



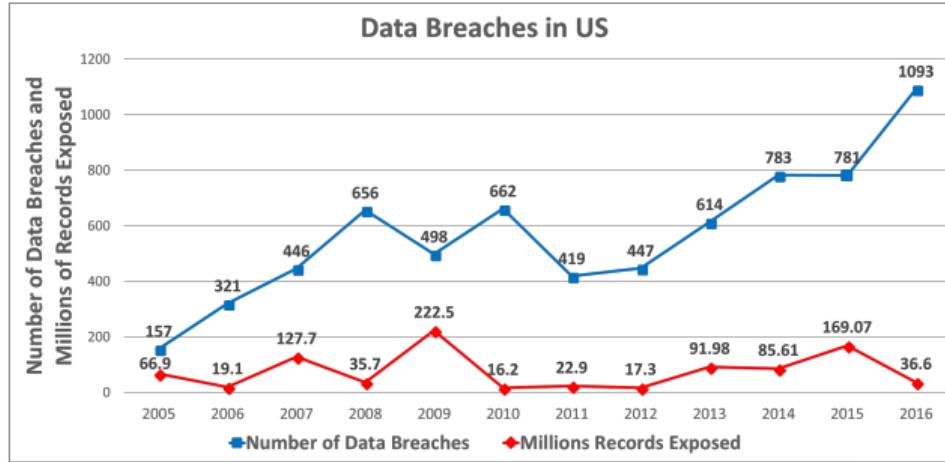
NuCypher

<fname lname>, <title>

<event>, <dd Mon yyyy>

Problem

Data Breaches

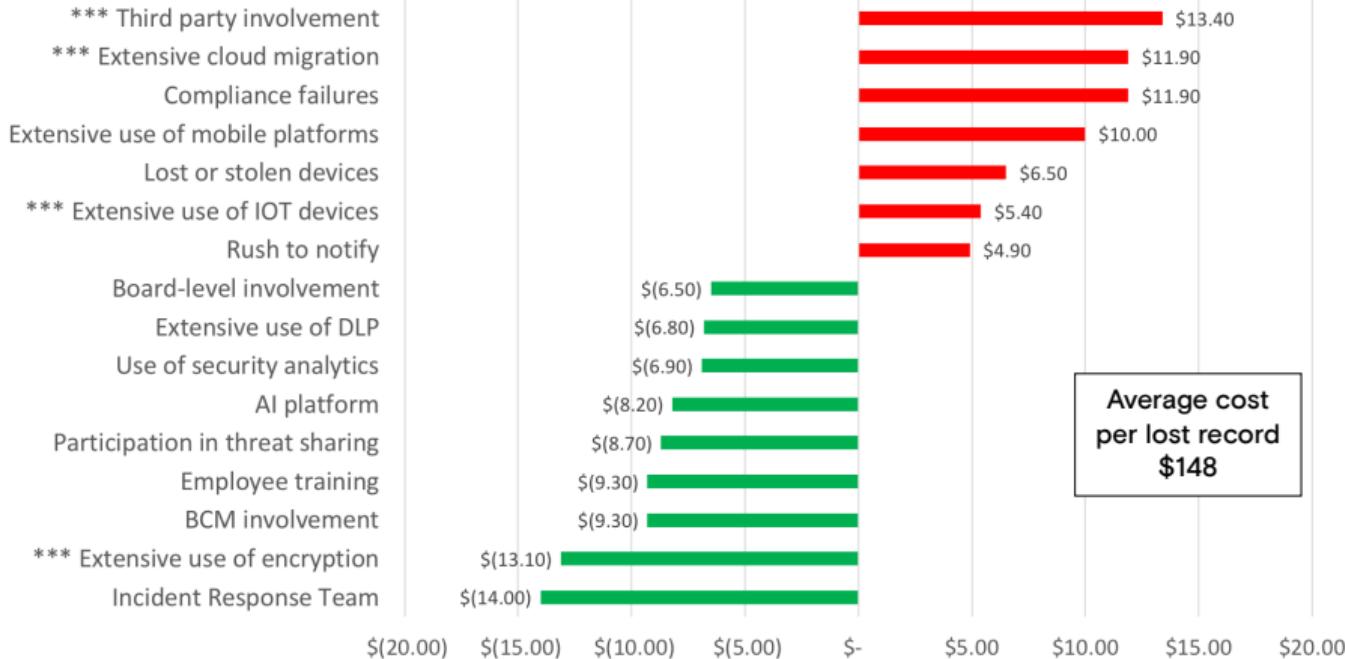


Source:

