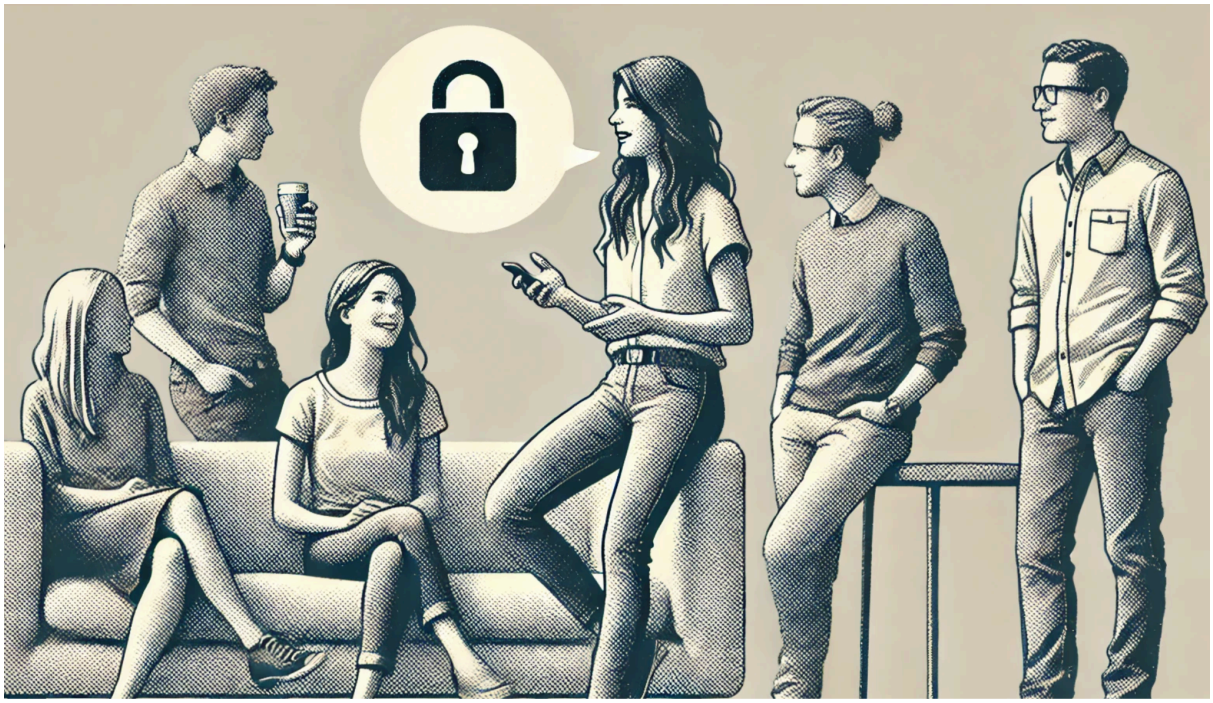
End-to-end Encryption in ActivityPub

AUTHOR 1	Evan Prodromou	AUTHOR 2	Tom Coates

CONCEPT	Every protocol deserves a second chance
TARGET PROTOCOL	ActivityPub (2018) is the distributed social network protocol. It was developed before end-to-end encryption (E2EE) was *de rigeur* for social network messaging. Our challenge is to bring E2EE to this well-established standard.
PHOTO/ SKETCH	
CENTRAL TENSION	"Security versus implementability." We need a protocol secure enough to keep users safe, but simple enough that a large percentage of existing developers can implement it.
LINKS	Safe New World: Protocols that evolve to make their users safer Protocol System Experience: Focusing on the individual experience in a larger protocol system The Death and the Death of Orkut: Sociality, meeting user needs, archiving

HOW WE HOPED TO IMPROVE THE PROTOCOL	**We thought we'd be taking an existing Internet standard, Messaging Layer Security (MLS) RFC 9420, and map it onto ActivityPub as an extension.**
WHAT ACTUALLY HAPPENED?	We backed way, way up and looked at the problem with a wide lens. We reviewed the user interfaces of a dozen other messaging apps, considered the use cases for direct messaging in social network, the kinds of scenarios that users would experience in their daily experience, ways to integrate messaging into ActivityPub, and architectural variations for E2EE. We compared a half-dozen abstract protocols, met with implementers, wrote up a guide to recommended design use, and in the end, decided to take MLS and map it onto ActivityPub as an extension.
WHAT WE LEARNED	1. Building interfaces with backwards compatibility is a serious challenge. 2. The most widely-used abstract protocol in this area, Signal, is under a tight trademark license that makes it difficult to use for an open standard. 3. People want to feel like their messages are safe. Even people sending shopping lists to their roommate and photos of a dripping faucet to their landlord want their messages to be secure. 4. Encryption has multiple time scales. One is at the instant a single message is sent between two people. Another is at the level of a conversation, days or weeks. And a third is at the level of important access being forgotten, lost, or stolen – hopefully a long time, but not never. 5. Technology trends matter. The simplest protocols we considered have been rock-solid for almost three decades, but more recent developments feel new, current, and better. Developers would find the old ones easy and users would find them secure, but we ultimately went with the new hotness to get community buy-in. 6. Only certain kinds of very paranoid people want to do things like compare encryption keys out of band to ensure end-to-end security. But it's important to prove that they're able to do it, so nobody else has to worry about it. 7. In terms of user interface, the basic pattern for doing messaging is well established and clear. The core issues for a person adding E2EE messaging into their client are focused on key management, understanding how archiving works and how to communicate and differentiate encrypted and non-encrypted messaging.