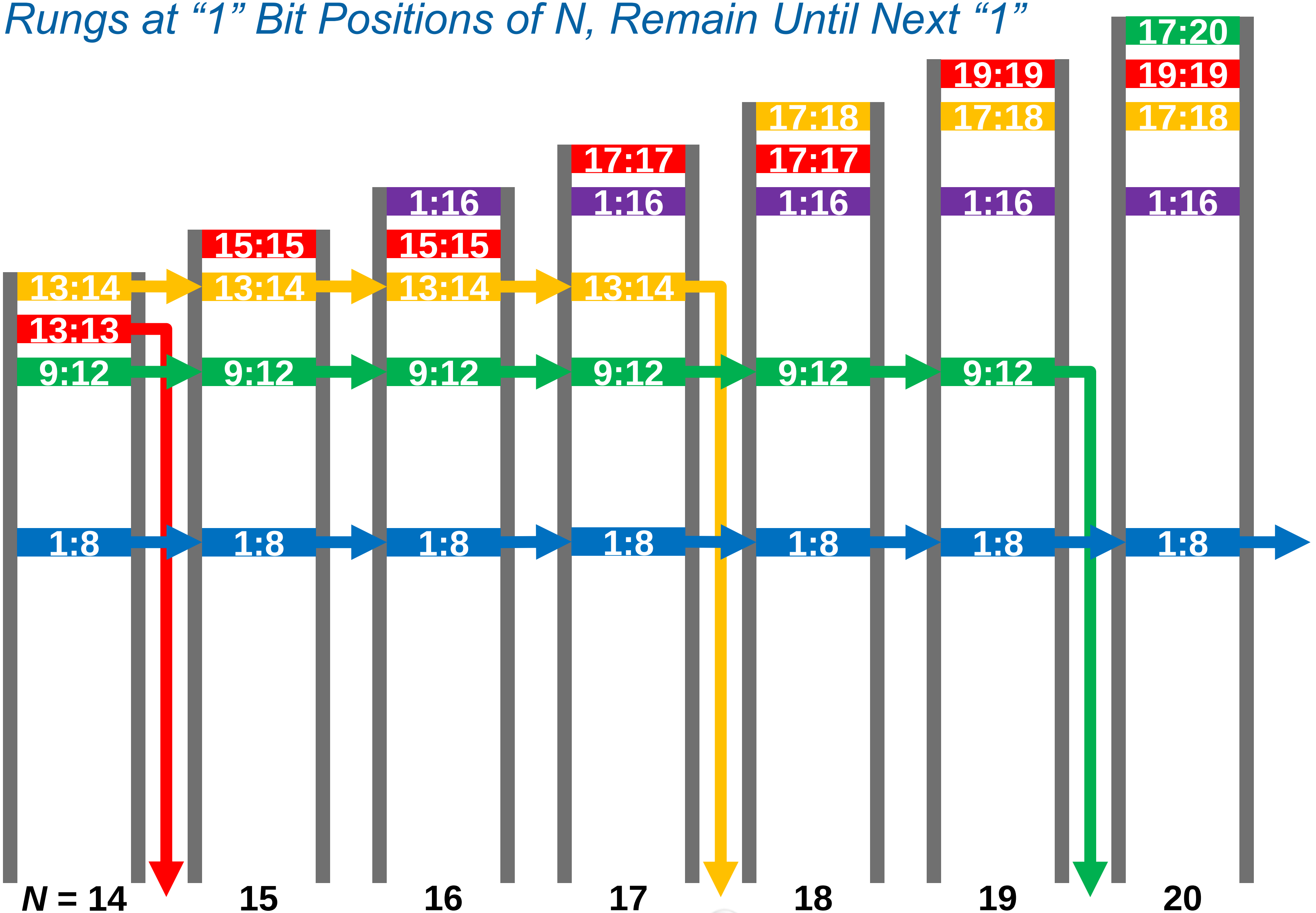
Baseline: Full-Rung Strategy

One Rung for Each Data Value



Improvement: Extended Binary-Rung Strategy

Rungs at "1" Bit Positions of N, Remain Until Next "1"



Comparing Strategies (under initial model)

Strategy	Number of Rungs	Endurance (Queries)
Full-Rung	N	$\sim\sqrt{2\ln 2}\,\sqrt{N}$
Extended Binary-Rung	$\sim\log_2 N$	$\sim\sqrt{\frac{2}{3}\ln 2}\,\sqrt{N}$ to $\sim\sqrt{2\ln 2}\,\sqrt{N}$

Analysis similar to Birthday Paradox

Revised Model: α Appends, ρ Queries / Iteration

- α new data values appended per iteration
- ρ queries per iteration, each for a random data value
- Response = data value with random index

