

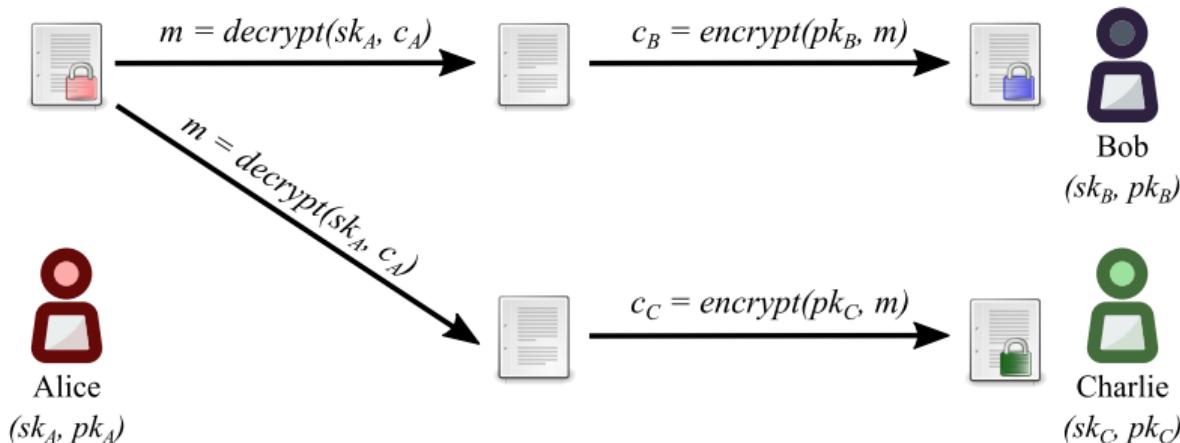


NuCypher

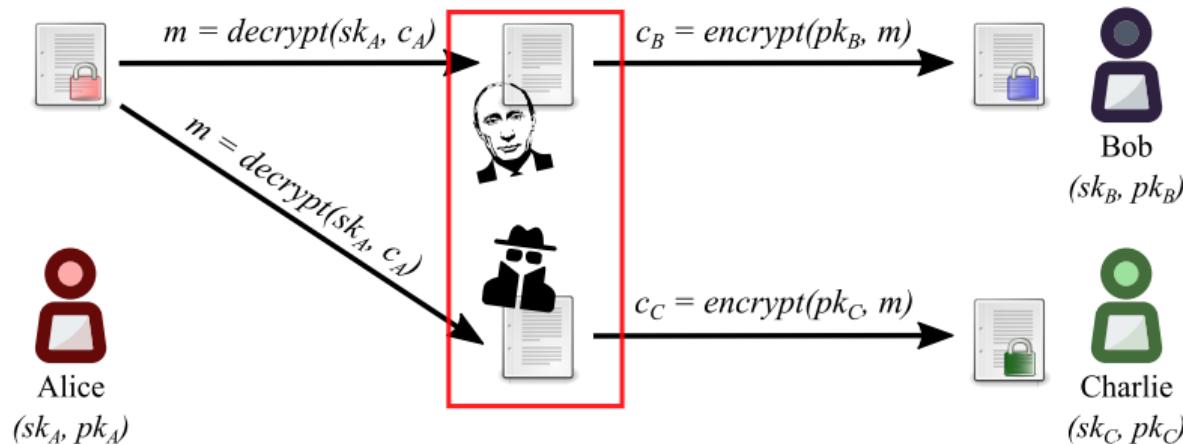
Michael Egorov, CTO

ETHSingapore, 2018, 7 Dec 2018

