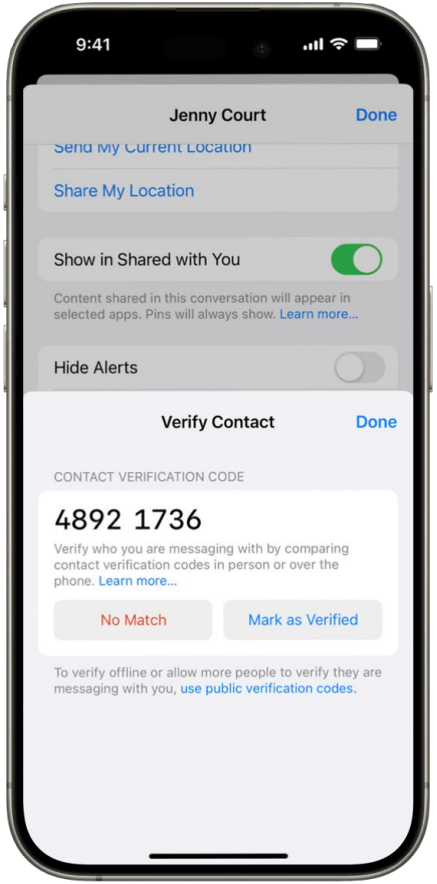
Key Transparency: Problem Statement

(Taylor's Version)

Brendan McMillion
IETF 118 / November 10, 2023

Problem

- E2EE service providers often have difficulty finding secure ways to associate end-user identities with long-term public keys
- Users can sometimes manually verify the public key of each user they communicate with (but people rarely actually do this)
- Compromised key management can undermine any encryption



Solution: Key Transparency

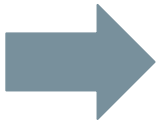
From bofreq:

“Key Transparency (KT) is a safe, publicly-auditable way to distribute cryptographically-sensitive data like public keys.”

Works like a key-value database with two main, cryptographically-assured properties:

1. Alice’s key as seen by Alice = Alice’s key as seen by everyone else
2. Alice’s key today = Alice’s key yesterday + Anything new

Current approach:
Users manually verify that a public key belongs to a specific, real life person



Key Transparency approach:
A user’s device monitors their account for unexpected changes that could be impersonation

This all sounds great but why are you telling me?

Key Transparency has relatively little serious adoption – why?

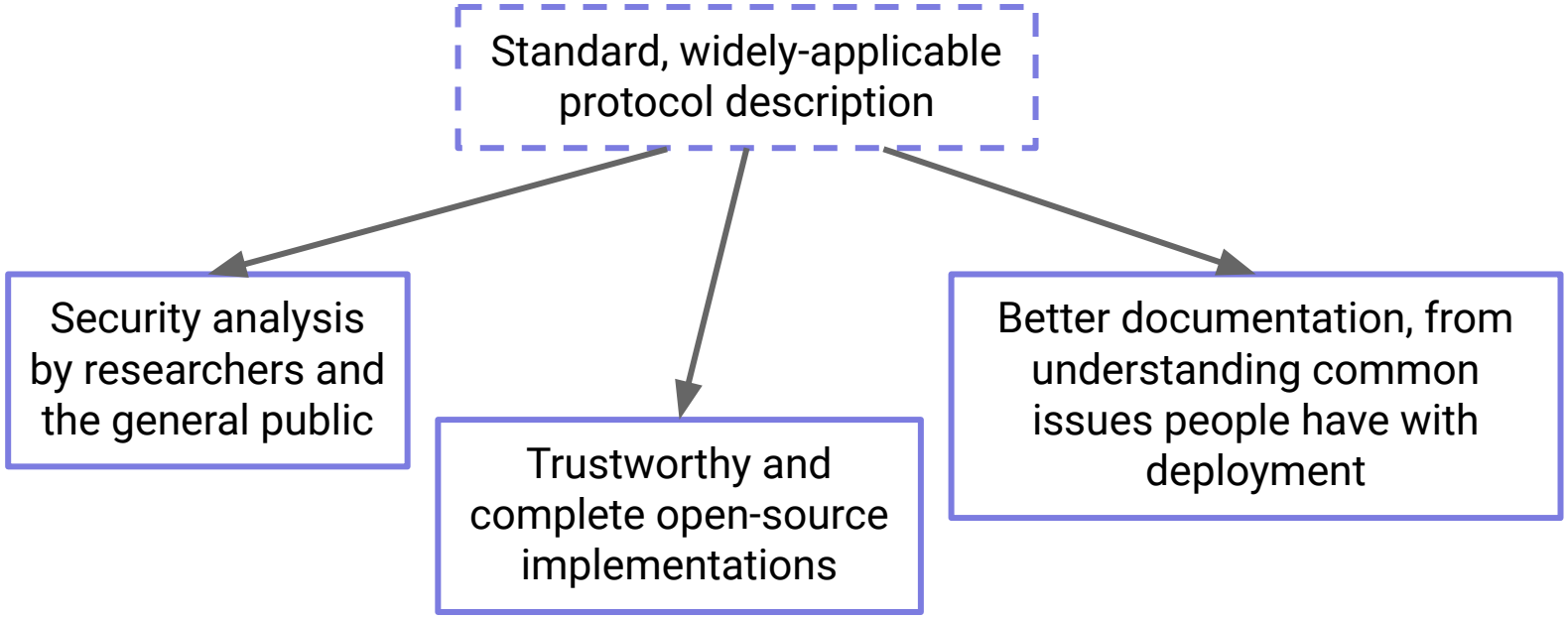
Deploying KT is incredibly difficult:

- Very technically complicated
- Large amount of academic literature
- No guidance on what the “right” choices in the design space are
- Few existing implementations, and those that exist often leave important aspects unresolved
- **Reputational consequences for getting it wrong**
- No trusted, one-size-fits-all protocols or implementations

Even very dedicated implementers get overwhelmed and give up*

* Or their manager tells them to stop

Ideal End Goal



Actually Getting There

Understand the state of what's been deployed and what's possible

(Happened at IETF 116!)

Align a community on a set of common, achievable requirements

- Partially: charter negotiations in interim
- IETF 118: Architecture document...?

Write a protocol that achieves those goals

Charter Conclusions

Authentication Service: ← Potentially different from service provider (to allow federation)

- **Transparent:** All users receive a globally-consistent view
- **User-Friendly:** Little/no user awareness of system ← Any manual verification should be truly optional
- **Private:** Information about a user is only ever revealed to those authorized to know about that user ← Baseline requirement. Will be refined
- **Efficient:** Practical to deploy at Internet-scale

Questions?
Thoughts?