The *Erik Synchronization Protocol* for use with the RPKI

draft-spaghetti-sidrops-rpki-erik-protocol

Job Snijders, Tim Bruijnzeels, Tom Harrison, Wataru ‘Alt’ Ohgai
job@sobornost.net, tim@ripe.net, tomh@apnic.net, alt@nic.ad.jp

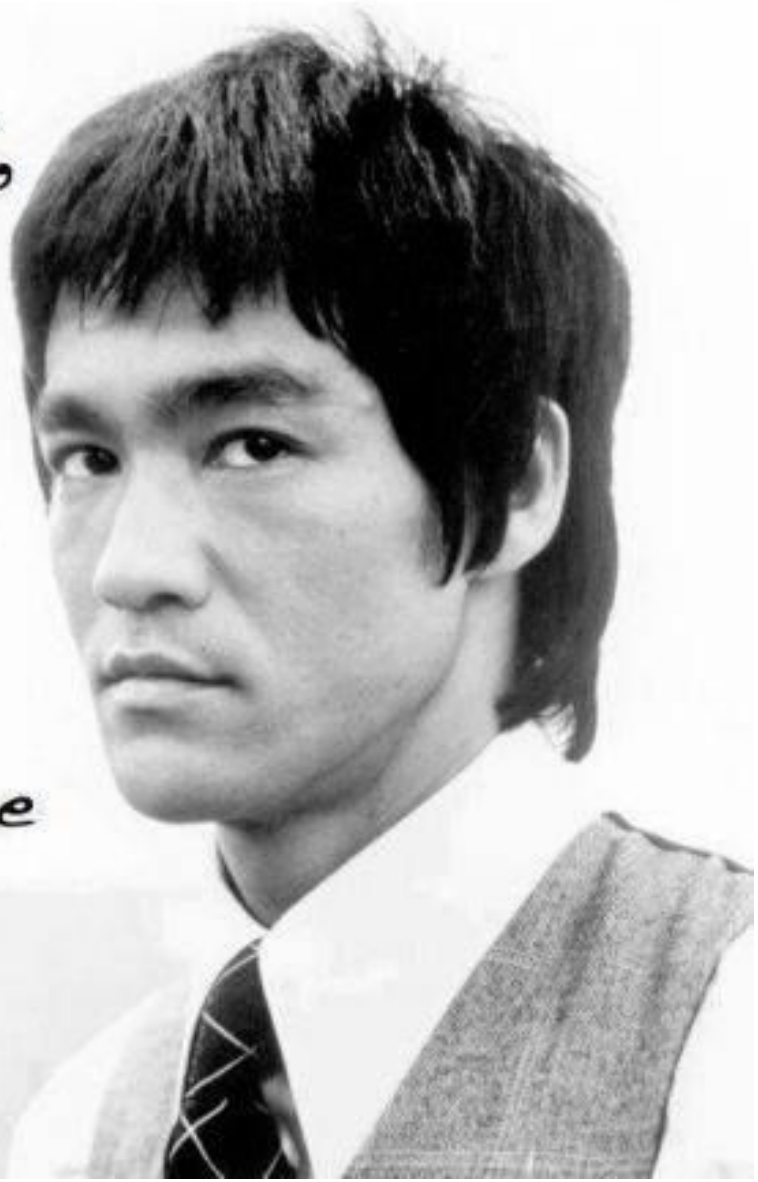
Knowing is not enough,

We must APPLY.

Willing is not enough,

We must DO.

– *Bruce Lee*



Proportions of the RPKI data distribution problem

Count:	today, the RPKI is ~ **456,000** tiny objects (files)
Churn:	~ half those files change at least once a day
Churn rate:	~ 2 new objects issued every second
Total size:	**~** 0.8 gibibytes
Consumers:	**~** 5000 validators
Trend:	*up and to the right*

https://labs.ripe.net/author/job_snijders/rpkis-2024-year-in-review/

So, what's used to distribute RPKI data today?

