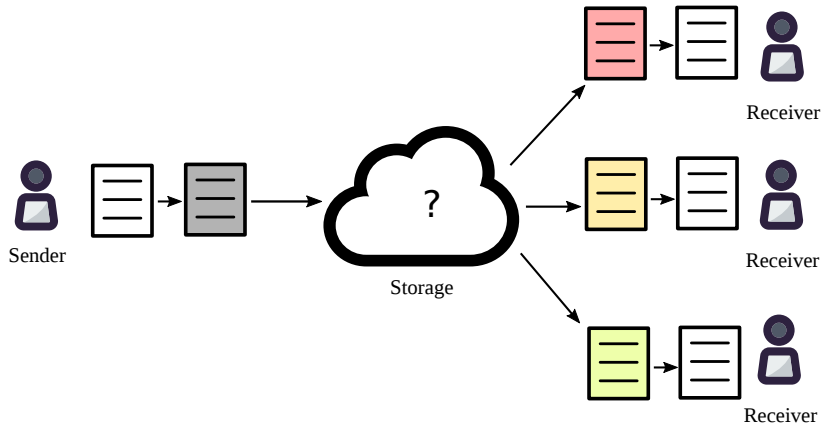# NuCypher

<fname lname>

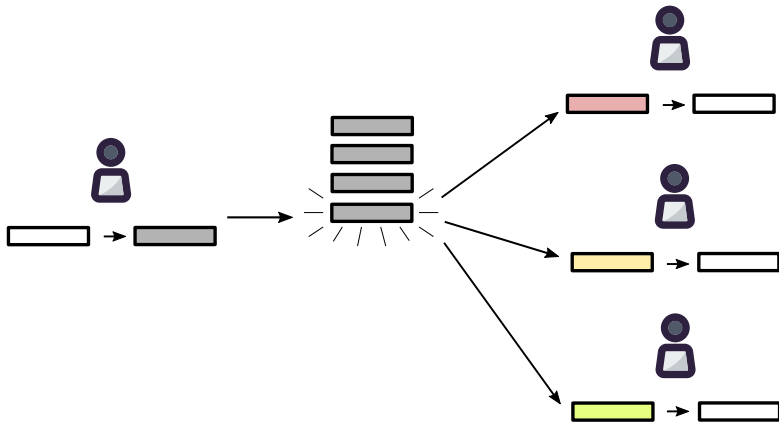**<event name>, <dd Month yyyy>**

Why
Encrypted file sharing

Why

Encrypted multi-user chats

Why
Decentralized Netflix

Content producer

CDN

if (₿): Receiver

if (₿): Receiver

if (₿): Receiver
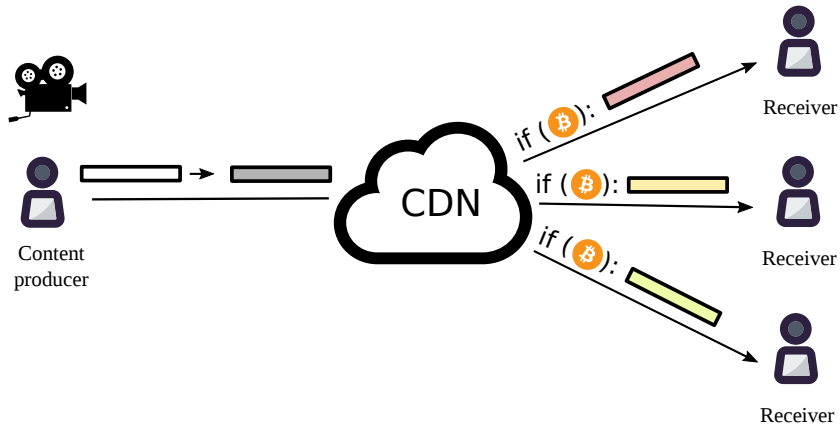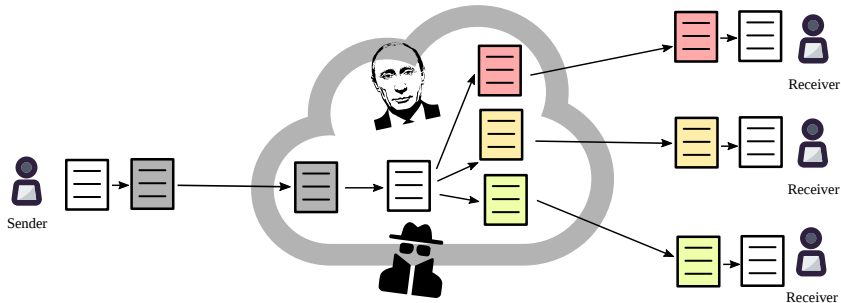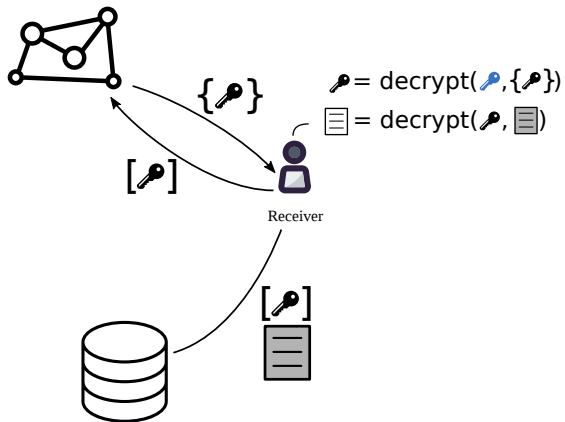
Central server + TLS

