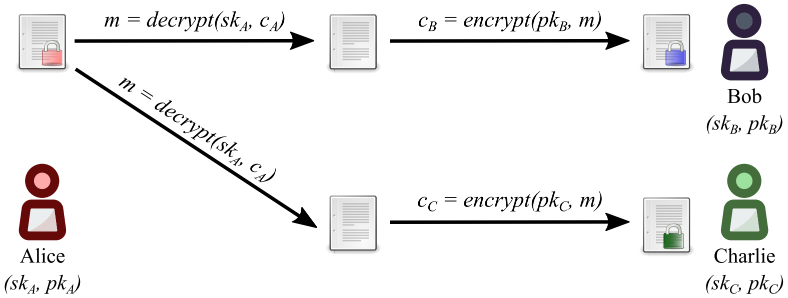
# NuCypher

<presenter name(s), role(s)>

<event name>, <dd MMM yyyy>

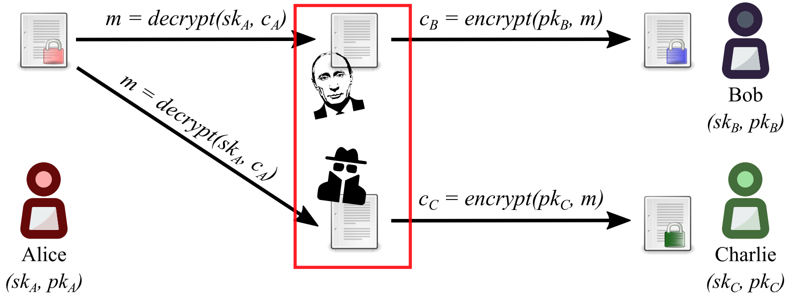
## Public Key Encryption (PKE)



Limitations

- Decryption required before sharing
- Not scalable
- Complex access revocation

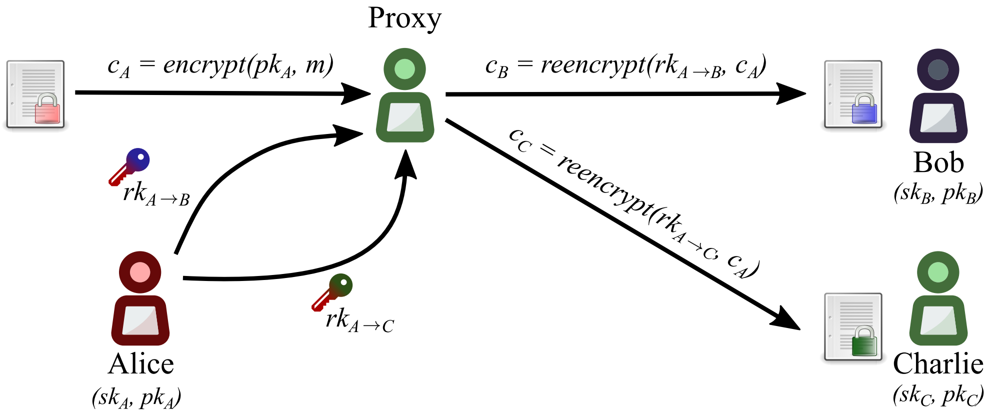
## Public Key Encryption (PKE)



Limitations

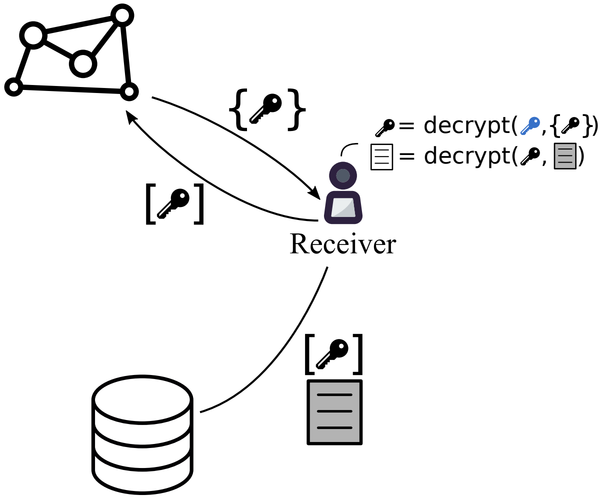
- Decryption required before sharing
- Not scalable
- Complex access revocation