+

Internet Engineering Task Force (IETF)
Request for Comments: 8182
Category: Standards Track
ISSN: 2070-1721

T. Bruijnzeels
O. Muravskiy
RIPE NCC
B. Weber
Cobenian
R. Austein
Dragon Research Labs
July 2017

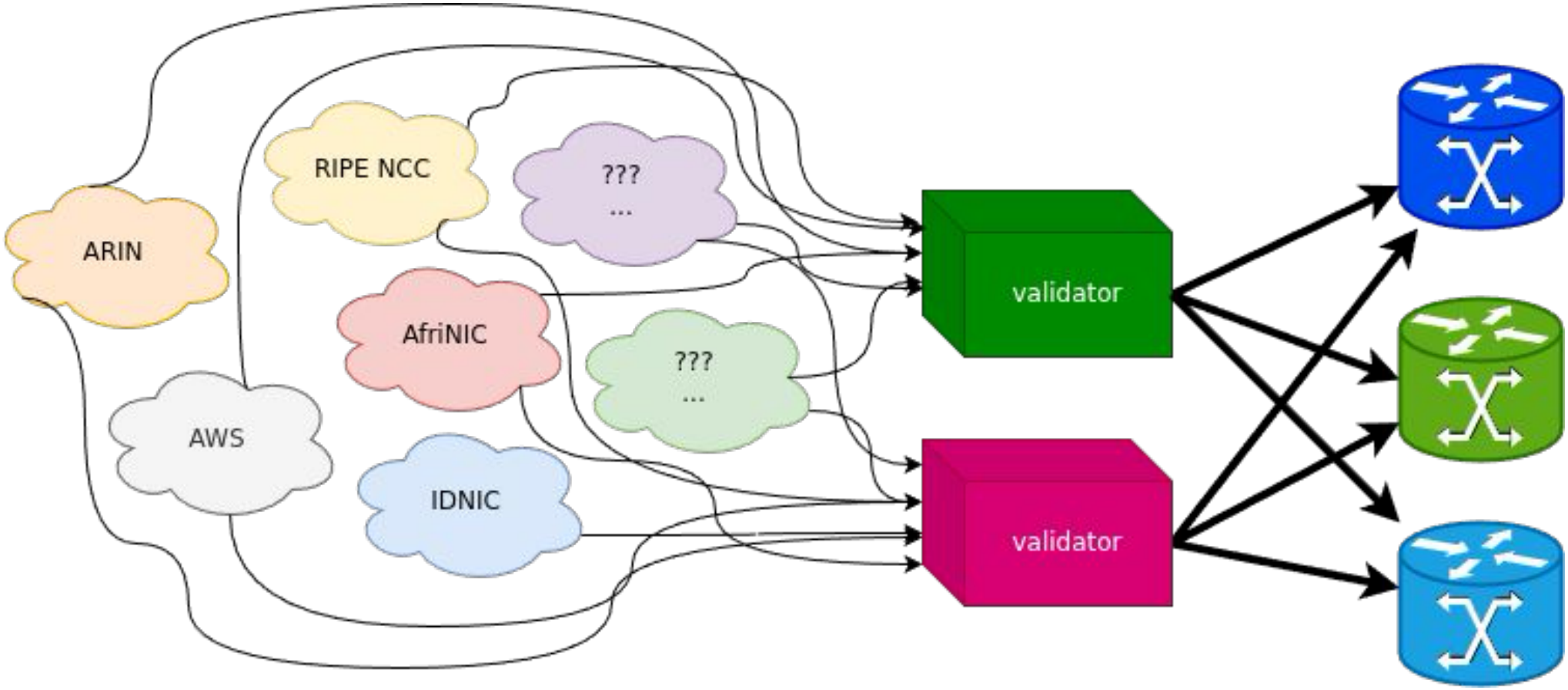
The RPKI Repository Delta Protocol (RRDP)

Abstract

In the Resource Public Key Infrastructure (RPKI), Certificate Authorities (CAs) publish certificates, including end-entity certificates, Certificate Revocation Lists (CRLs), and RPKI signed objects to repositories. Relying Parties retrieve the published

High level overview of RPKI data supply chain

RPKI Repositories → *Rsync / RRDP* → Validators → BGP Routers



But… rsync is efficient, right?!