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

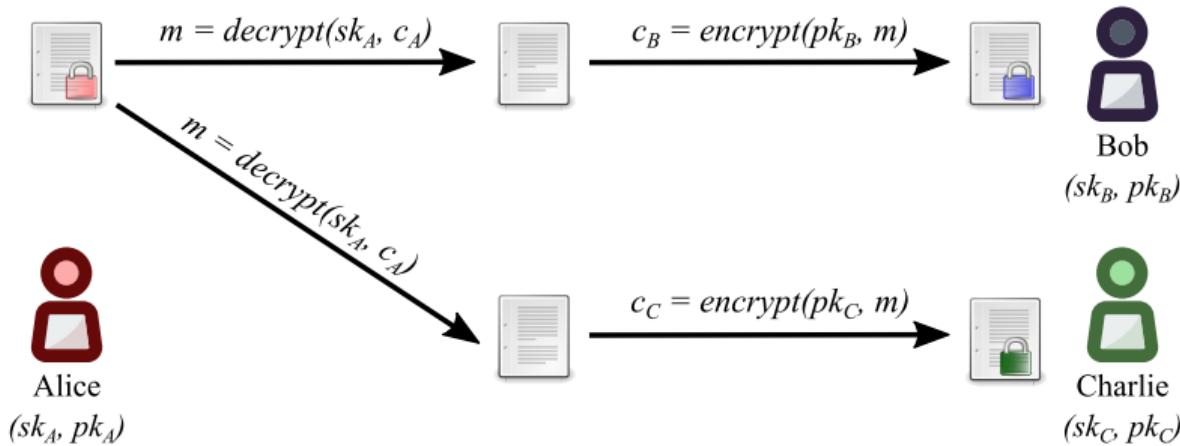
Impact of Data Breaches

Impact on Per Lost Record Cost (US\$)



Source: IBM 2018 Cost of a Data Breach Study: Global Overview, <https://www.ibm.com/security/data-breach>

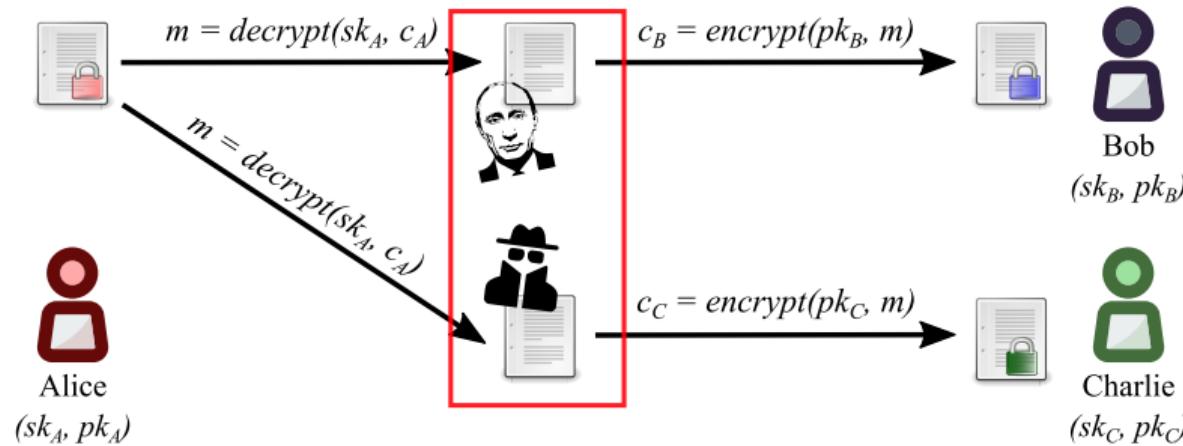
Public Key Encryption (PKE)



