Where Are My Keys?

A Survey of Matrix Cryptographic Key Infrastructure

Sumner Evans

21 September 2024

Beeper (Automattic)

Where Are My Keys?

A Survey of Matrix Cryptographic Key Infrastructure

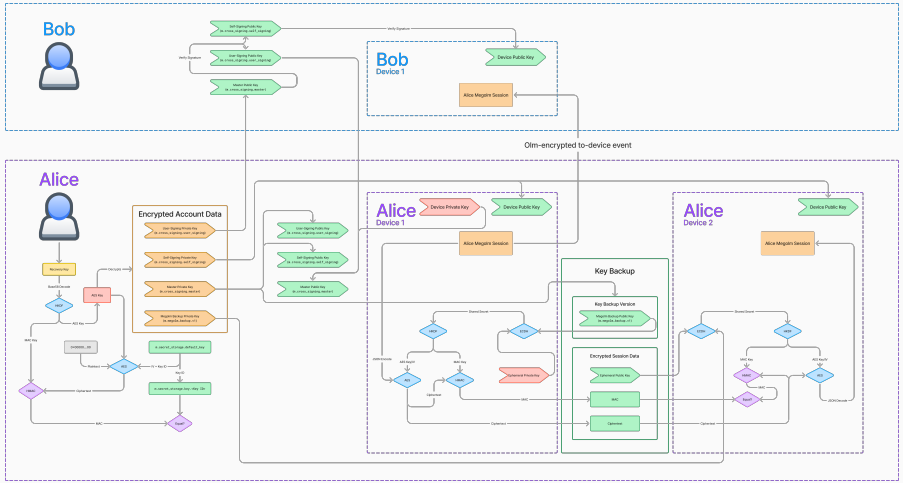
Sumner Evans

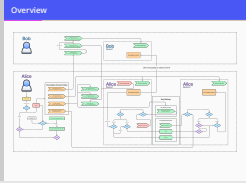
21 September 2024

Beeper (Automattic)

1. Hello, my name is Sumner, I'm a software engineer at Automattic working on Beeper and today I'm going to be talking about cryptographic key infrastructure in Matrix.
2. End-to-end encryption is one of the things which brought me to Matrix, and I'm sure that it's one of the factors that brought many of you to Matrix as well.
3. However, Matrix's user experience with cryptography is often confusing. I mainly blame the other chat networks for their incompetence. Most other chat networks don't even provide any cryptographically-guaranteed security and privacy. Some networks provide encryption in a way that does not truly leave the user in control of their keys. Only a few networks (Signal) truly leave the user in control, and their UX is arguably worse than Matrix.
4. In this talk, my goal is to discuss the cryptographic key *infrastructure* in Matrix. What do I mean by "infrastructure"? I mean all of the features which support key sharing and identity verification, but don't actually themselves provide security. You can think of this talk as discussing the "UX layer of cryptography in Matrix". None of the things that I'm going to discuss are strictly necessary for ensuring secure E2EE communication, but without them, Matrix' UX would be horrible.