Yes and no!

Rsync is efficient because it transfers “only the difference”...

But, calculating the difference on-the-fly, also consumes CPU & network resources!

Measuring Rsync

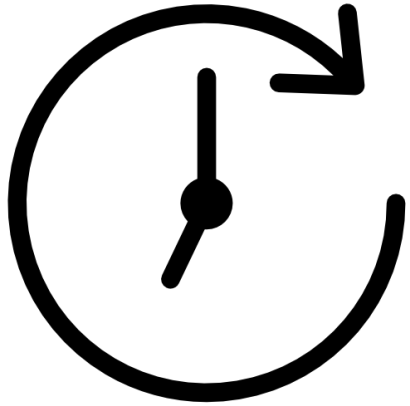
Interval	Bandwidth consumption
Minimum	~ 4 megabytes
Rsync every 15 minutes	~ 40 megabytes
Rsync every 60 minutes	~ 52 megabytes

RRDP in a nutshell

- HTTP-based protocol, static pre-calculated content
- All add/update/delete operations are written into a “journal”
- RPs download the “journal” and replay it

Consequently:

Fetches of data ***already overtaken by events***



Measuring RRDP

Interval	Bandwidth consumption
Minimum	~ 0.5 megabytes
RRDP every 15 minutes	~ 5 megabytes
RRDP every 60 minutes	~ 100 megabytes

Comparing Rsync and RRDP

Interval	Rsync	RRDP
Minimum	~ 4 megabytes	~ 0.5 MB
Every 15 minutes	~ 40 megabytes	~ 5 MB
Every 60 minutes	~ 50 megabytes	~ 100 MB

Rsync / RRDP intrinsic design issues

- Rsync servers vulnerable to easy denial of service (CPU hogging)
- Rsync expensive traversal of resources (no “if-modified-since”)
- RRDP servers can easily DoS their clients (disk space hogging)
- RRDP “loss of state” → reinitialize through snapshot download
- In RRDP FSM, many error paths lead to “snapshot download”

Neither Rsync nor RRDP are really optimal

What *is* the Erik* Synchronization Protocol?

