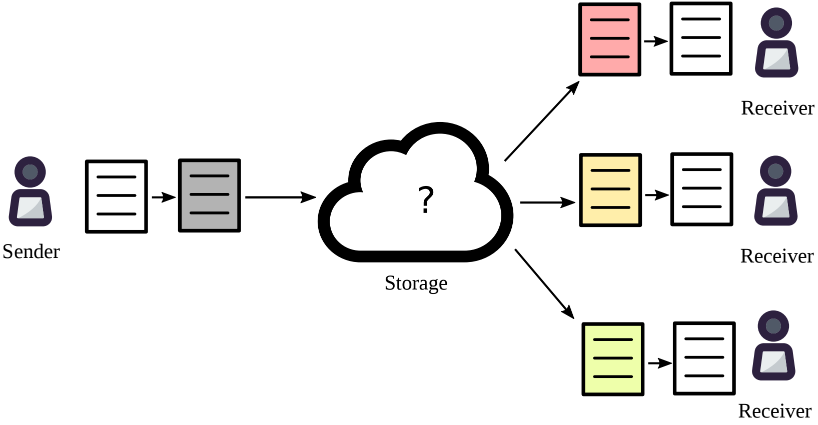
NuCypher KMS: Decentralized Key-Management System

Michael Egorov, CTO

BPASE, 26 Jan 2018

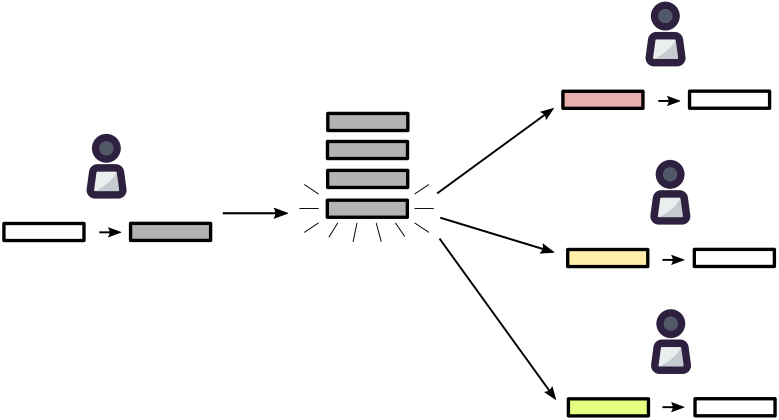
Why

Encrypted file sharing



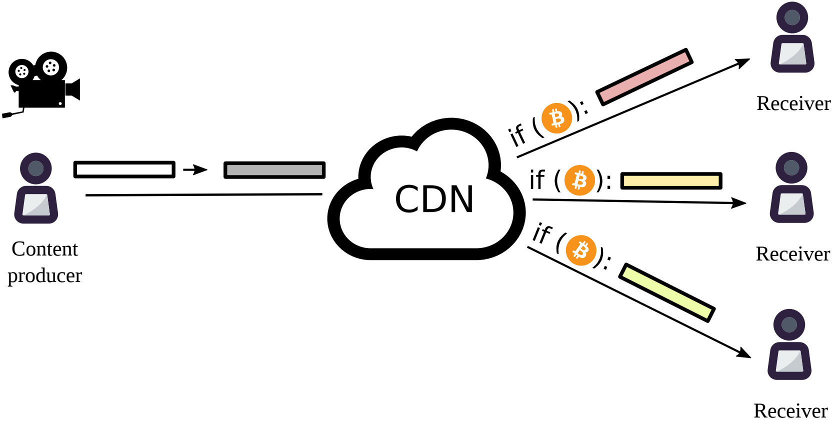
Why

Encrypted multi-user chats



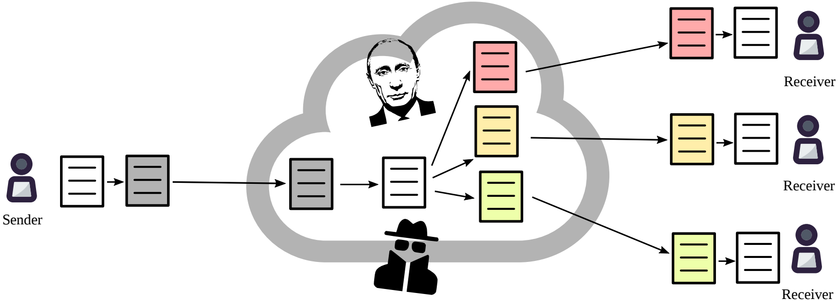
Why

Decentralized Netflix



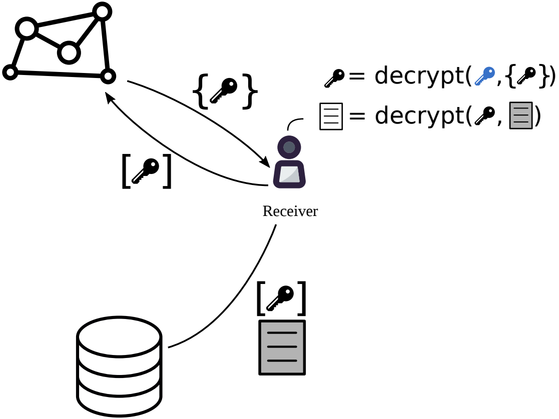
Central server + TLS