Limitations

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

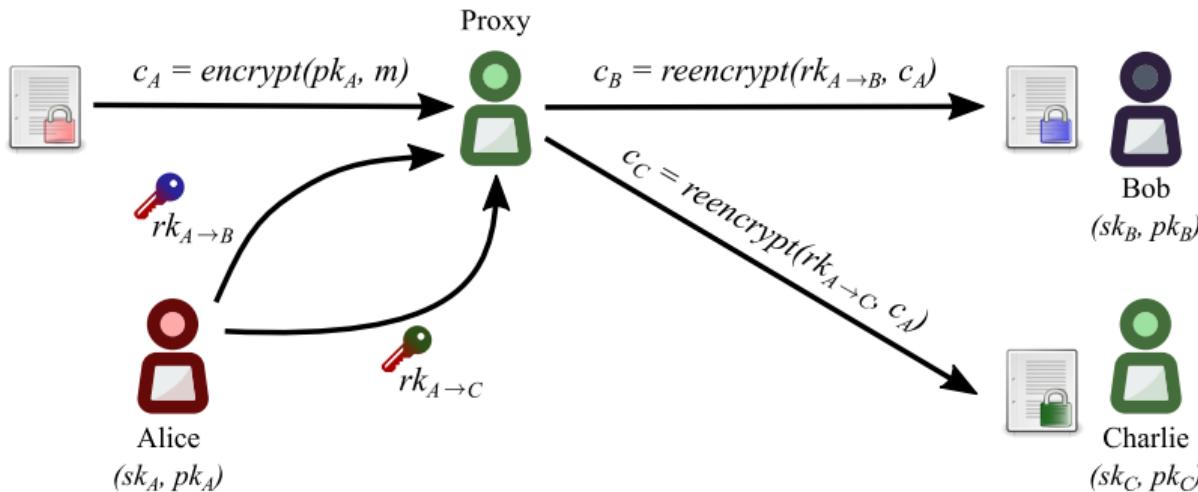
Public Key Encryption (PKE)



Limitations

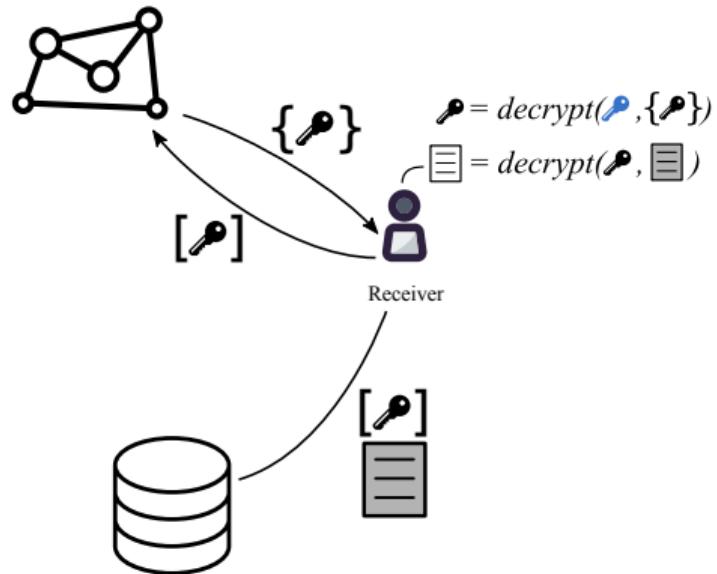
- Decryption required before sharing
- Not scalable
- Complex access revocation

What is proxy re-encryption (PRE)



Solution

Proxy Re-encryption + KMS

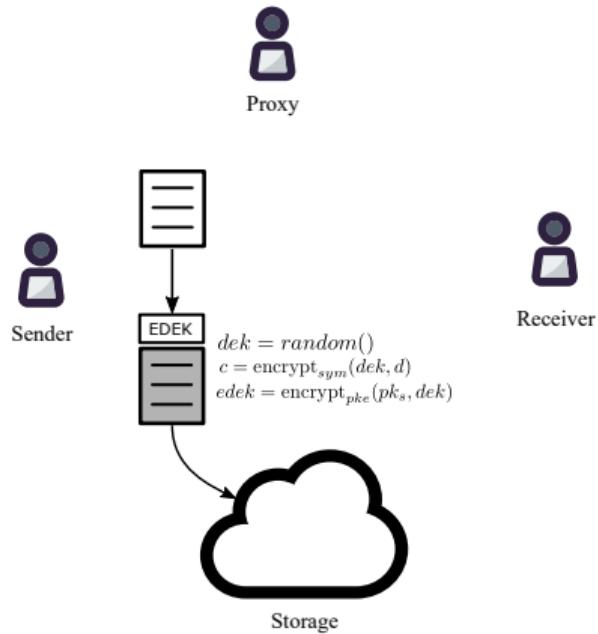


Advantages

- Data not decrypted to facilitate sharing
- Scalable and performant
- Access revocation through re-encryption key deletion