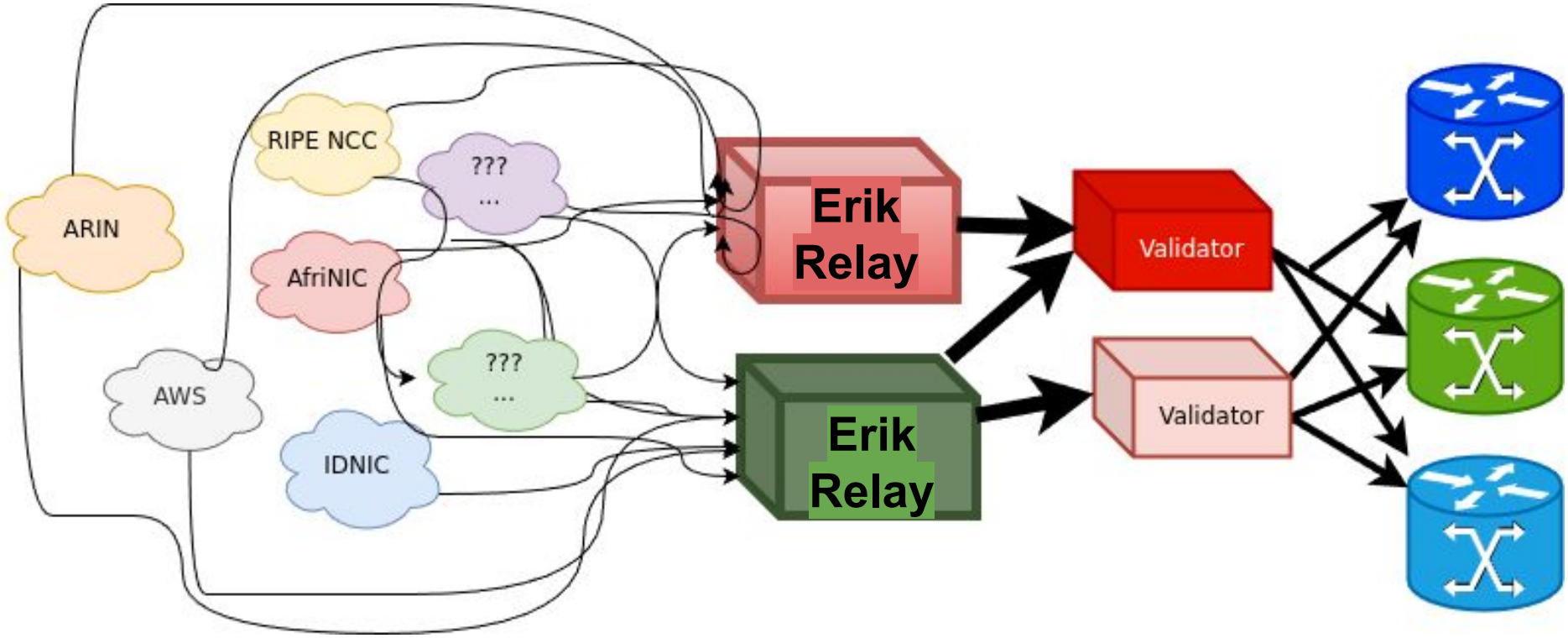
A data replication system using the
following components:

- Merkle Trees (1979)
- Content-addressable naming scheme (1950s?)
- Timestamp-based concurrency control (1975)
- HTTP transport (1989)

*Named in honor of Erik Bais who passed away in 2024.

Possible Erik Protocol deployment strategy

Repositories → *Rsync / RRDP* → **Erik** relay → Validators → BGP Routers



Advantages of the Erik Synchronization Protocol

- Enables clients to “jump straight to latest” (like Rsync)
- Mirroring / replication / offloading is a supported use-case
- Fetch “only what changed” (like Rsync)
- Static pre-calculated content (like RRDP)
- HTTP-based (like RRDP)
- Light on state, no “session” (like Rsync, unlike RRDP)
- Fallback from Erik to Rsync and back is easy