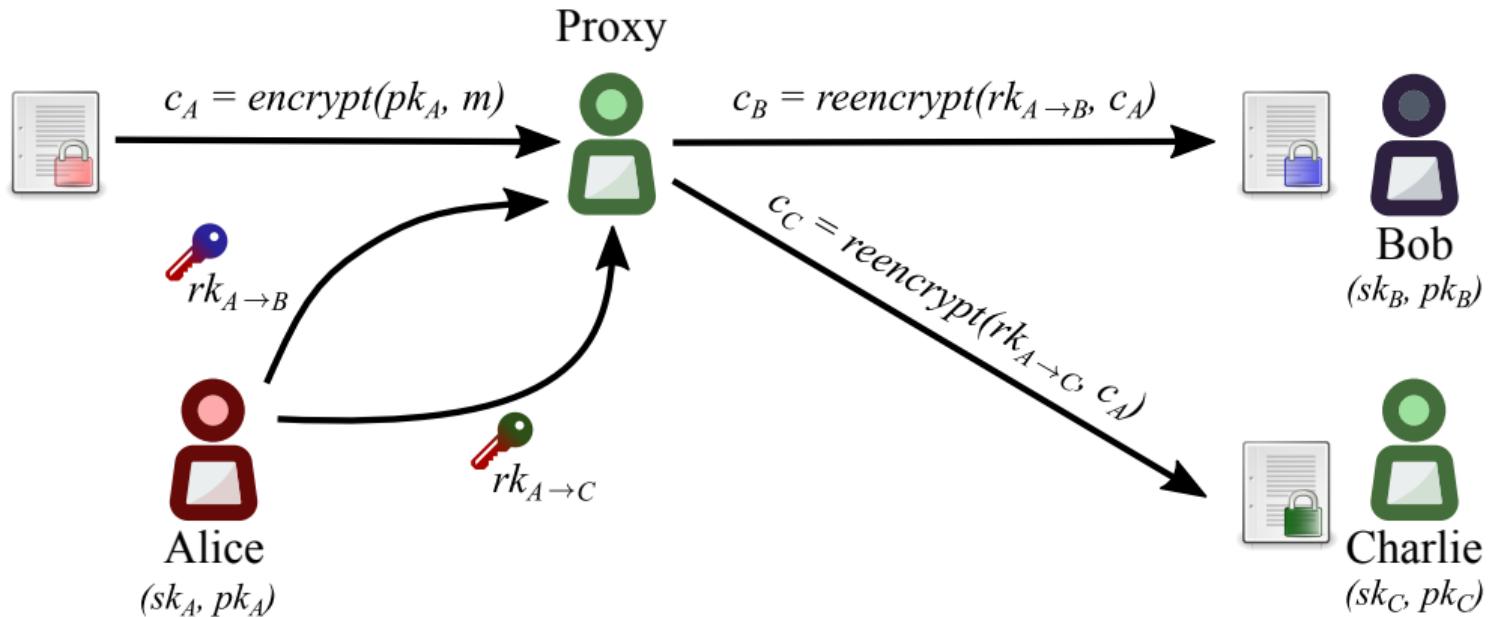
Public Key Encryption (PKE)



Public Key Encryption (PKE)



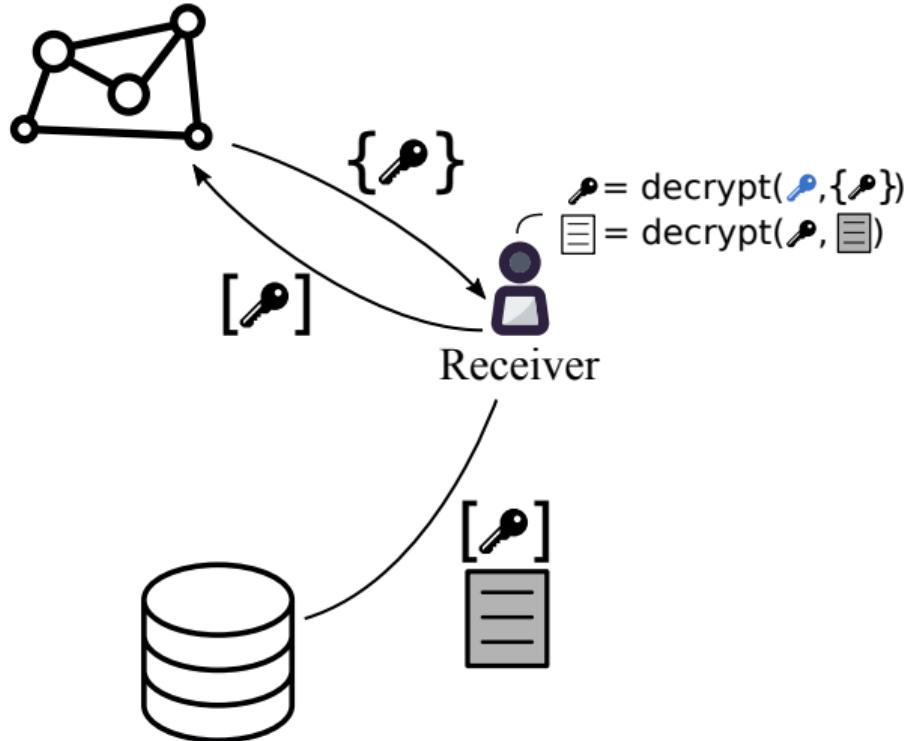
What is proxy re-encryption (PRE)



