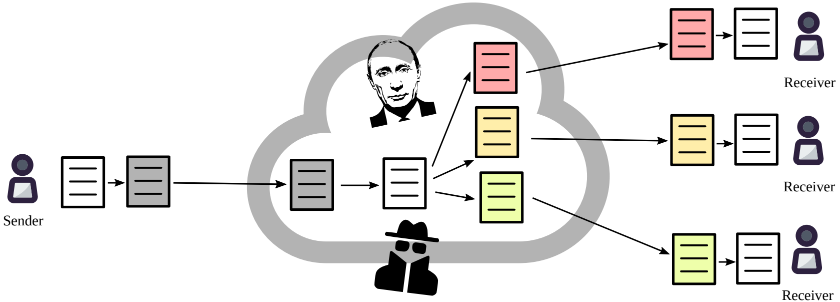
NuCypher

Michael Egorov and John Pacific

ETHBerlin, 10 Sep 2018

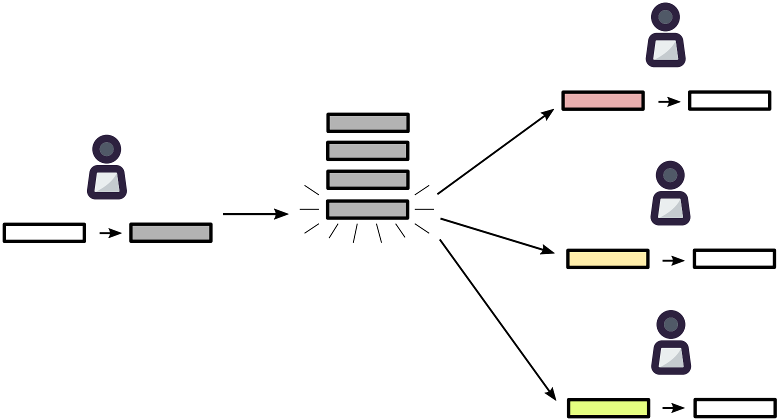
Central server + TLS

Data vulnerable to hackers, state actors etc

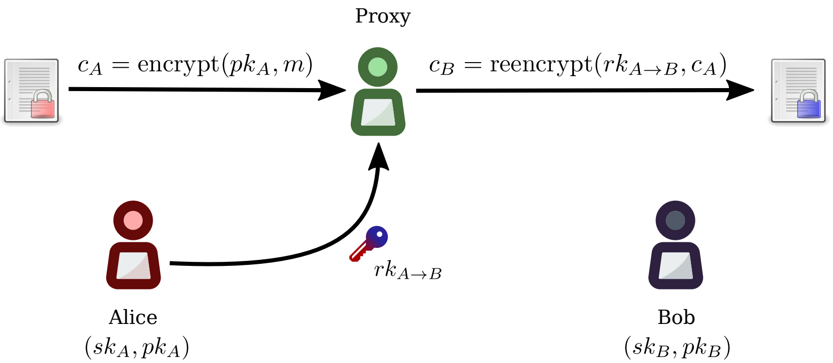


Why

Encrypted multi-user chats



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

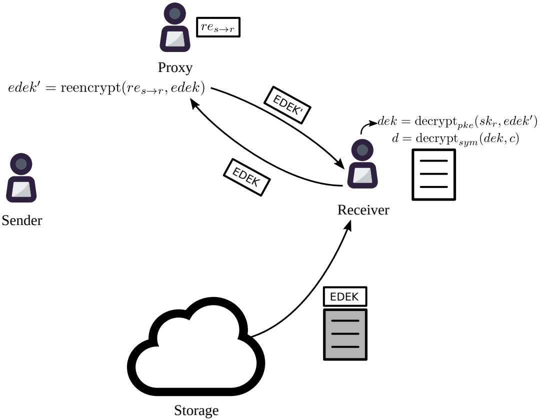
(sk_A, pk_A)

Bob

(sk_B, pk_B)

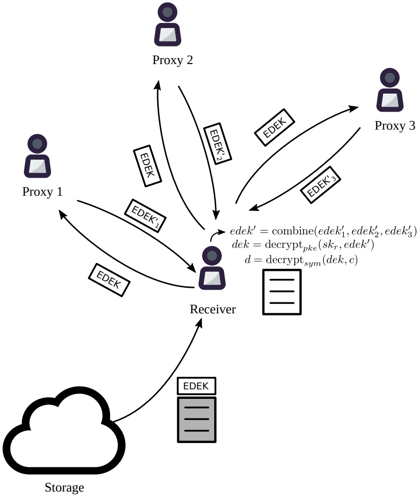
Key management using PRE

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)



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

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
  - 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

Decentralized Application

Medical & Biotech Data Platform

MEDIBLOC

MedCredits

medixain

PointNurse

wholesome
BUSINESS-IN-A-BOX FOR DOCTORS

IKU
BIOTECH REDEFINED

Key Management

EMBLEM

Automobile Data

CarBlock

Location Data

DATAEUM

Genomics Data

博士链
DOCTOR CHAIN

NuCypher Ecosystem

Protocol & Infrastructure

Medical Data Protocol

IRYO

HealthDex

Decentralized Database

bluzelle

fluence.ai

WOLK

Sharing Economy

ORIGIN

Internet of Things

SPHERITY

XAIN.

User Data Monetization

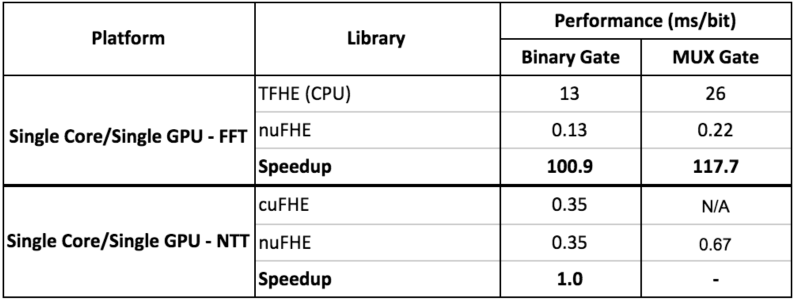
SWIPE

datum

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**	**-**

Activities + done

- Threshold m-of-n proxy re-encryption Umbral: done;
- Staking smart contracts: done;
- On-chain smart contract verification: done;
- On-chain enforcement of correctness: to do;
- NuCypher network: federated done, decentralized on the way;
- On-chain conditions for NuCypher network: to do after testnet;
- Research on FHE.

Useful links

NuCypher

Website: `https://nucypher.com`
NuCypher network: `https://github.com/nucypher/nucypher/`
PyUmbral: `https://github.com/nucypher/pyUmbral/`
GoUmbral: `https://github.com/nucypher/goUmbral/`
FHE: `https://github.com/nucypher/nufhe/`
Discord: `https://discord.gg/7rmXa3S`
Whitepaper: `https://www.nucypher.com/whitepapers/english.pdf`
E-mail: `hello@nucypher.com`

FHE demo: homomorphic smart contracts assembly

https://github.com/nucypher/Sputnik/