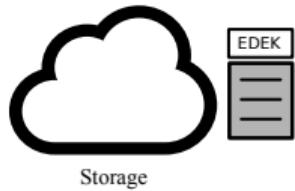
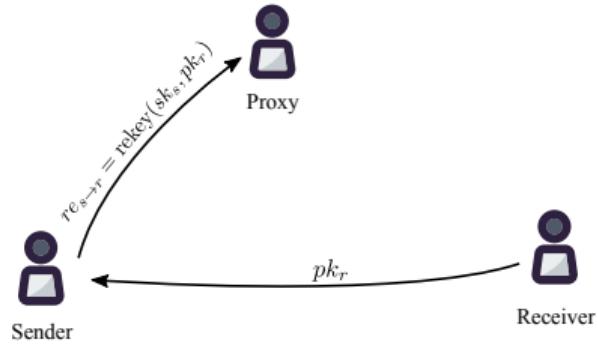
Centralized KMS using PRE

Encryption



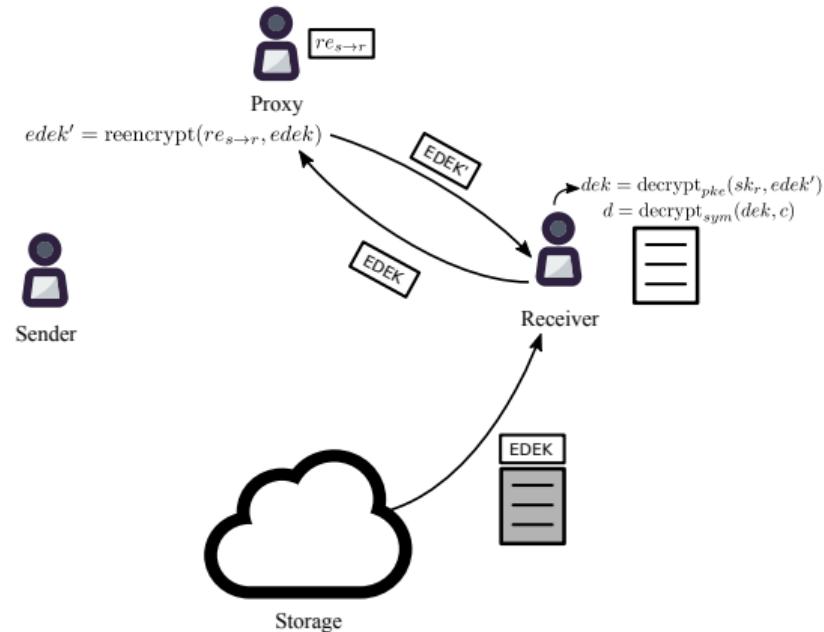
Centralized KMS using PRE

Access delegation



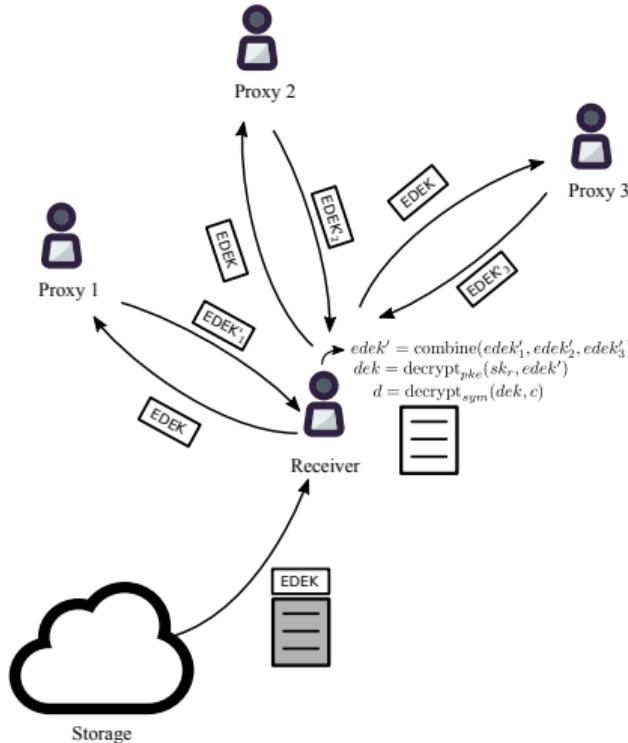