Proxy re-encryption video

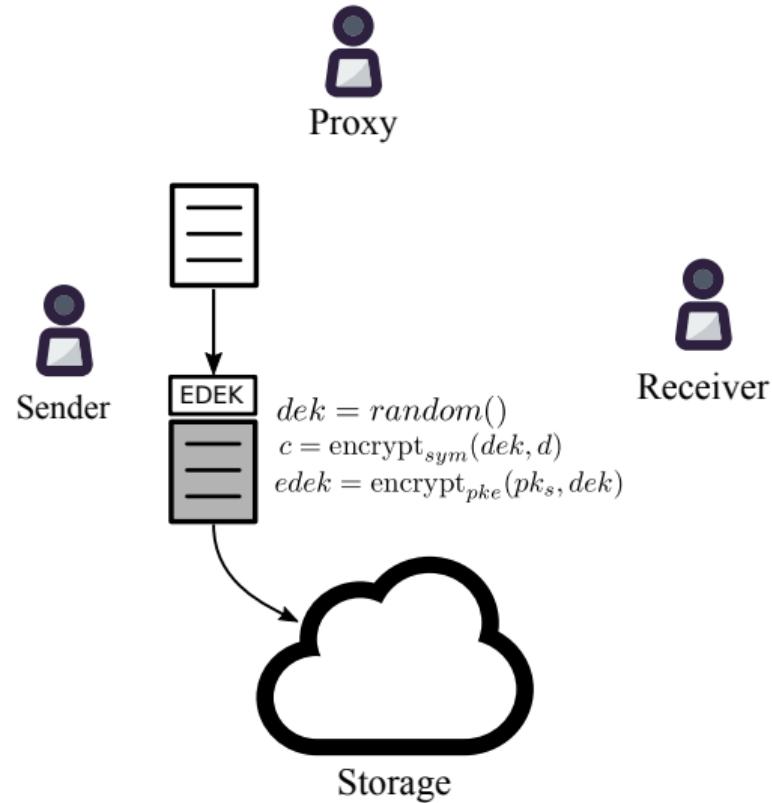
Solution

Proxy re-encryption + decentralization



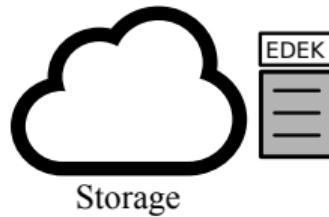
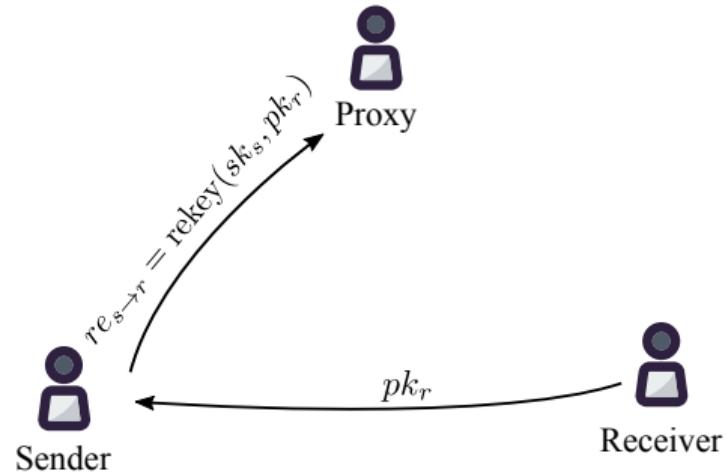
Centralized KMS using PRE

