NuCypher

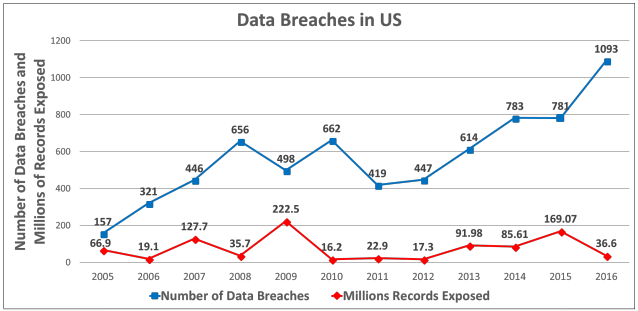
Derek Pierre
Business Development Lead

NuCypher Overview

- Use cryptography to build the tools & infrastructure to preserve data privacy
- Privacy-preserving solutions for distributed applications
  - Proxy Re-encryption (PRE)
    - Secure data-sharing and access control of encrypted data
  - Fully Homomorphic Encryption (FHE)
    - Perform arbitrary operations on encrypted data
- Blockchain & Private Deployments

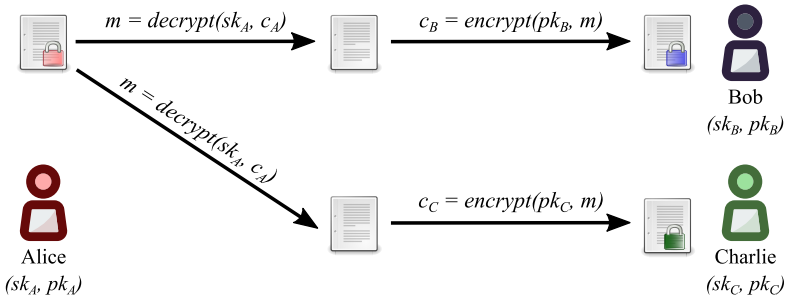
Problem

Data Breaches



Source:
https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed/

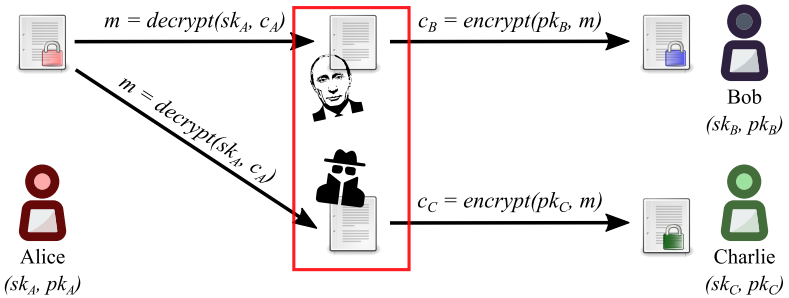
Public Key Encryption (PKE)



Limitations

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

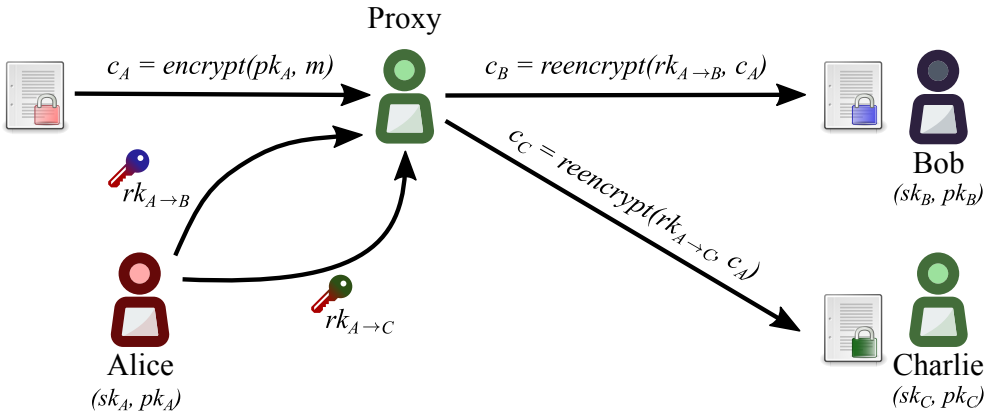
Public Key Encryption (PKE)