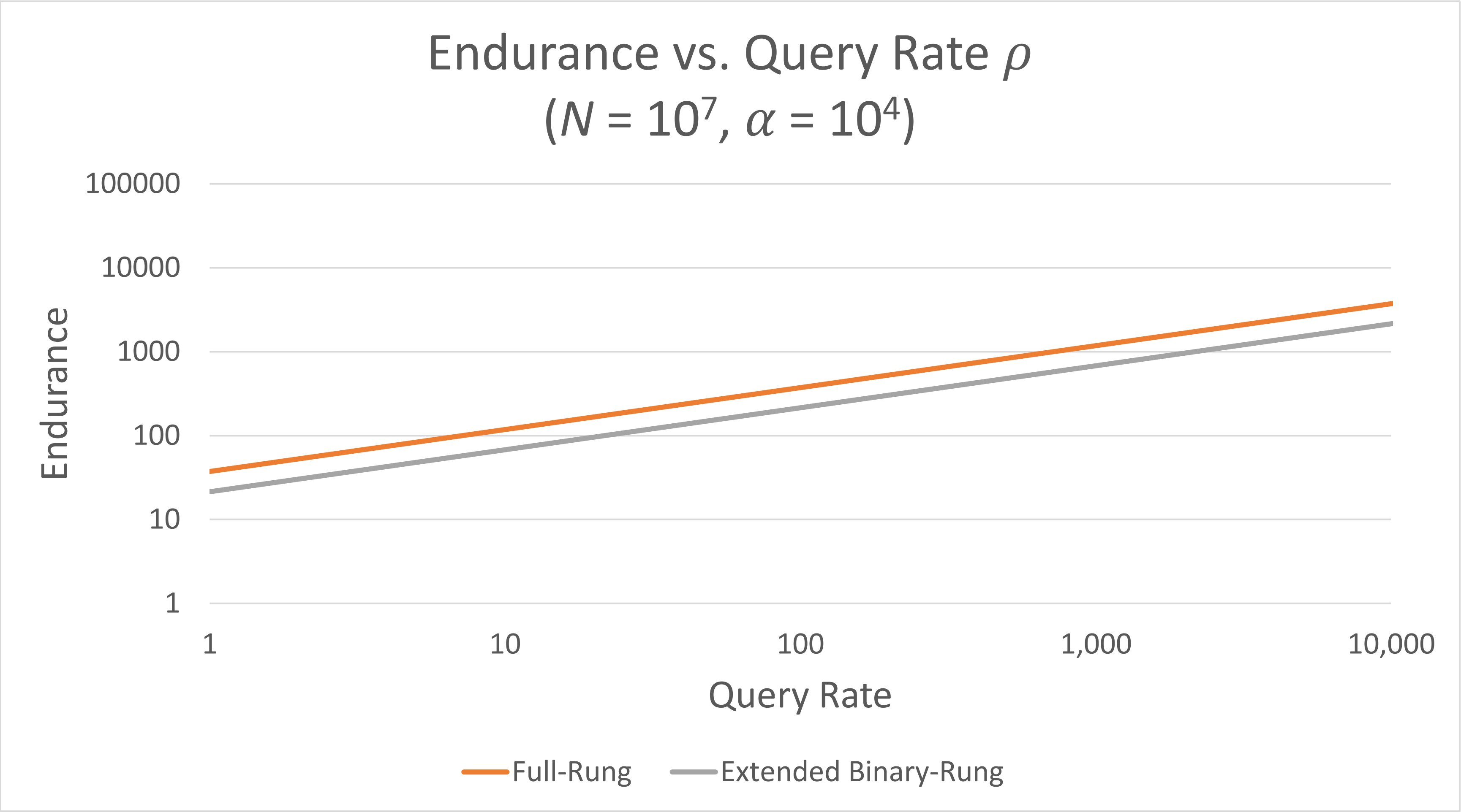
Comparing Strategies (under revised model)

Strategy	Number of Rungs	Endurance (Queries)
Full-Rung	N	$\sim\sqrt{2\ln 2}\sqrt{\frac{\rho}{\alpha}}\sqrt{N}$
Extended Binary-Rung	$\sim\log_2 N$	$\geq \sim\sqrt{\frac{2}{3}\ln 2}\sqrt{\frac{\rho}{\alpha}}\sqrt{N}$

Many variants and optimizations possible

Endurance Grows as Query Rate Increases

Extended Binary-Rung Almost as Good as Full-Rung



5. PQ DNSSEC Next Steps

PQ DNSSEC Next Steps: Key Message

DNSSEC needs a **dedicated research and standards effort** to ensure long-term cryptographic resiliency

Revisiting Key Messages

DNS is core protocol for **internet naming**; **DNSSEC** is extension for **authenticating records**

DNSSEC use case has **unique priorities** for **practical long-term cryptographic resiliency**

Merkle Tree Public Keys can help provide **long-term resiliency** with relatively **short signatures**

Merkle Tree Ladders are a way to **model and optimize** Merkle Tree Public Keys

DNSSEC needs its own **research and standards effort** for long-term cryptographic resiliency

Recommended Next Steps

- PQ DNSSEC research collaborations
- PQ DNSSEC working / research group in IETF / IRTF
- Internet-Draft on Merkle Tree Public Keys
- Internet-Drafts on complementary approaches
- Prototyping and interoperability testing
- Modeling with real-world workloads
- *(over time)*
- DNSSEC cryptographic resiliency upgrades
- DNS security for next 35+ years

Questions?

Questions?