Encryption



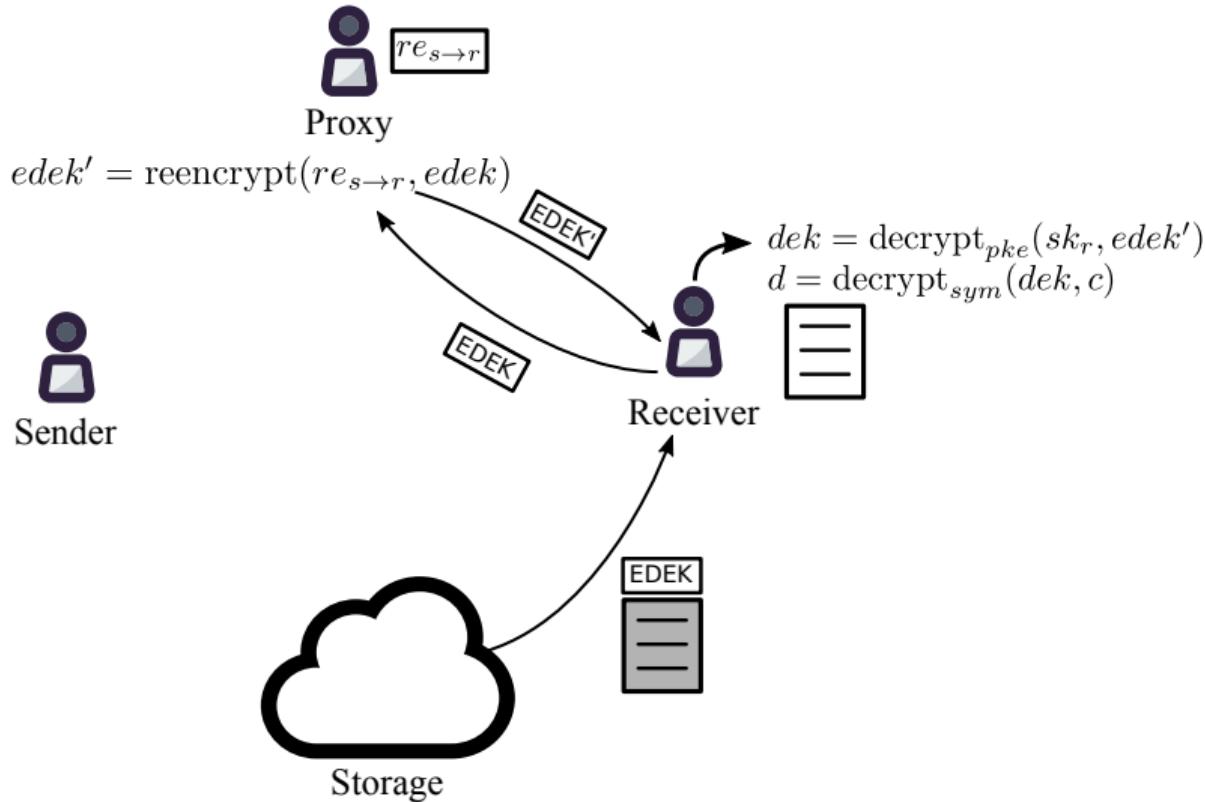
Centralized KMS using PRE

Access delegation



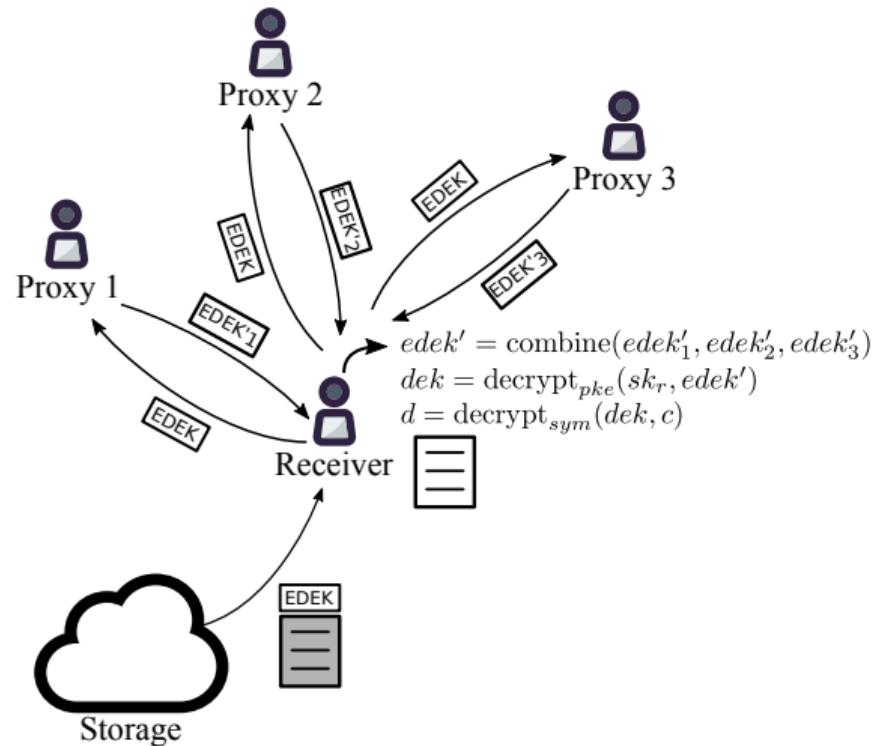
Centralized KMS using PRE

Decryption



Decentralized Key Management

Using threshold split-key re-encryption (Umbral)



Umbral: Threshold Proxy Re-encryption

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

NU Token

Purpose

- Splitting trust across re-encryption nodes
 - ▶ More tokens = more trust, more work, and more compensation