## What is proxy re-encryption (PRE)



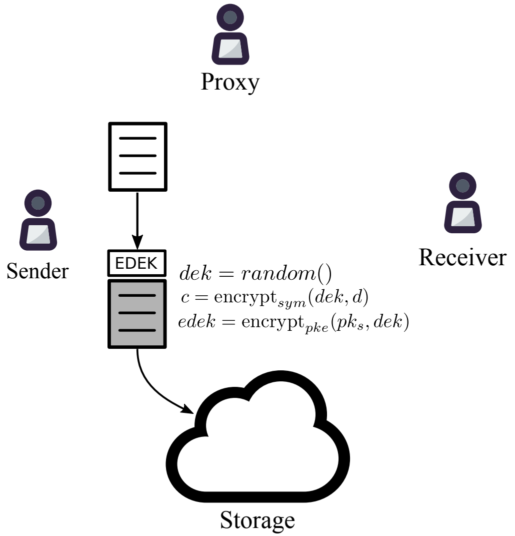
## Solution

### Proxy re-encryption + Key Management



## Centralized KMS using PRE

### Encryption



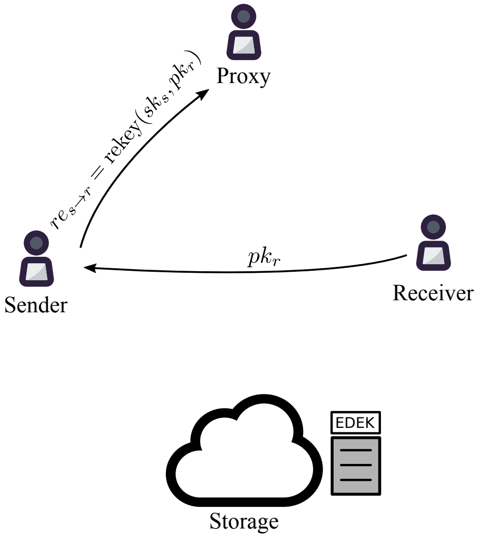
$dek = random()$

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

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

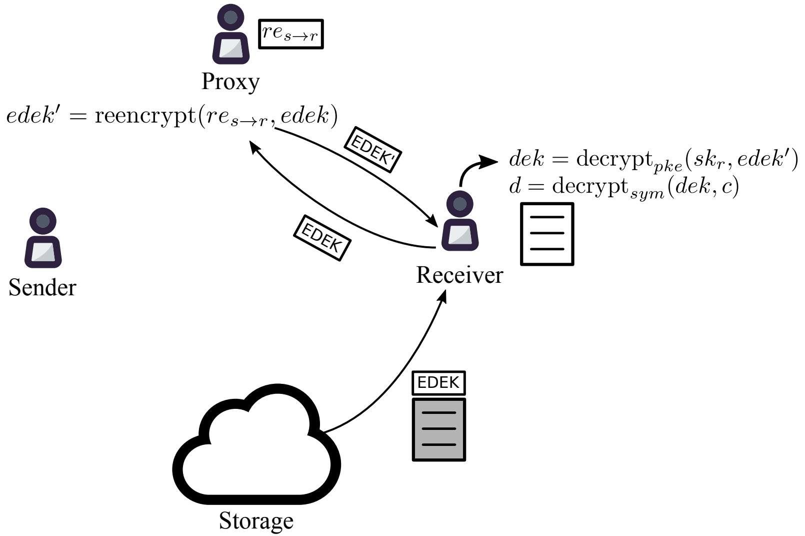
## Centralized KMS using PRE

### Access delegation



# Centralized KMS using PRE

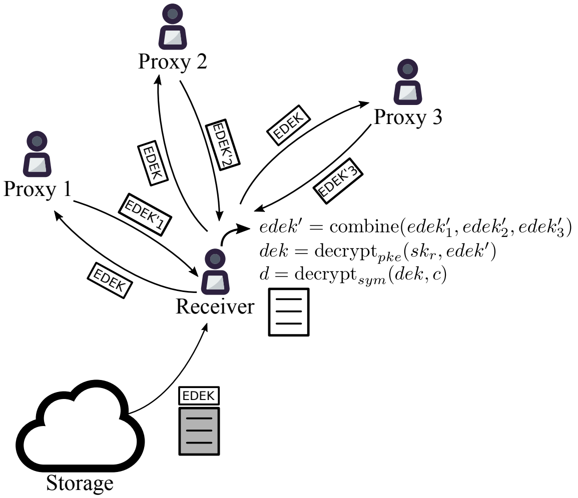
## Decryption



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

# Decentralized Key Management

## Using threshold split-key re-encryption (Umbral)



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

## 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`

## Types of policies

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

## 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:

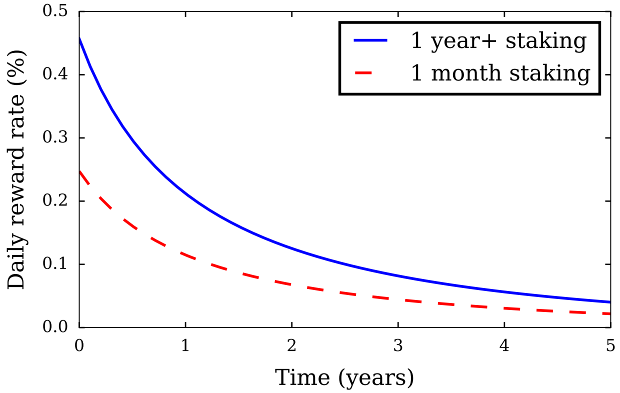
$$\begin{aligned}
\kappa &= \left(0.5 + 0.5\frac{\min(\mathsf{T_i}, \mathsf{T}_1)}{\mathsf{T}_1}\right) && (1)\\
