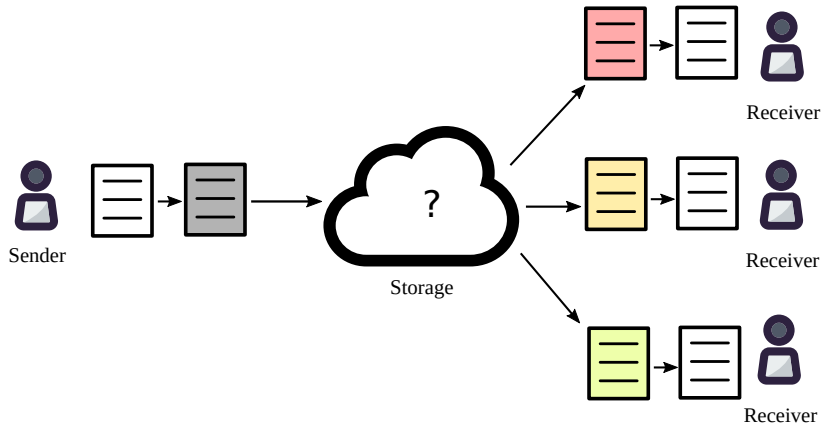


Michael Egorov and Justin Holmes

ETHBerlin, 7 Sep 2018

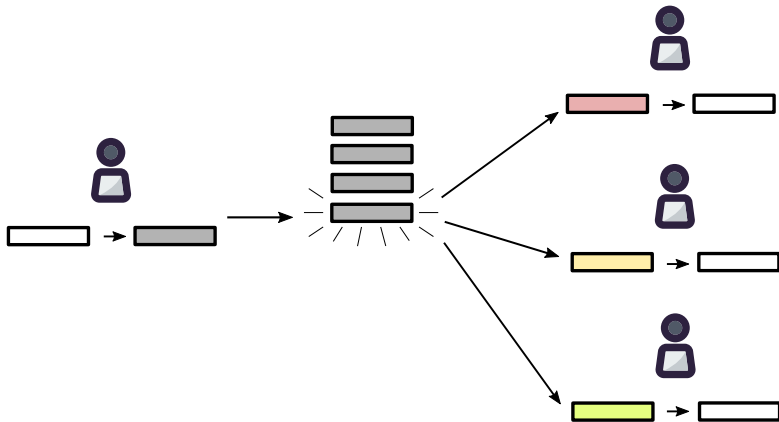
# Why

## Encrypted file sharing



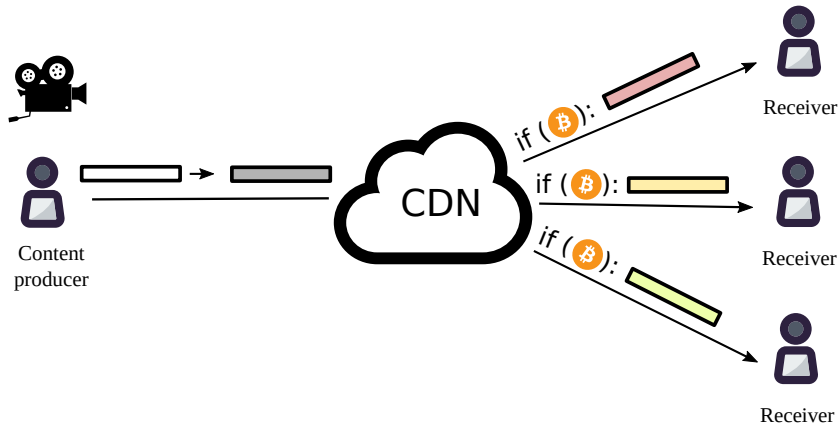
# Why

## Encrypted multi-user chats



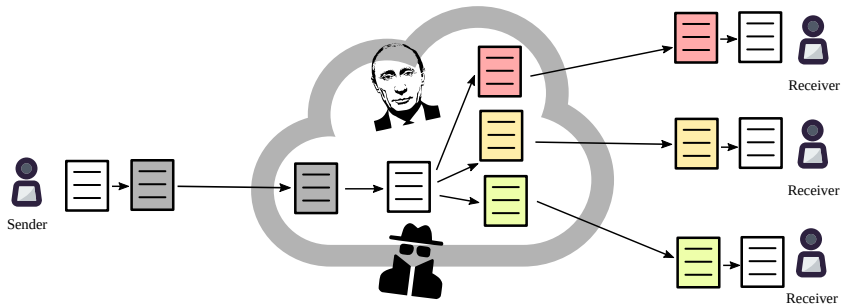
# Why

## Decentralized Netflix



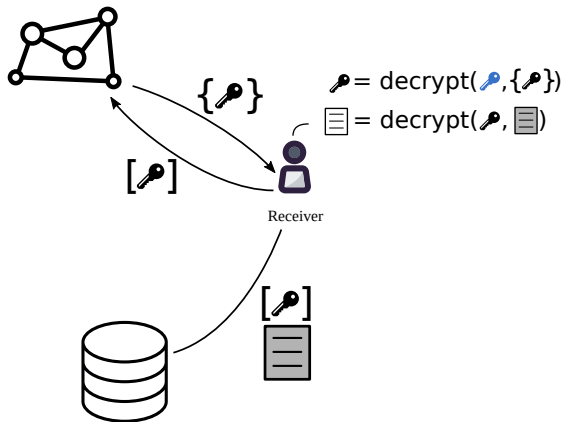
# Central server + TLS

Data vulnerable to hackers, state actors etc

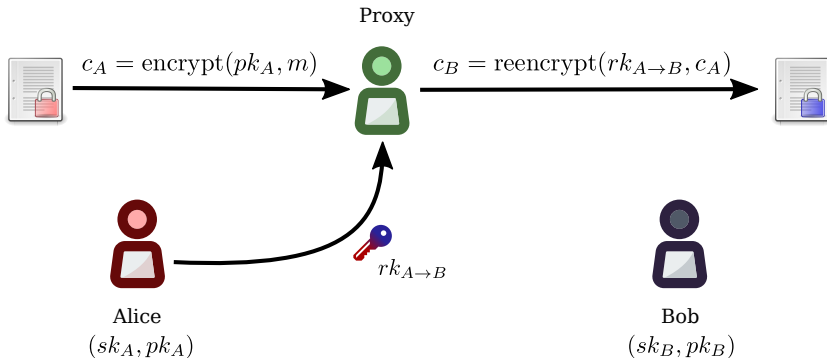


# Solution

## Proxy re-encryption + decentralization

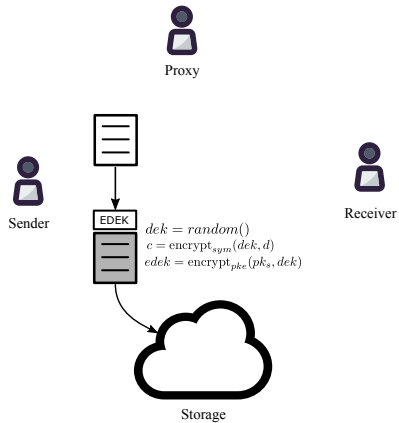


# What is proxy re-encryption (PRE)



# Centralized KMS using PRE

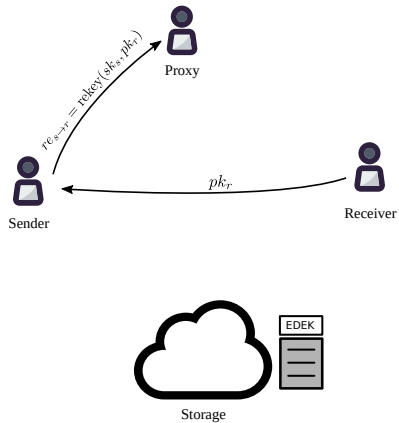
## Encryption





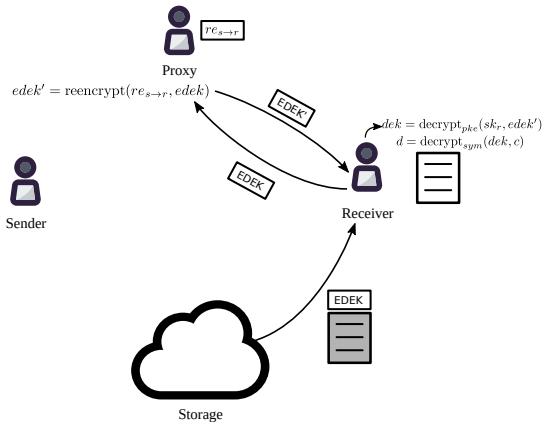
# Centralized KMS using PRE

## Access delegation



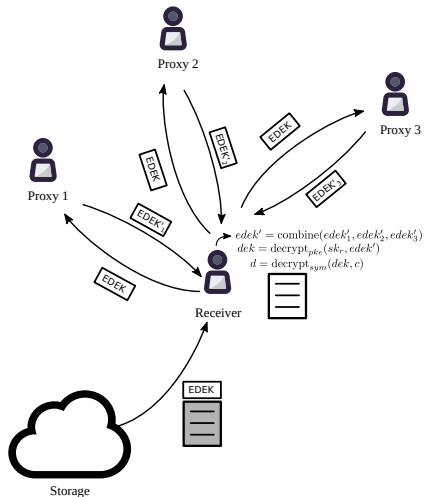
# Centralized KMS using PRE

## Decryption



# Decentralized key management

Using threshold split-key re-encryption (Umbral)



<https://github.com/nucypher/nucypher-kms/>

# Types of policies

- Time-based;
- On payment (“grant access once paid, continue granting while paying”);
- Smart contract (public) method.