FAST – EFFICIENT – CHEAP

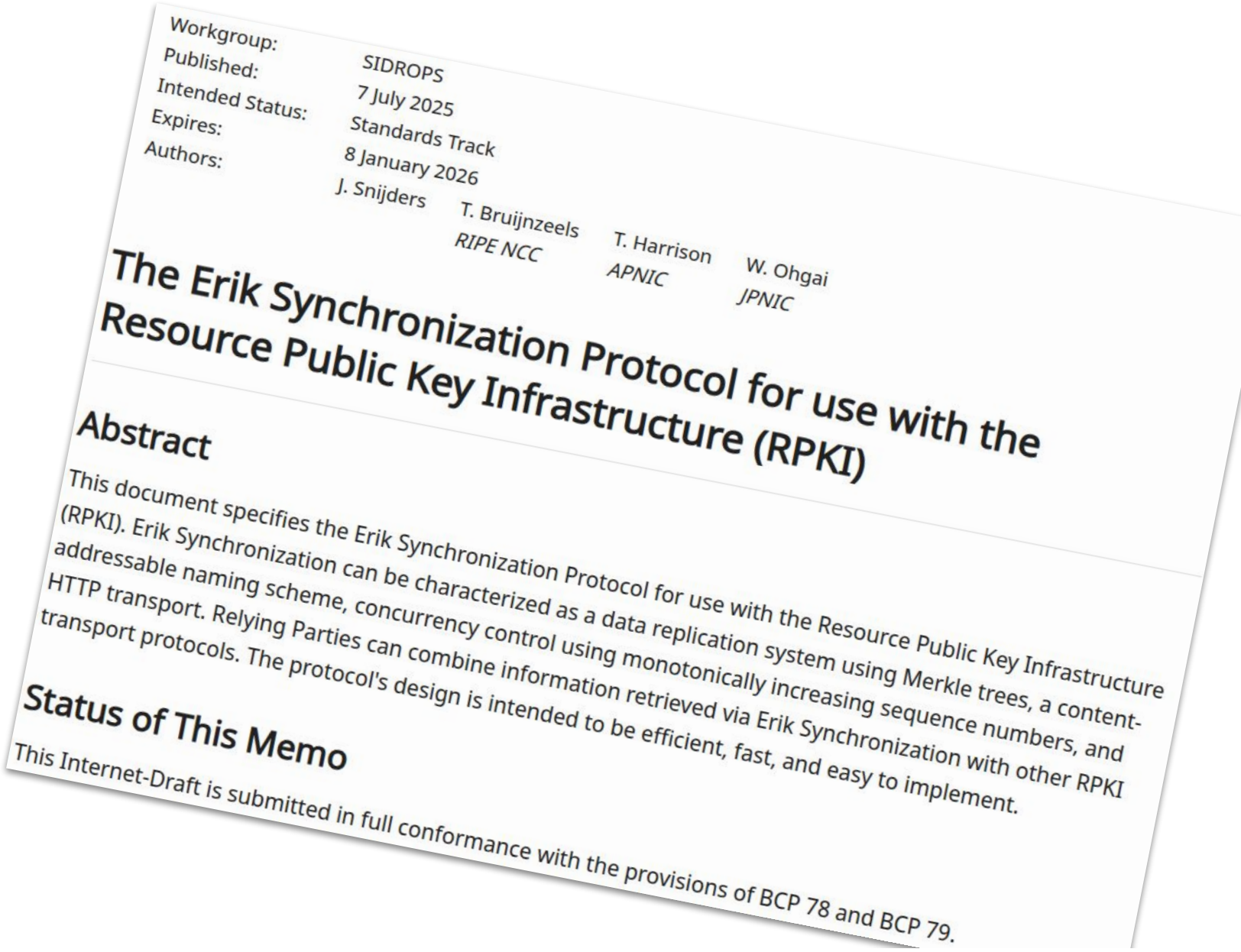
Erik relays are accelerators!

Comparing Rsync and RRDP and **Erik**

Interval	**Rsync**	**RRDP**	**Erik**
Minimum	~ 4 megabytes	~ 0.5 MB	~ 0.5 MB
Every 15 minutes	~ 40 megabytes	~ 5 MB	~ 4 MB
Every 60 minutes	~ 50 megabytes	~ 100 MB	~ 20 MB

The plan

- Write IETF draft
- Write software
- Measure
- Experiment
- Optimize
- Iterate
- Deploy



Workgroup: SIDROPS
Published: 7 July 2025
Intended Status: Standards Track
Expires: 8 January 2026
Authors: J. Snijders, T. Bruijnzeels (RIPE NCC), T. Harrison (APNIC), W. Ohgai (JPNIC)

The Erik Synchronization Protocol for use with the Resource Public Key Infrastructure (RPKI)

Abstract

This document specifies the Erik Synchronization Protocol for use with the Resource Public Key Infrastructure (RPKI). Erik Synchronization can be characterized as a data replication system using Merkle trees, a content-addressable naming scheme, concurrency control using monotonically increasing sequence numbers, and HTTP transport. Relying Parties can combine information retrieved via Erik Synchronization with other RPKI transport protocols. The protocol's design is intended to be efficient, fast, and easy to implement.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.