\mathsf{T_{i,initial}} &\geq \mathsf{T}_{\min}, && (2)\\
\delta \mathsf{s_{i,t}} &= \kappa \frac{\mathsf{l_i}}{\sum \mathsf{l_j}} \frac{\ln 2}{\mathsf{T}_{1/2}} \left(\mathsf{S}_{\max} - \mathsf{S}_{\mathsf{t}-1}\right). && (3)\\
& && (4)
\end{aligned}$$

Results into:

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

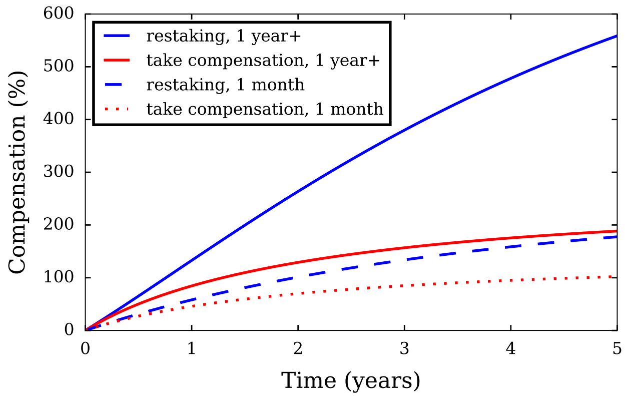
## NU token

### Graph of daily mining compensation



# NU token

## Relocking mining rewards



## Security Audits

# Early Users

**Medical & Biotech Data Platform**

MEDIBLOC

MedCredits

medixain

PointNurse

wholesome
BUSINESS-IN-A-BOX FOR DOCTORS

IKU
BIOTECH REDEFINED

**Key Management**

EMBLEM

**Automobile Data**

CarBlock

**Decentralized Application**

**NuCypher Ecosystem**

**Location Data**

DATAEUM

**Genomics Data**

博士链
DOCTOR CHAIN

**Medical Data Protocol**

IRYO

HealthDex

**Decentralized Database**

bluzelle

fluence.ai

WOLK

**Protocol & Infrastructure**

**Sharing Economy**

ORIGIN

**Internet of Things**

SPHERITY

XAIN.