Limitations

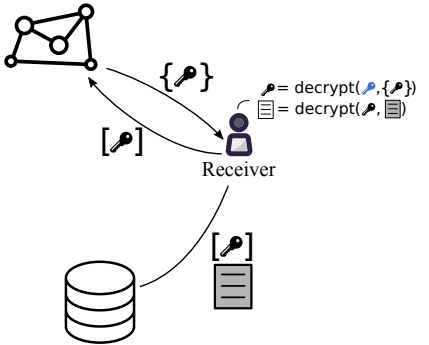
- Decryption required before sharing
- Not scalable
- Complex access revocation

What is proxy re-encryption (PRE)



Solution

Proxy Re-encryption + Key Management

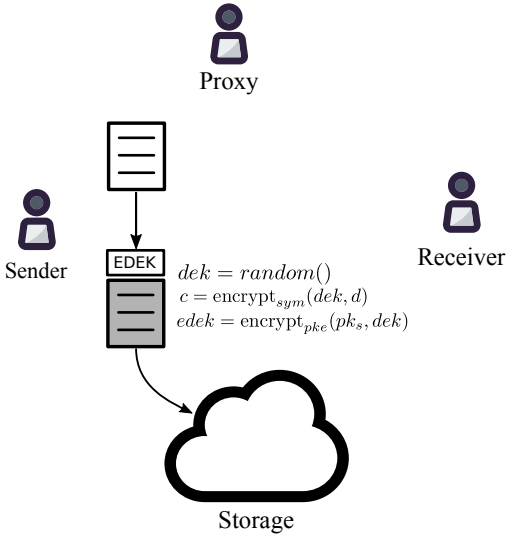


Advantages

- Data not decrypted to facilitate sharing
- Scalable and performant
- Access revocation through re-encryption key deletion
- Secure use of data storage providers