Centralized KMS using PRE

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

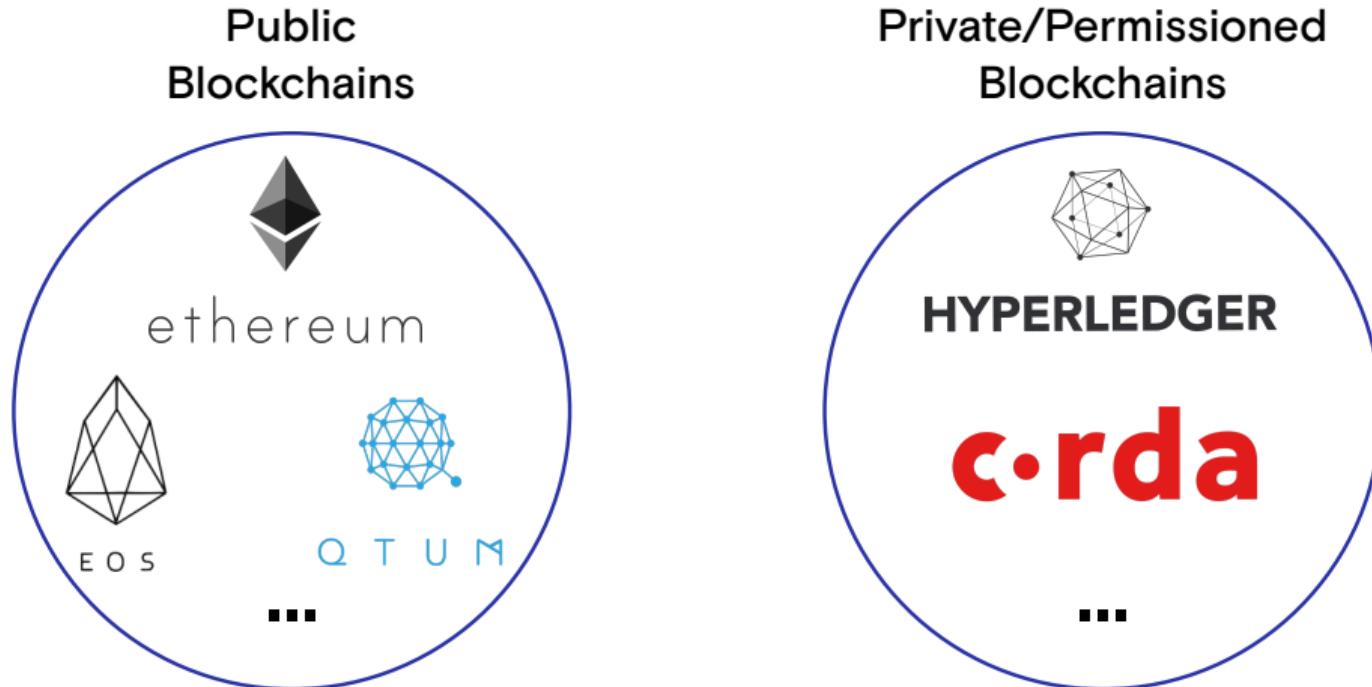
- Verification through non-interactive ZK-proof
- Incentive layer via NU staking token
- Re-encryption validated by challenge protocol