Data vulnerable to hackers, state actors etc

Solution

Proxy re-encryption + decentralization

$$\text{\pointkey} = \text{decrypt}(\bluekey, \{\pointkey\})$$
$$\boxed{\equiv} = \text{decrypt}(\pointkey, \boxed{\equiv})$$

Receiver

What is proxy re-encryption (PRE)

Proxy

$c_A = \text{encrypt}(pk_A, m)$

$c_B = \text{reencrypt}(rk_{A \rightarrow B}, c_A)$

$rk_{A \rightarrow B}$

Alice
(sk_A, pk_A)

Bob
(sk_B, pk_B)
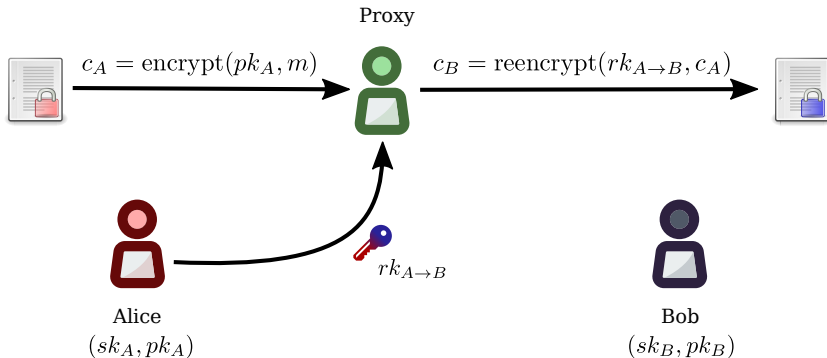
Centralized KMS using PRE

Encryption

Proxy

Sender

EDEK

$dek = random()$
$c = \text{encrypt}_{sym}(dek, d)$
$edek = \text{encrypt}_{pke}(pk_s, dek)$

Receiver

Storage
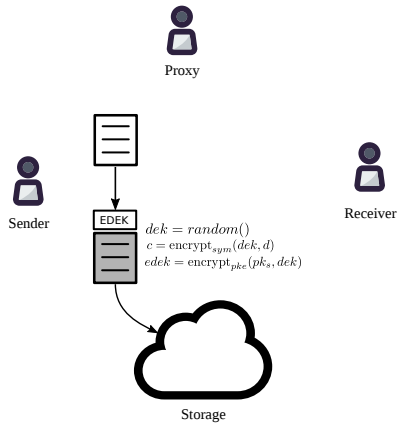
Centralized KMS using PRE

Access delegation

Proxy

$re_{s \to r} = \text{rekey}(sk_s, pk_r)$

Sender

pk_r

Receiver
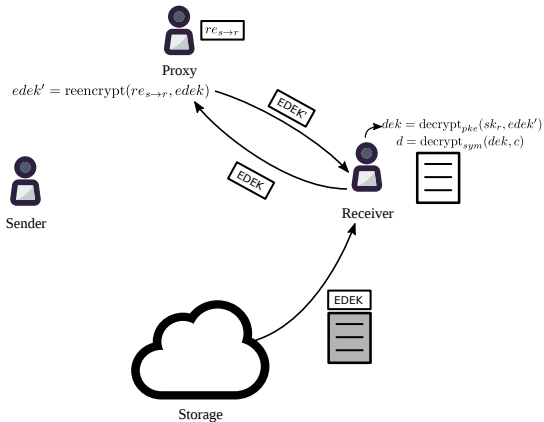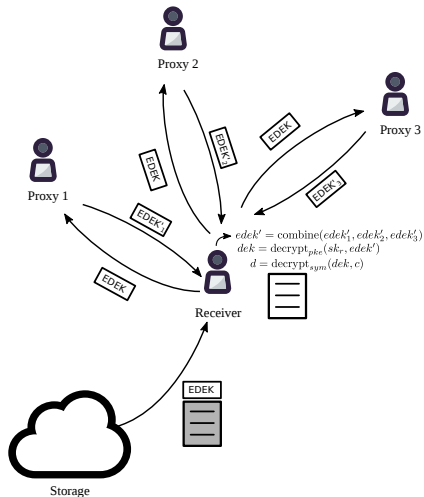
EDEK