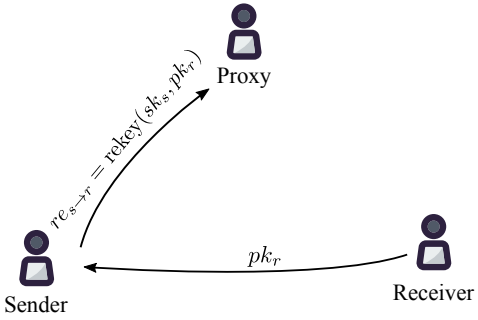
Centralized KMS using PRE

Encryption



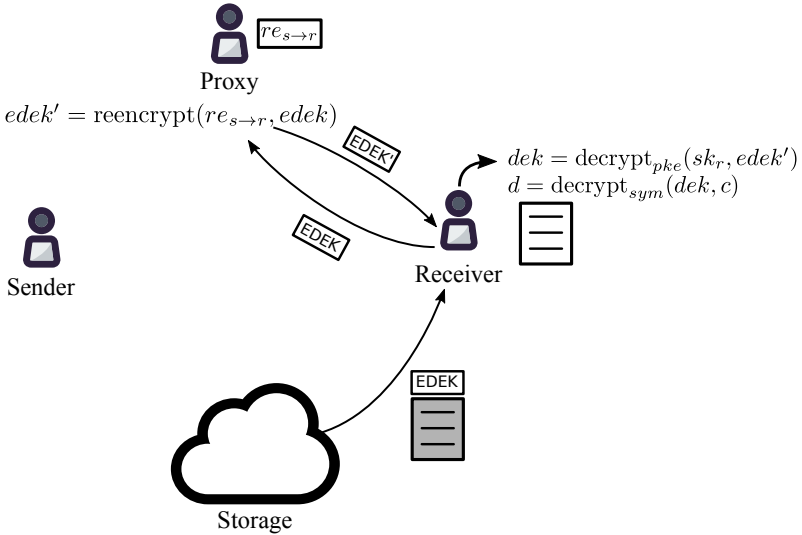
Centralized KMS using PRE

Access delegation



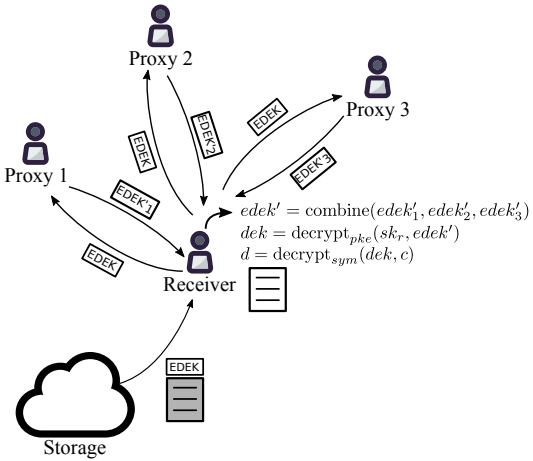
Centralized KMS using PRE

Decryption



Decentralized KMS using PRE

Using threshold split-key re-encryption (Umbral)



NuCypher PRE Properties

- Unidirectional
- Single hop
- Non-interactive

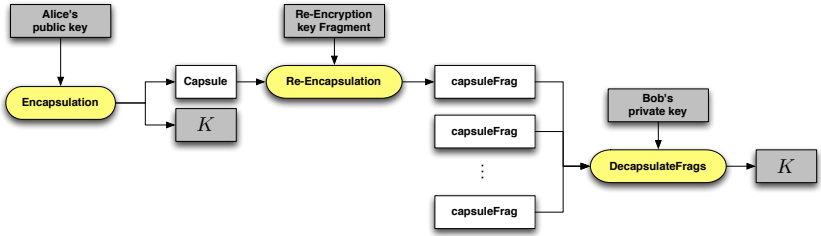
KEM/DEM Approach

- Umbral KEM for threshold re-encryption
- ECIES for key encapsulation
- DEM can be any AE (ChaCha20-Poly1305)

Verification of Correctness

- Verification through non-interactive ZK proof
- Incentive layer via NU staking token
- Re-encryption validated by challenge protocol

Umbral Flow Diagram



- Documentation: `https://github.com/nucypher/umbral-doc`
- Reference implementation: `https://github.com/nucypher/pyUmbral`

Decentralized KMS: 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

Data Sharing Policies

- Time-based
- Conditional on payment
  - “Grant access once paid, continue granting while paying”
- Smart contract (public) method

Decentralized re-encryption nodes (Ursulas) relied on to apply conditions without having the ability to decrypt data