Data vulnerable to hackers, state actors etc



Solution

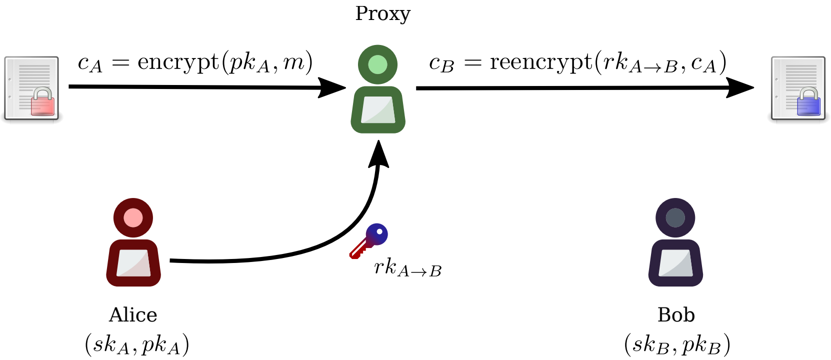
Proxy re-encryption + decentralization



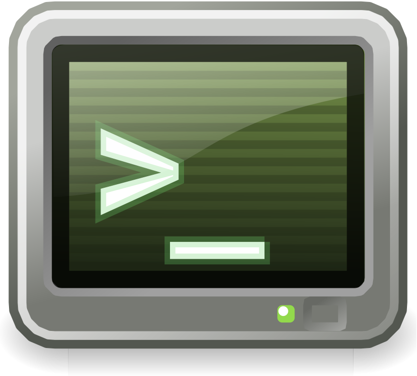
🔑= decrypt(🔑,{🔑})

📄= decrypt(🔑, 📄)

What is proxy re-encryption (PRE)

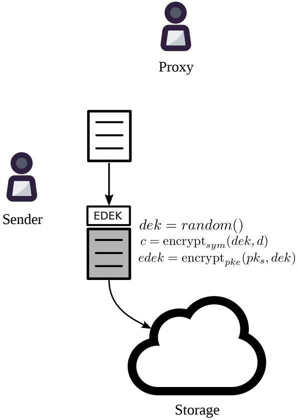

Proxy
$c_A = \text{encrypt}(pk_A, m)$
$c_B = \text{reencrypt}(rk_{A\to B}, c_A)$
$rk_{A\to B}$
Alice
(sk_A, pk_A)
Bob
(sk_B, pk_B)