Ursulas are trusted to apply conditions without decrypting data

# Umbral: threshold proxy re-encryption

- “Umbral” is Spanish for “threshold”
- PRE properties: Unidirectional, single-hop, non-interactive
- It follows a KEM/DEM approach:
  - ▶ UmbralKEM provides the threshold re-encryption capability
  - ▶ Uses ECIES for key encapsulation with zero knowledge proofs of correctness for verifiability on prime order curves (such as secp256k1)
  - ▶ The DEM can be any authenticated encryption (currently ChaCha20-Poly1305)
- IND-PRE-CCA security
- Verification of re-encryption correctness through Non-Interactive ZK Proofs
- Reference implementation: <https://github.com/nucypher/pyUmbral/>
- Documentation (WIP): <https://github.com/nucypher/umbral-doc>

# NU token

## Purpose

- Splitting trust between re-encryption nodes (more tokens = more trust and more work);
- Proof of Stake for minting new coins according to the mining schedule;
- Security deposit to be at stake against malicious behavior of nodes

# Early Users

## Medical & Biotech Data Platform



## Key Management



## Automobile Data



## Decentralized Application



NuCypher  
Ecosystem

## Location Data

DATAEUM

## Genomics Data



## Medical Data Protocol



## Protocol & Infrastructure

## Internet of Things



## Decentralized Database



## Sharing Economy



## User Data Monetization



# Fully Homomorphic Encryption

## nuFHE Library

- GPU implementation of fully homomorphic encryption
- Uses either FFT or integer NTT
- GitHub: <https://github.com/nucypher/nufhe>
- Achieved 100x performance over TFHE benchmarks

Platform	Library	Performance (ms/bit)	
		Binary Gate	MUX Gate
Single Core/Single GPU - FFT	TFHE (CPU)	13	26
	nuFHE	0.13	0.22
	<b>Speedup</b>	<b>100.9</b>	<b>117.7</b>
Single Core/Single GPU - NTT	cuFHE	0.35	N/A
	nuFHE	0.35	0.67
	<b>Speedup</b>	<b>1.0</b>	-



## Useful links



**Website:** <https://nucypher.com>

**NuCypher network:** <https://github.com/nucypher/nucypher/>

**PyUmbral:** <https://github.com/nucypher/pyUmbral/>

**GoUmbral:** <https://github.com/nucypher/goUmbral/>

**Mocknet:** <https://github.com/nucypher/mock-net/>

**Discord:** <https://discord.gg/7rmXa3S>

**Whitepaper:** <https://www.nucypher.com/whitepapers/english.pdf>

**E-mail:** [hello@nucypher.com](mailto:hello@nucypher.com)

## PRE demo



Network you can run: <https://github.com/nucypher/nucypher/>