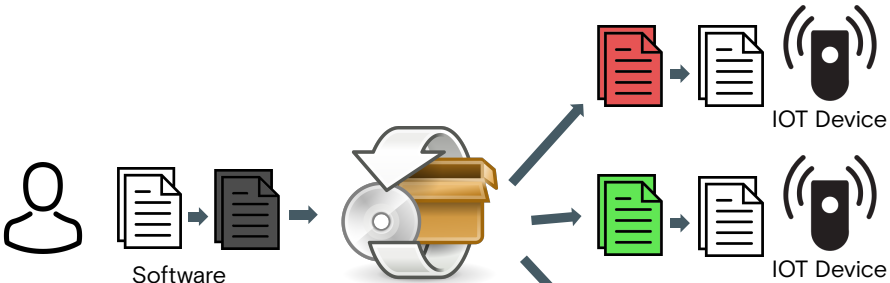
Blockchain & Smart Contract Agnostic

Public Blockchains

Private/Permissioned Blockchains

Use Cases

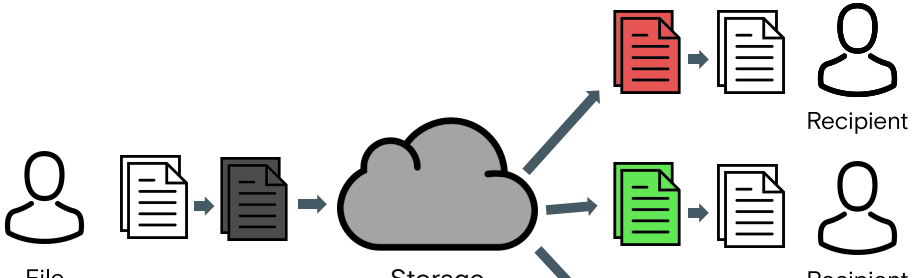
Scalable, Secure IOT Updates



1. Encrypt → **2. Re-Encrypt** → **3. Decrypt**

Use Cases

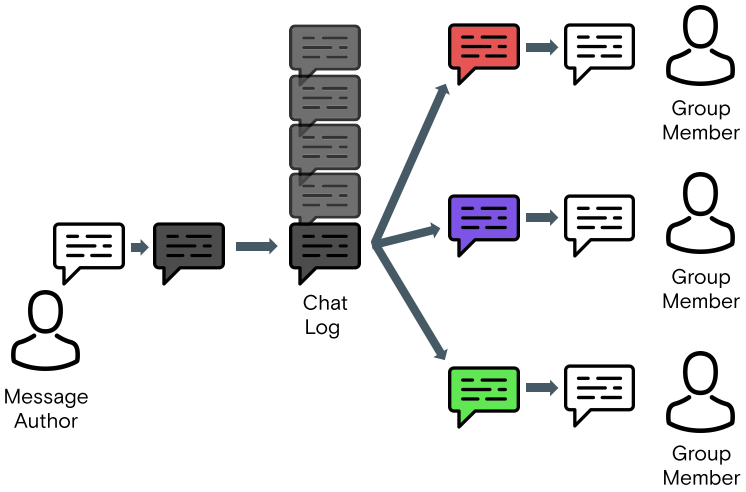
Encrypted file sharing



1. Encrypt → **2. Re-Encrypt** → **3. Decrypt**

Use Cases

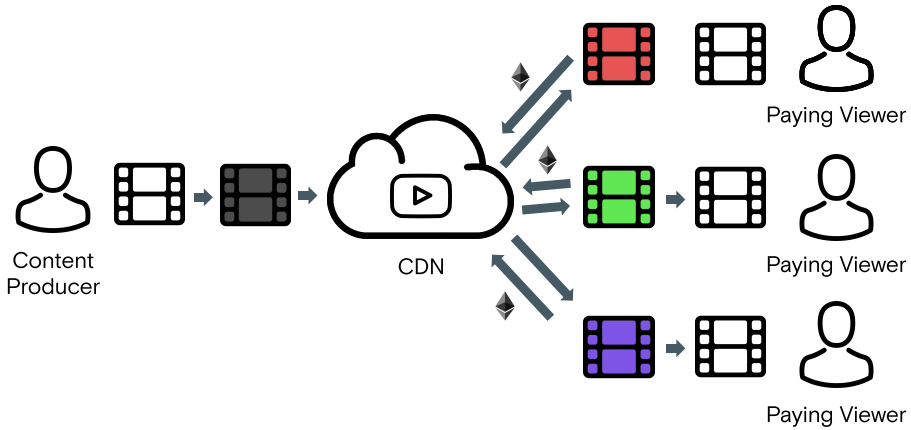
Encrypted multi-user chats



1. Encrypt → **2. Re-Encrypt** → **3. Decrypt**

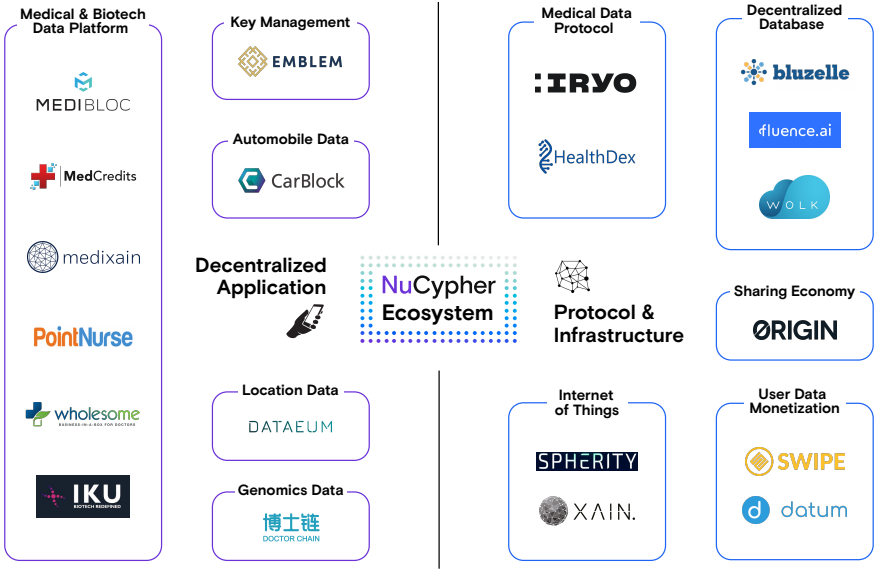
Use Cases

Decentralized access-controlled content



1. Encrypt → **2. Conditional Re-Encrypt** → **3. Decrypt**

Early Users


Medical & Biotech Data Platform
MEDIBLOC
MedCredits
medixain
PointNurse
wholesome
IKU
Key Management
EMBLEM
Automobile Data
CarBlock
Location Data
DATAEUM
Genomics Data
博士链
DOCTOR CHAIN
Decentralized Application
NuCypher Ecosystem
Protocol & Infrastructure
Medical Data Protocol
IRYO
HealthDex
Internet of Things
SPHERITY
XAIN.
Decentralized Database
bluzelle
fluence.ai
WOLK
Sharing Economy
ORIGIN
User Data Monetization
SWIPE
datum

Umbral PRE Demo

- **Documentation:** `https://github.com/nucypher/umbral-doc`
- **Reference implementation:** `https://github.com/nucypher/pyUmbral`