- Proof of Stake for minting new coins according to the mining schedule
- Security deposit at stake against malicious behavior of nodes

Where to start

```
virtualenv _venv -p python3
source _venv/bin/activate
pip3 install git+https://github.com/nucypher/nucypher.git@federat

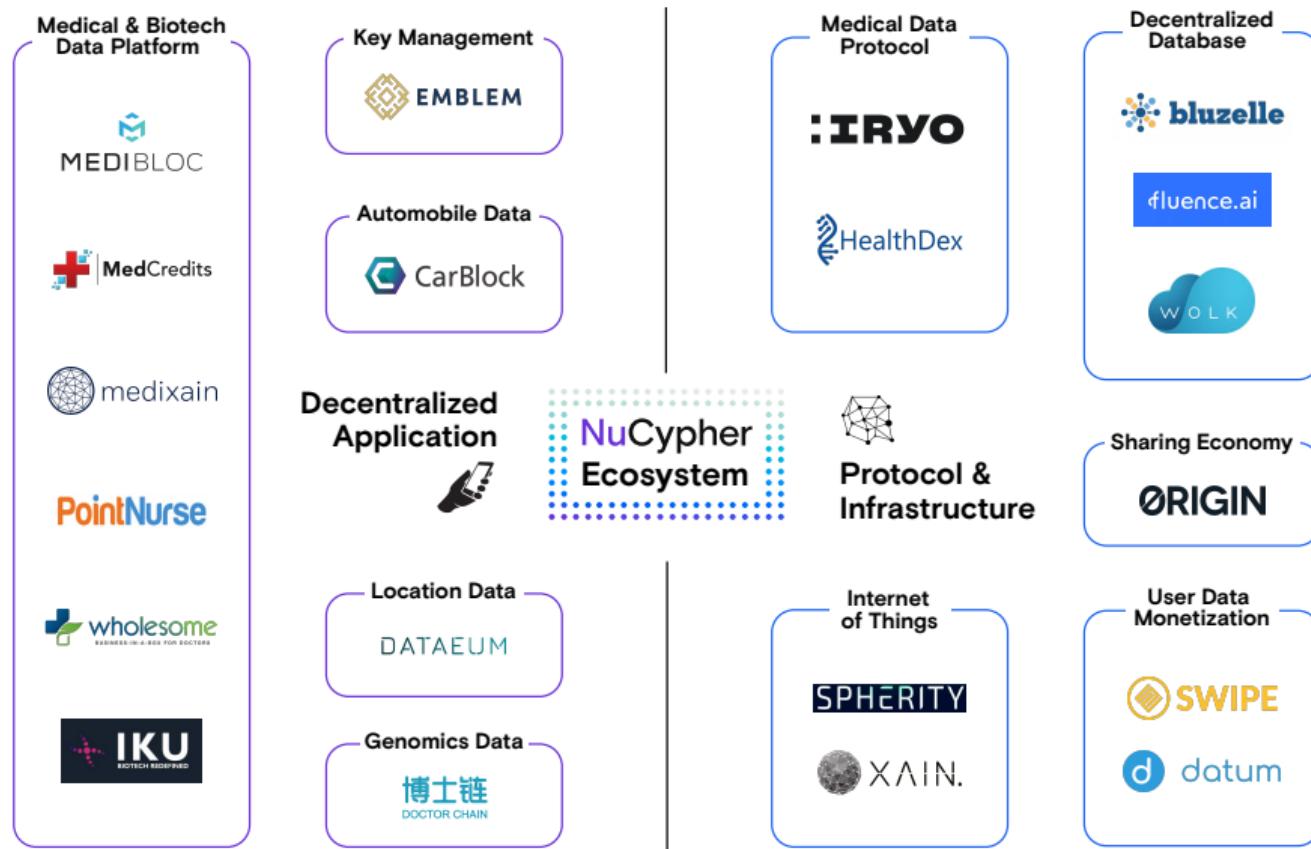
git clone https://github.com/nucypher/nucypher.git
cd nucypher
git checkout federated
cd examples/finnegans_wake_demo
python3 finnegans-wake-concise-demo.py
```