Decentralized KMS: Token

Purpose

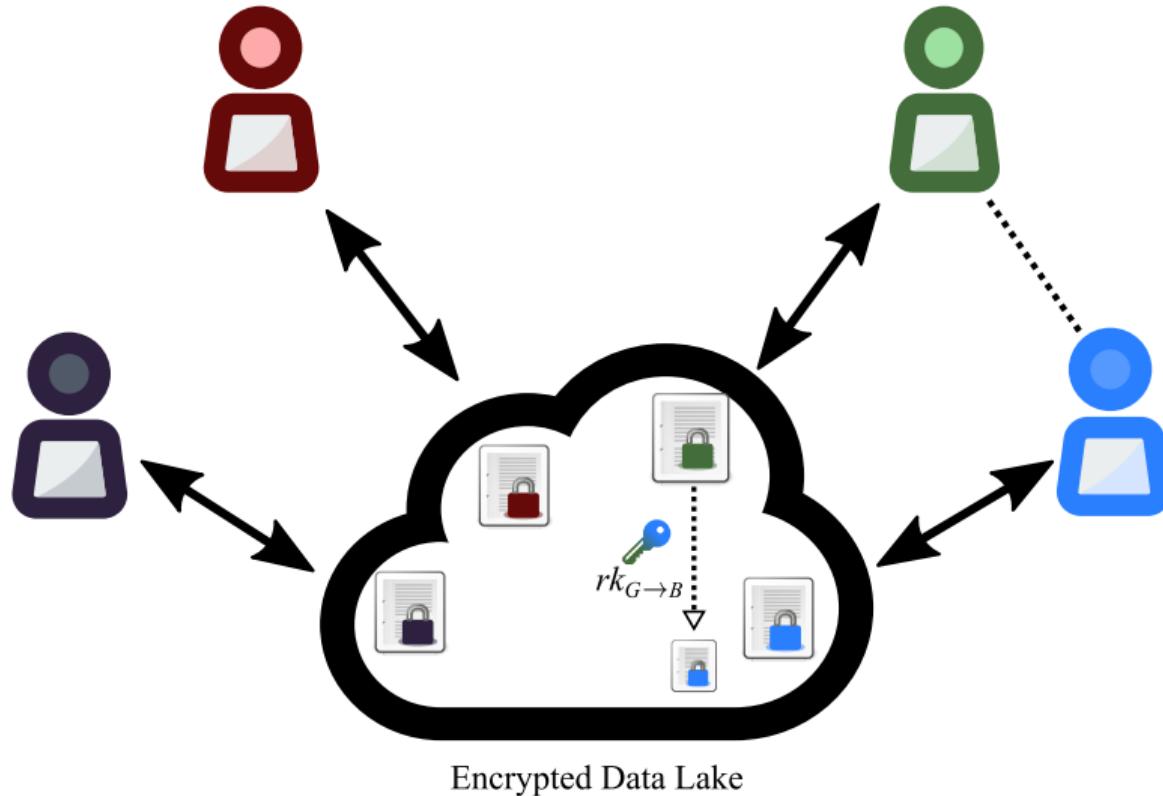
- Splitting trust between re-encryption nodes (more tokens = more trust and more work)
- Proof of Stake for minting new coins according to the mining schedule
- Security deposit to be at stake against malicious behavior of nodes