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**	**-**

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: derek@nucypher.com
E-mail: hello@nucypher.com

Appendix: 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`

Appendix: 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

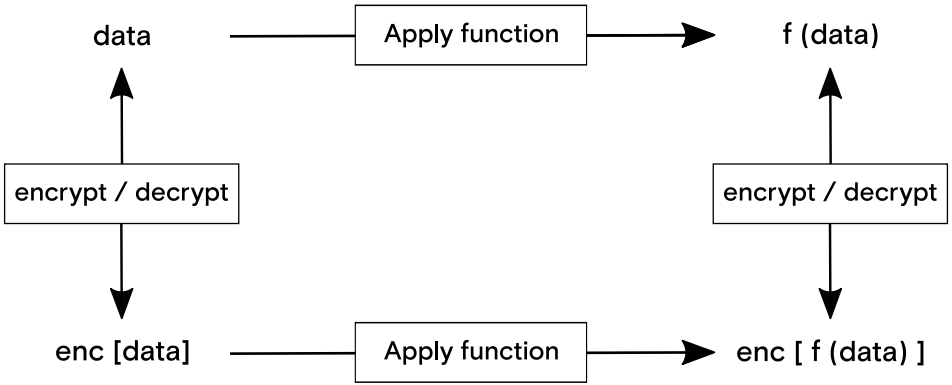
Multi-Party Computation

- Interactive protocol
- Slow Performance

Fully Homomorphic Encryption

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

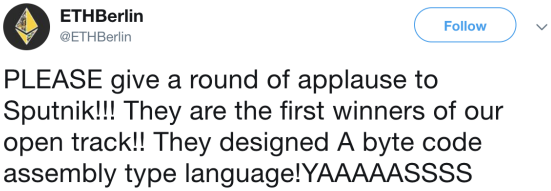
Appendix: Fully Homomorphic Encryption



Appendix: FHE Proof of Concept

Sputnik

- GitHub: `https://github.com/nucypher/sputnik`
- Assembly language and interpreter for FHE that uses nuFHE
- Commits a merkle root of computation to the blockchain for proof of logic flow
- Used to execute first homomorphic smart contract at ETHBerlin 2018



Appendix: Team

Founders

MacLane Wilkison
Co-founder and CEO

Michael Egorov, PhD
Co-founder and CTO

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 Developmer

Arjun Hassard
Product & Partnerships

Keaton Bruce
Engineer

Eva Evergreene
Engineer

Advisors

Prof. Dave Evans

Prof. Giuseppe Ateniese
Stevens Inst. of Technology

John Bantleman
Rainstor

Tony Bishop
Equinix