Storage
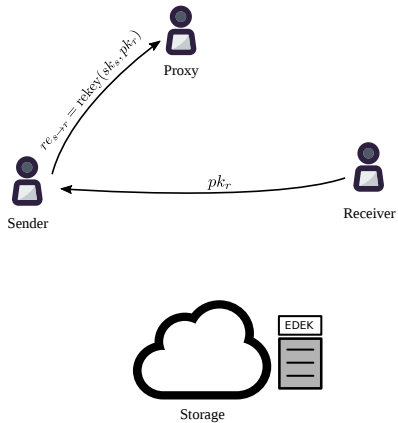
Centralized KMS using PRE

Decryption

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/

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
  - ▸ Encryption is ECIES with extra things for proof of correctness, on curve 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
- Code: `https://github.com/nucypher/pyUmbral/`
- Documentation (WIP): `https://github.com/nucypher/umbral-doc`

PRE demo

Demo network: `https://github.com/nucypher/mock-net/`

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

NU token

Mining

Mining reward:

$$\kappa = \left(0.5 + 0.5\frac{\min(\mathsf{T_i}, \mathsf{T_1})}{\mathsf{T_1}}\right) \tag{1}$$

$$\mathsf{T_{i,initial}} \geq \mathsf{T_{min}}, \tag{2}$$

$$\delta\mathsf{s_{i,t}} = \kappa\frac{\mathsf{l_i}}{\sum \mathsf{l_j}}\frac{\ln 2}{\mathsf{T}_{1/2}}\left(\mathsf{S_{max}} - \mathsf{S_{t-1}}\right). \tag{3}$$

$$\tag{4}$$

Results into:

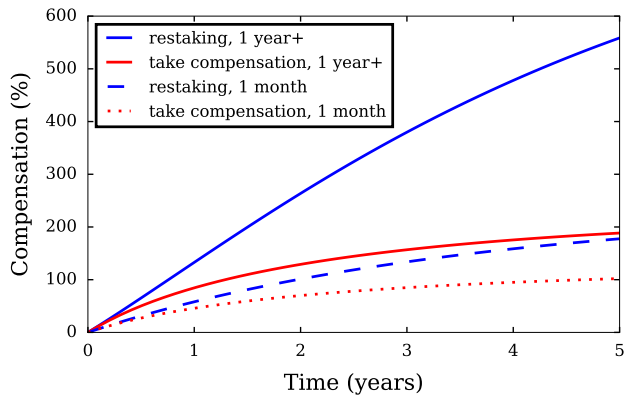$$\mathsf{reward} \propto 2^{\frac{\mathsf{t}}{\mathsf{T}_{1/2}}}$$

NU token

Graph of daily mining compensation
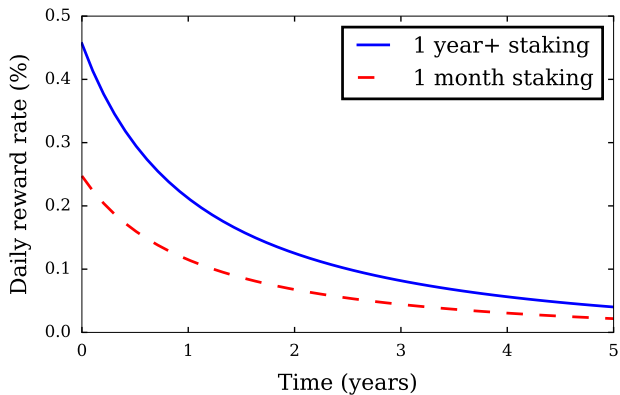
NU token

Relocking mining rewards

Usage examples

Decentralized marketplaces:

- Datum;
- Origin protocol;
- The Seam;
- SwipeCrypto.

Decentralized databases:

- Bluzelle;
- Fluence;
- Wolk.

Medical data sharing

- Medibloc;
- IRYO;
- Medixain;
- Wholesome;
- Medcredits;
- HealthCombix / PointNurse;
- Genobank;
- iku.network.

IoT

- Spherity (together with BigchainDB);
- Carblock.io;
- XAIN.

Cryptocurrency keys

- Coval Emblem Vault.

Useful links

Website: `https://nucypher.com`
Github: `https://github.com/nucypher/`
PyUmbral on Github: `https://github.com/nucypher/pyUmbral/`
Mochnet: `https://github.com/nucypher/mock-net/`
Discord: `https://discord.gg/7rmXa3S`
Whitepaper: `https://arxiv.org/abs/1707.06140`
E-mail: `michael@nucypher.com`
E-mail: `hello@nucypher.com`