Blockchain & Smart Contract Agnostic



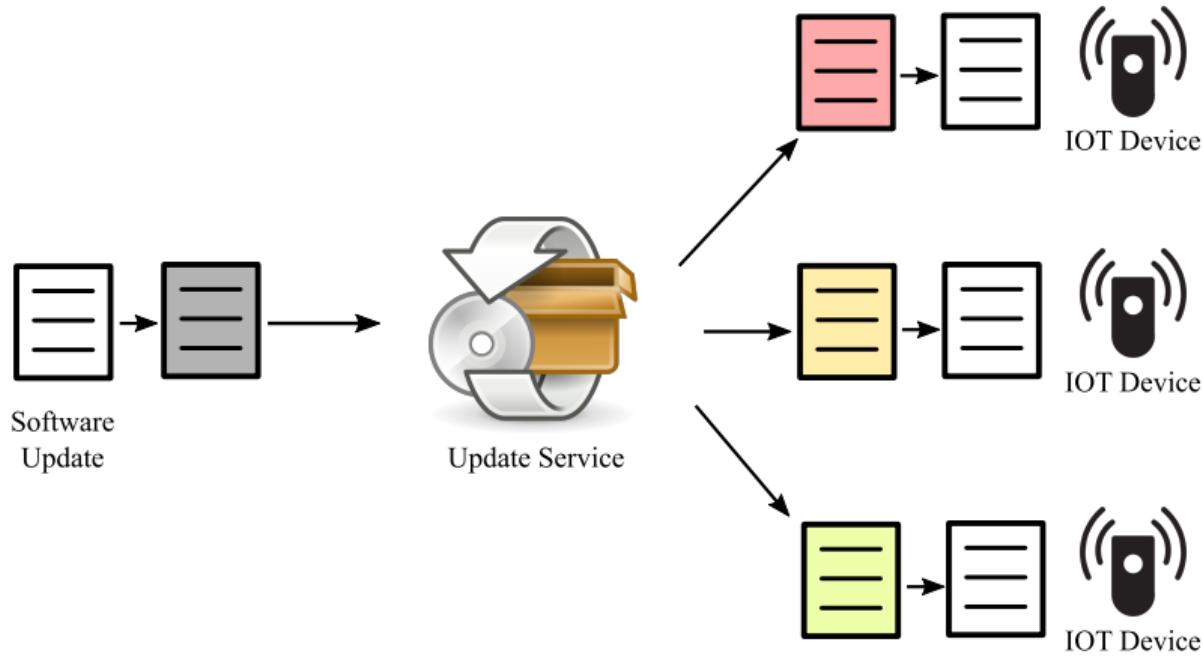
Use Cases

Multi-tenant, Multi-source Encrypted Data Lake



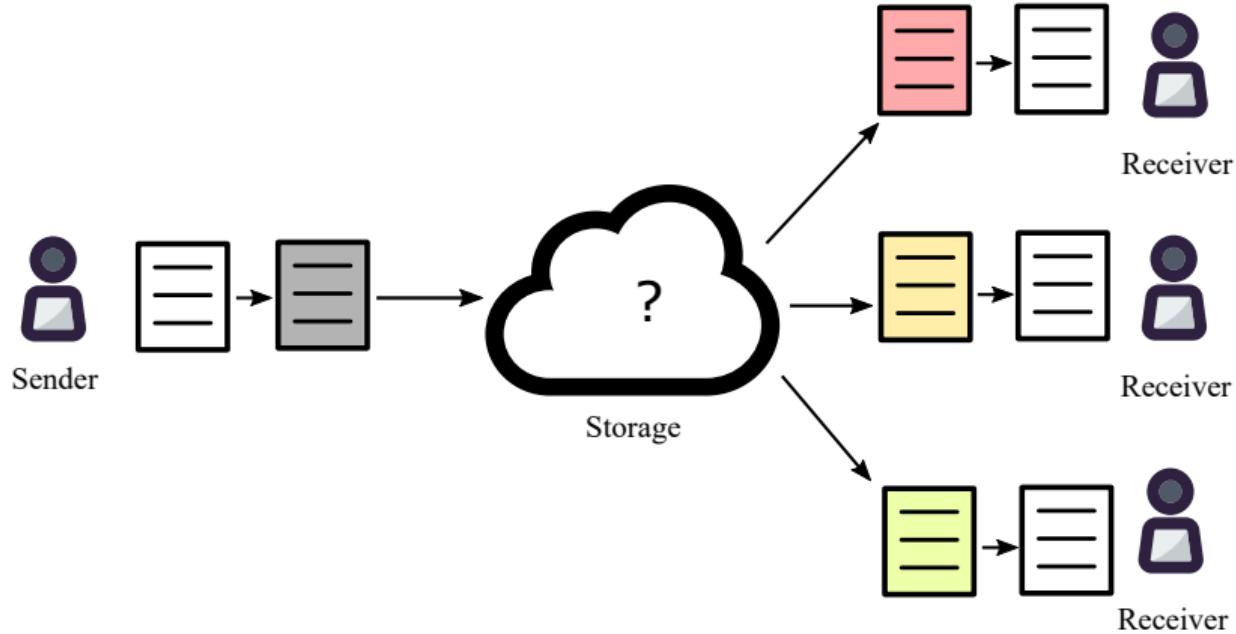
Use Cases