PRE demo



Centralized KMS using PRE

Encryption



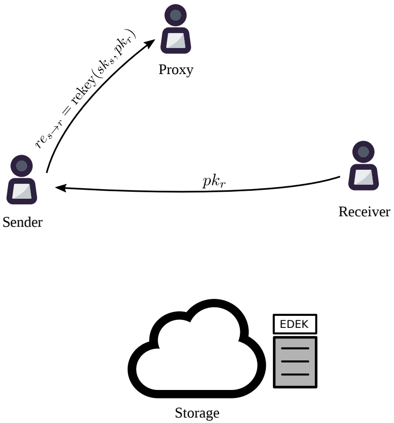
$dek = random()$

$c = \mathrm{encrypt}_{sym}(dek, d)$

$edek = \mathrm{encrypt}_{pke}(pk_s, dek)$

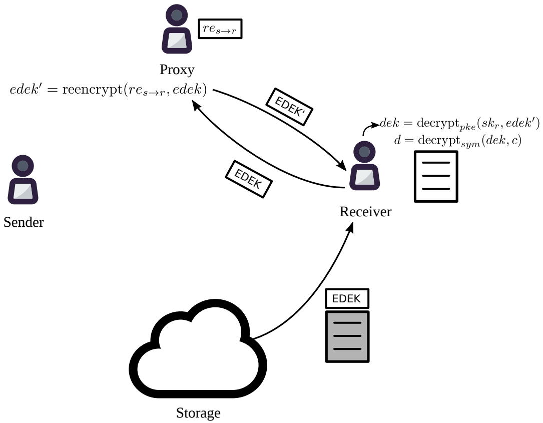
Centralized KMS using PRE

Access delegation



Centralized KMS using PRE

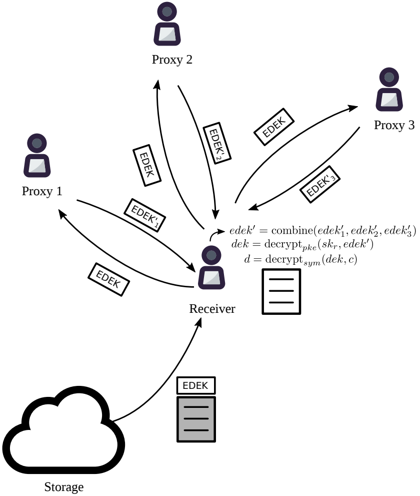
Decryption



$re_{s\to r}$
Proxy
$edek' = \text{reencrypt}(re_{s\to r}, edek)$
EDEK'
$dek = \text{decrypt}_{pke}(sk_r, edek')$
$d = \text{decrypt}_{sym}(dek, c)$
EDEK
Sender
Receiver
EDEK
Storage

Decentralized key management

Using threshold split-key re-encryption (Umbral)



$$edek' = \text{combine}(edek'_1, edek'_2, edek'_3)$$
$$dek = \text{decrypt}_{pke}(sk_r, edek')$$
$$d = \text{decrypt}_{sym}(dek, c)$$

`https://github.com/nucypher/nucypher-kms/`

`https://github.com/nucypher/pyUmbral/`

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
  - The DEM can be any authenticated encryption (currently ChaCha20-Poly1305)
- IND-PRE-CCA security
- Verification of re-encryption correctness through Non-Interactive ZK Proofs of Knowledge
- Code: `https://github.com/nucypher/pyUmbral/`
- Documentation (WIP): `https://github.com/nucypher/umbral-doc`

KMS token

Purpose

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

KMS token

Mining

Mining reward:

$$\text{reward} = \frac{\text{locked\_tokens} \times \text{reward\_rate}}{\sum_{\text{all miners}} \text{locked\_tokens}} + \sum_{\text{this miner}} \text{miner\_fees}$$

Early users

Decentralized marketplaces:

- Datum.

Decentralized databases:

- Bluzelle;
- Fluence;
- Wolk.

Medical data sharing

- Medibloc;
- IRYO;
- Medixain;
- Wholesome.

IoT

- Spherity (together with BigchainDB).

Cryptocurrency keys

- Coval Emblem Vault

Investors

POLYCHAIN CAPITAL

GALAXY
DIGITAL ASSETS

compound

Y

F | B | G CAPITAL

Satoshi•Fund

Kenetic Capital

DHVC

AMINO Capital

semantic capital

BASE

MISSION MARKET

1kx

CoinFund

Blockchain Partners Korea

FIRST MATTER

Team

Founders

Advisors

How to contribute, learn

Website: `https://nucypher.com/blockchain.html`
Github: `https://github.com/nucypher/`
Slack: `https://nucypher-kms-slack.herokuapp.com/`
Whitepaper: `https://arxiv.org/abs/1707.06140`
E-mail: `michael@nucypher.com`