# Hackdays 2025 – End-to-end encryption for La Suite

# CAÇAÏ

Quentin ACHER, Victor-Henrique DE-MOURA-NETTO,
Jean-Philippe EISENBARTH, Lisa FORMENTINI, Ludovic PAILLAT

{firstname.lastname}@inria.fr

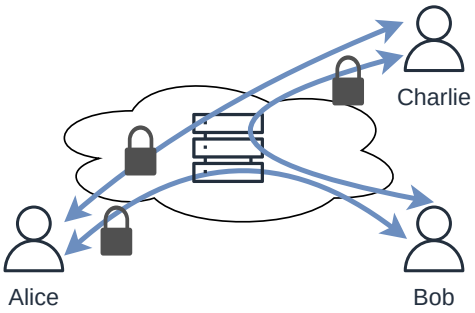




# Current state: Data confidentality in La Suite

- Both La Suite Docs and Visio currently rely on centralized architectures (Hocuspocus, LiveKit).
- Content is encrypted for exchange but decrypted on the server side, where it is processed in plaintext.
- Users are increasingly concerned about their privacy and the security of their personal information.
- Introducing E2EE provides an alternatives to current solutions, such as Google Docs, Microsoft 365 and Notion.

# Our solution




- End to End Encryption (E2EE) ensures that only the sender and the recipient are able to read any exchanged data
- It leverages a cryptographic protocol to negotiate a shared group key used to encrypt the data exchanged between the users
- No eavesdropper (not even a centralized server used by the system) is able to decrypt and read the data

# End-to-end Encryption with MLS Protocol - introduction

New standard to secure communications between groups of users: Messaging Layer Security Protocol (RFC 9420) [1]



Contributors :



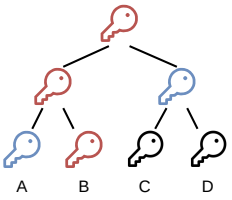


- IETF standard, verified by multiple security analyses,
- multiple open-source implementation available

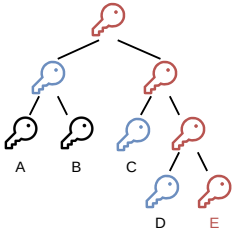
[1] https://www.rfc-editor.org/rfc/rfc9420.html

# End-to-end Encryption with MLS Protocol - performance

The key advantage of the MLS Protocol is its ability to scale thus supporting large groups of users. This is possible with the help of the structure called "Ratcheting Tree"s



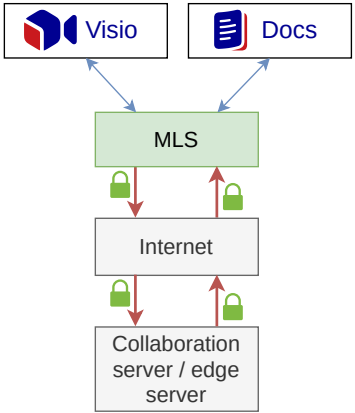

Key update for member B




Adding new member E

- The protocol was establish a tree consisting of intermediate keys. Each intermediate key actually represents a key that is shared by a sub-group in the tree.
- Therefore when a sub-group is unchanged (i.e. no remove operation, no key update) we can directly use that key to encrypt new secrets (i.e. the key updates of other members or in case of remove).

# How to integrate MLS in La Suite applications

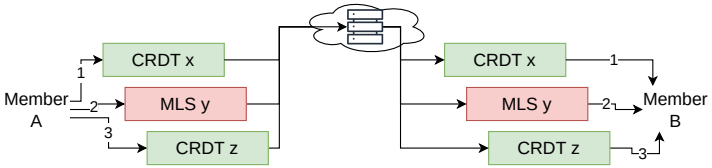



- Due to the conflict-free nature of the exchanged content we do not need a central authority to coordinate users for data :
  - defined with CRDT like Docs (related to the use of Yjs),
  - with individual data, like audio and video stream for Visio,

→ Therefore, we can simply integrate MLS by using the encryption keys provided by MLS to secure all exchanged communications.

# Asynchronous capabilities of the protocol

One concern might be wherever solutions such as Docs can continue to work, and encryption keys can be updated when most group members are not currently connected.




When the centralized service, such as the collaboration server of Docs, receive the messages whose content are protected with MLS, it attributes a sequence number to keep track of a strict order of messages.
Then, if members process all messages in the same order, they will stay in a coherent state and keep the same view of the application data as well as MLS encryption keys.
This order will conveys information such as the need to process an MLS message to be able to decrypt the next application messages.

# Our Proof of Concept: MUTE

- MUTE[2] is a scalable collaborative document editor developped by the COAST research group at the Inria Centre at Université de Lorraine
- It implements a CRDT-based consistency algorithm (LogootSplit) for collaboration. It makes it very similar to La Suite's Docs.
- It also features end-to-end encrypted peer-to-peer communications.

$\Longrightarrow$ E2EE architecture can be implemented in La Suite apps

[2] https://github.com/coast-team/mute

# Next Steps

Currently MLS leverages a delivery service which is centralized. La Suite Docs also uses a centralized collaboration server (Hocuspocus). So the next step is :

- Managing MLS in a distributed manner is an ongoing research topic within the COAST Team at Inria Centre at Université de Lorraine.