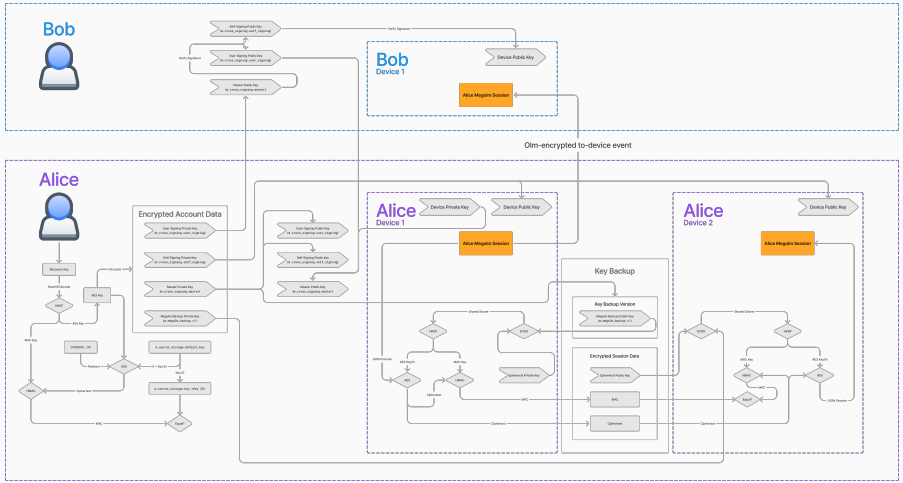
Overview





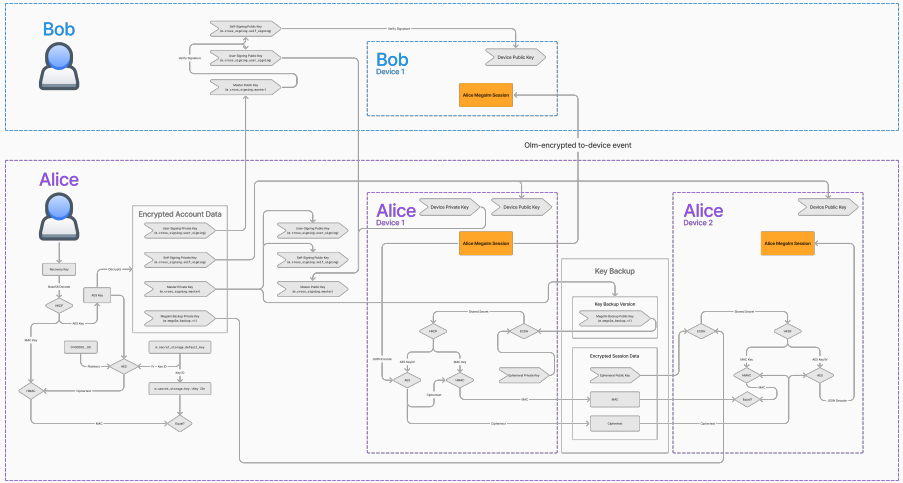
1. This is a diagram of the things we are going to talk about today. This diagram represents all of the infrastructure in Matrix for sharing, backing up, and verifying devices.
2. I know, it's pretty overwhelming. But don't worry, we are going to go step-by-step through this, at the end of the talk you should have an understanding of what each part of this diagram means.
3. Let's start by orienting ourselves to the big picture of this diagram, then we will take a short detour into a few core cryptography concepts required to understand the diagram, and then we will break down the diagram into manageable pieces. And at the end of the talk hopefully you have a complete understanding of Matrix cryptographic key infrastructure.
4. You can see that there are two users represented in the diagram: Bob on the top and Alice on the bottom. The diagram is focused on how the Megolm session created by Alice Device 1 is shared to Bob and to Alice's Device 2.
5. The diagram is color-coded.
   - Red nodes represent data that does not leave the device.
   - Green nodes represent data is public and can be shared with the server and other users.
   - Orange nodes represent data that can be shared with trusted parties, or with members of the same Matrix room.
   - Blue and purple nodes represent cryptographic operations.

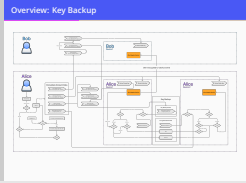
Overview: Keeping the Goal In Mind



1. In orange, we have the core of Matrix security: the Megolm session.
2. We aren't going to discuss this in detail today. I wrote an article about Megolm which you can find on my blog if you want to learn more. I'll provide a link at the end of the talk.
3. But for now, the only thing you need to know about it is that it provides end-to-end encryption for messages and needs to be shared with devices that Alice wants to be able to read her messages. So,
   - other users in the same Matrix room, or
   - other devices belonging to your own user.
4. All of the rest of the infrastructure in this diagram is to facilitate transferring that Megolm session, or verifying that a device should in fact have access to that Megolm session.

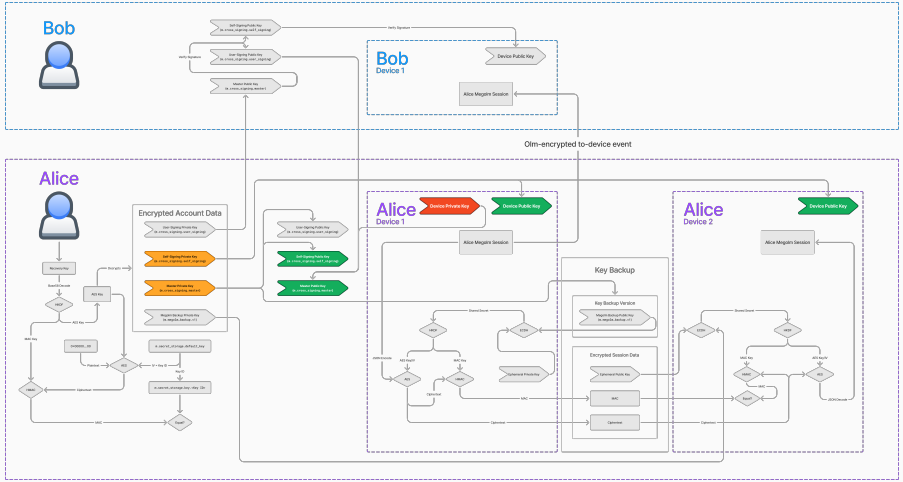
Overview: Key Backup

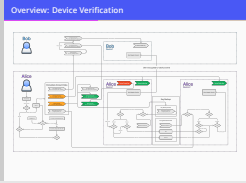




1. For example, in this lower portion of the diagram, we have key backup which allows you to backup your keys to the server for your other devices.