Demo with testnet



Extra: Video of testnet node running

Early Users



Competing Technology

Data Masking and Tokenization

- Less secure for data with underlying patterns
- Reduce the value of data by obfuscating it

Public Key Encryption

- Data must be decrypted before it is shared
- Not Scalable

Multi-Party Computation

- Interactive protocol
- Slow Performance

Fully Homomorphic Encryption

- Slow Performance
 - ▶ NuCypher has developed a GPU-accelerated FHE library: nuFHE

Fully Homomorphic Encryption

nuFHE library

- GitHub: <https://github.com/nucypher/nufhe>
- GPU implementation of fully homomorphic encryption
- Uses either FFT or integer NTT
- 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	-

API prize (2500 USD)

Anything of the following:

- Wrapper to interact with NuCypher network from Go, node.js, ...;
- Extension to interact with NuCypher network in browsers;
- Any exceptional dApp with good use of NuCypher for permission management.

More Information



NuCypher

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

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

Proxy Re-encryption Network: <https://github.com/nucypher/nucypher>

Umbral Reference Implementation: <https://github.com/nucypher/pyUmbral>

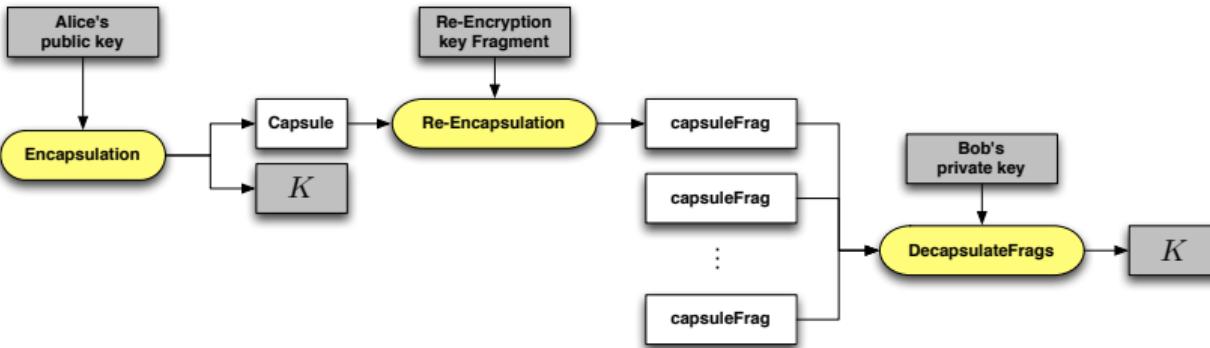
nuFHE: <https://github.com/nucypher/nufhe>

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

E-mail: michael@nucypher.com

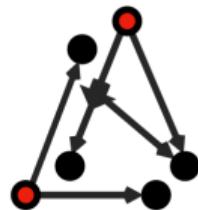
E-mail: hello@nucypher.com

Appendix: Umbral Flow Diagram



- Reference implementation: <https://github.com/nucypher/pyUmbra>
 - Documentation: <https://github.com/nucypher/umbra-doc>

Appendix: Security Audits



Least Authority
Freedom Matters

Appendix: NU Token Metrics

Mining

Mining & Staking Economics: <https://github.com/nucypher/mining-paper>

Mining reward:

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

$$T_{i,\text{initial}} \geq T_{\min}$$

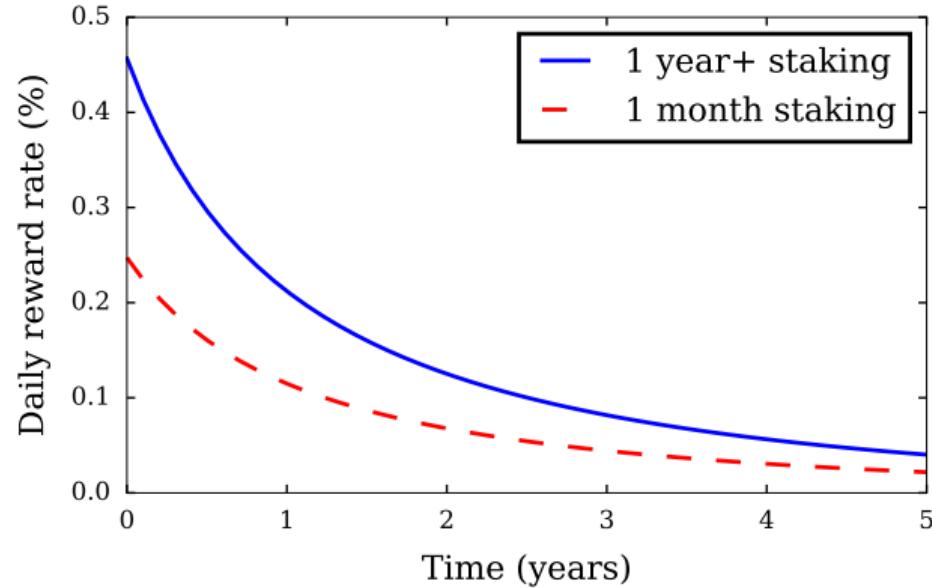
$$\delta S_{i,t} = \kappa \frac{l_i}{\sum l_j} \frac{\ln 2}{T_{1/2}} (S_{\max} - S_{t-1})$$

Results into:

$$\text{reward} \propto 2^{\frac{t}{T_{1/2}}}$$

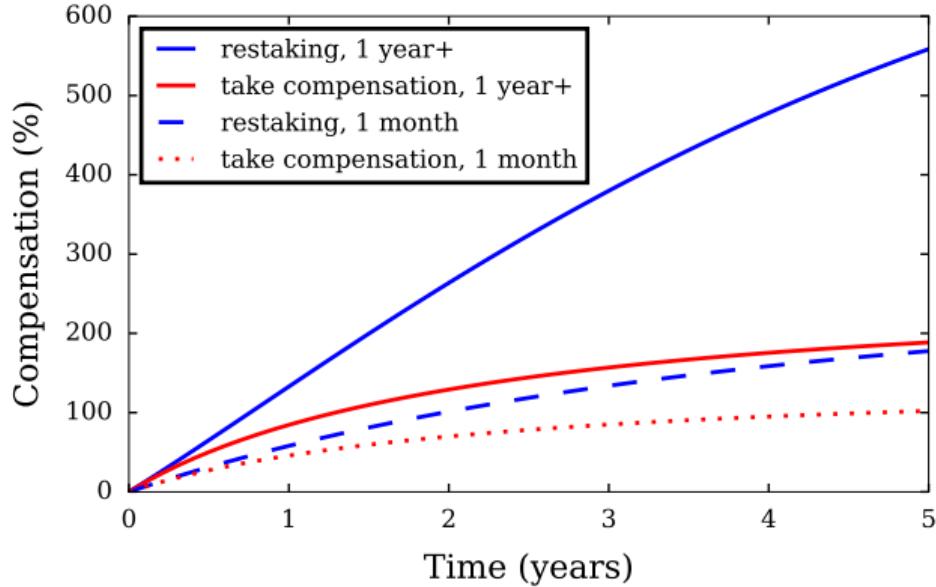
Appendix: NU Token Metrics

Graph of daily mining compensation



Appendix: NU Token Metrics

Relocking mining rewards



Appendix: Team

Founders



MacLane Wilkison
Co-founder and CEO



Michael Egorov, PhD
Co-founder and CTO

Advisors



Prof. Dave Evans



Prof. Giuseppe Ateniese
Stevens Inst. of Technology



John Bantleman
Rainstor



Tony Bishop
Equinix

Employees



David Nuñez, PhD
Cryptographer



John Pacific (tux)
Engineer



Justin Myles Holmes
Engineer



Sergey Zotov
Engineer



Kieran Prasch
Engineer



Bogdan Opanchuk, PhD
Engineer



Ryan Caruso
Community



Derek Pierre
Business Developer



Arjun Hassard
Product & Partnerships



Keaton Bruce
Engineer



Eva Evergreen
Engineer