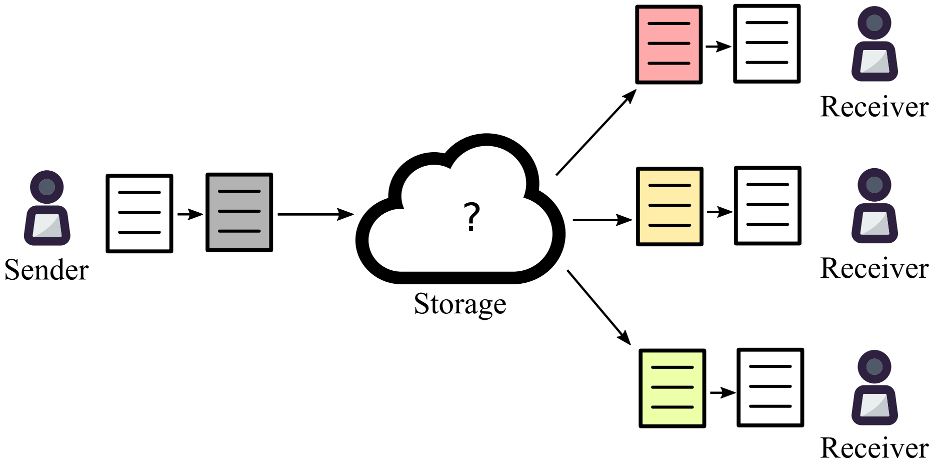
**User Data Monetization**

SWIPE

datum

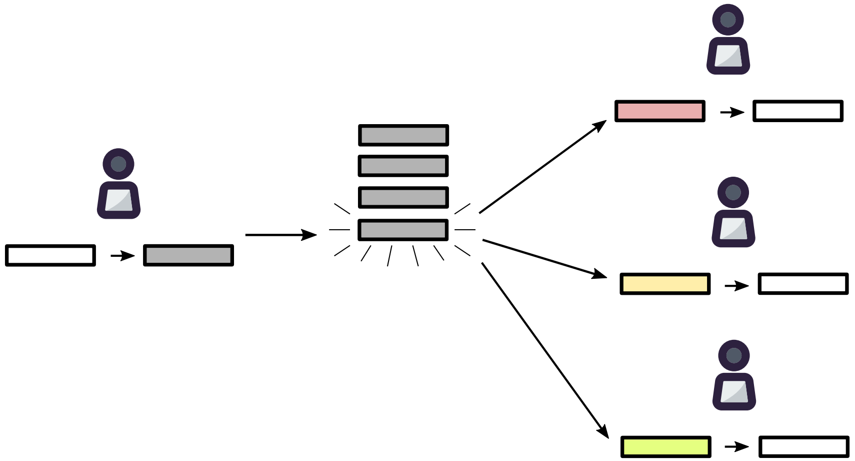
## Use Cases

### Encrypted file sharing



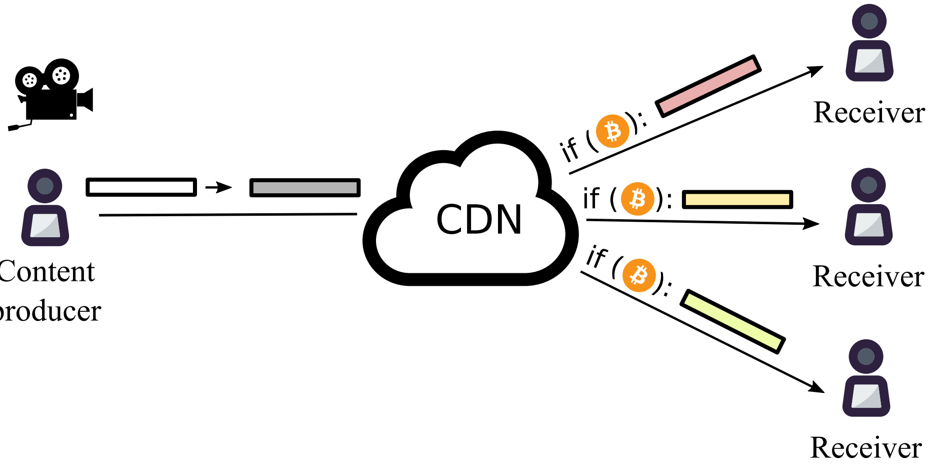
## Use Cases

### Encrypted multi-user chats



## Use Cases

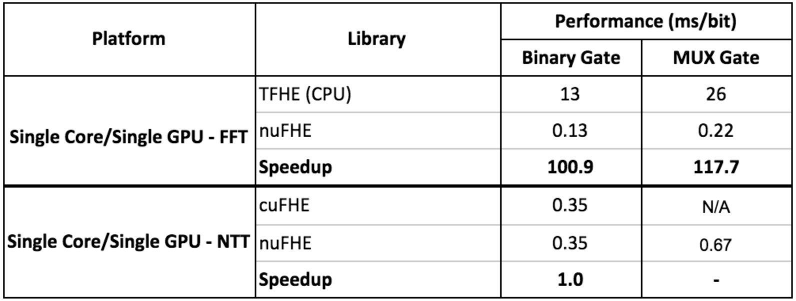
### Decentralized Netflix



# 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 |
| | **Speedup** | **100.9** | **117.7** |
| Single Core/Single GPU - NTT | cuFHE | 0.35 | N/A |
| | nuFHE | 0.35 | 0.67 |
| | **Speedup** | **1.0** | **-** |

## More Information

NuCypher

Website: `https://nucypher.com`
Github: `https://github.com/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: <email>@nucypher.com
E-mail: hello@nucypher.com