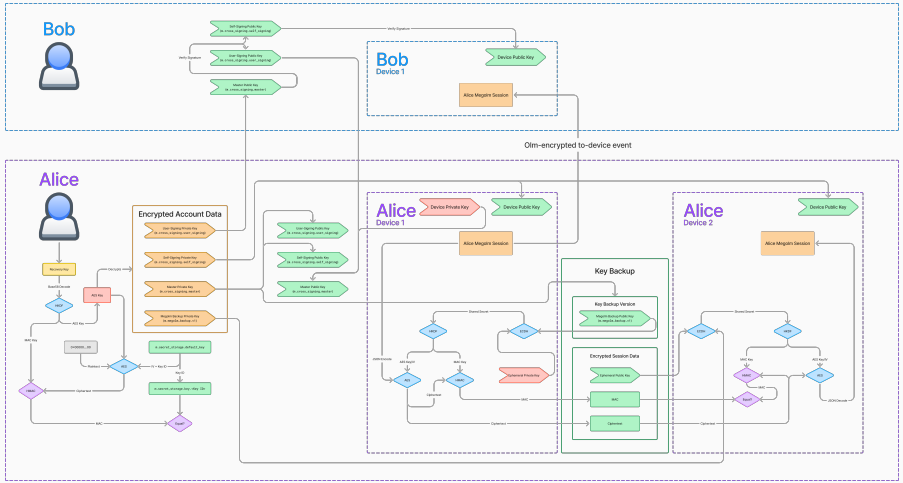
Overview: Device Verification

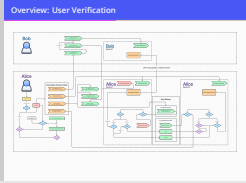




1. Here we have the infrastructure necessary for determining if we trust other of our own devices.

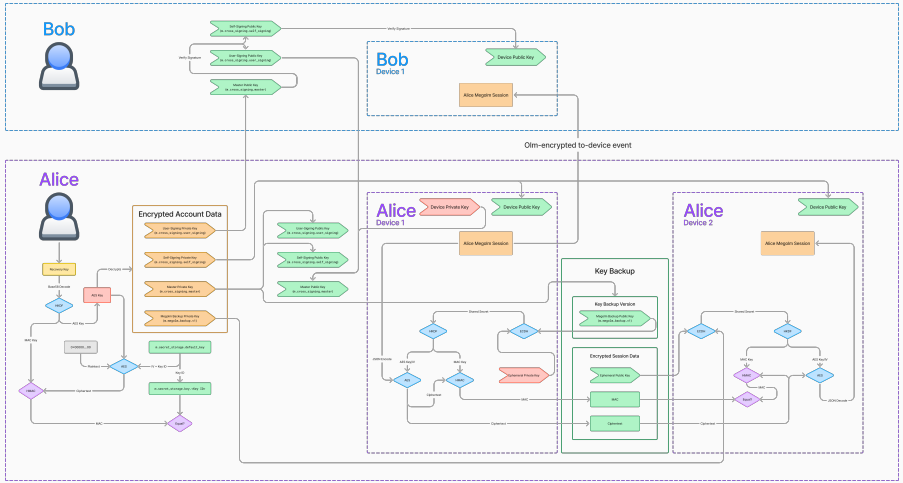
Overview: User Verification

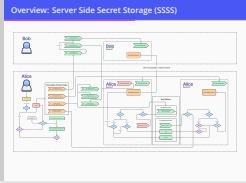




1. And here we have the infrastructure necessary for determining if we trust another user's device.

Overview: Server Side Secret Storage (SSSS)





1. Lastly, on the left we have the infrastructure necessary for storing secrets on the server encrypted by a recovery code.

Cryptography Crash Course

Before we dive deeper into the details of the diagram, we need to discuss some basic cryptography primitives.

I will try and explain these in simple terms. It's not going to be mathematically rigorous, but will focus on the **functionality** that each cryptographic primitive provides.

Encryption: Symmetric vs Asymmetric

Encryption: Symmetric vs Asymmetric

Asymmetric Signatures

Hashes and HMAC

Hashes and HMAC

Key-Derivation Functions (HKDF)

Key-Derivation Functions (HKDF)

Diffie-Hellman Key Exchanges

Diffie-Hellman Key Exchanges

Sharing Keys

Sharing Keys

Thank You for Listening!

I've been Sumner Evans.

TODO slide link

Thank You for Listening!

I've been Sumner Evans.
TODO slide link

Questions?

Questions?