Scalable, Secure IOT Updates



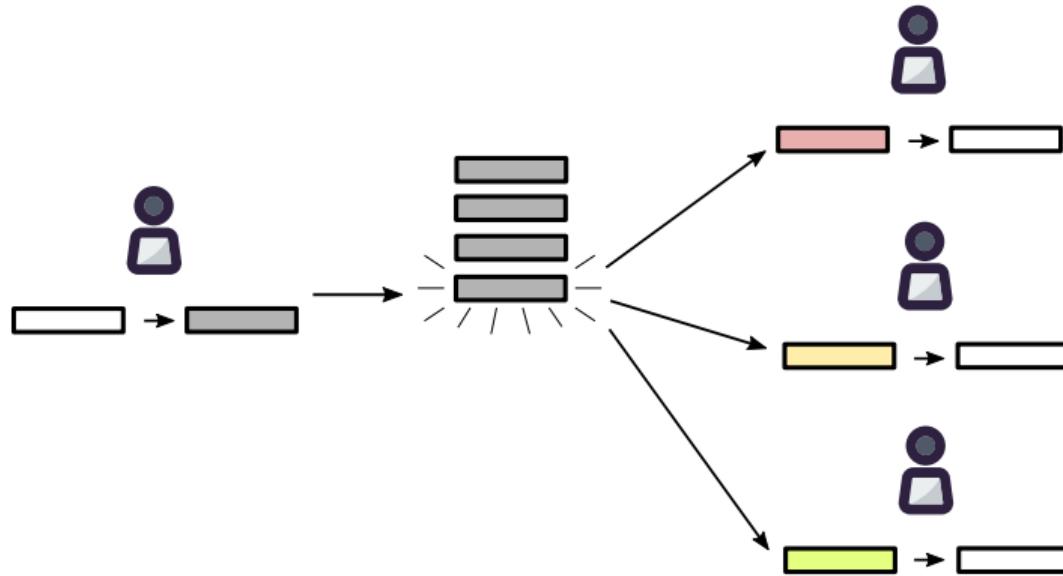
Use Cases

Encrypted file sharing



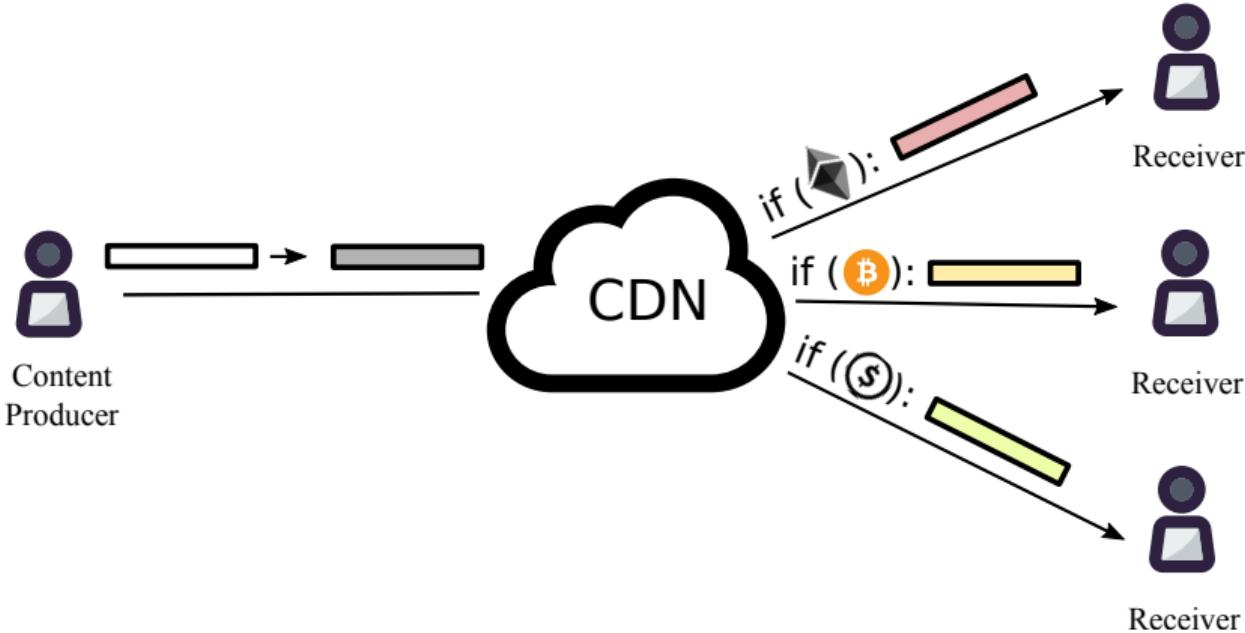
Use Cases

Encrypted multi-user chats

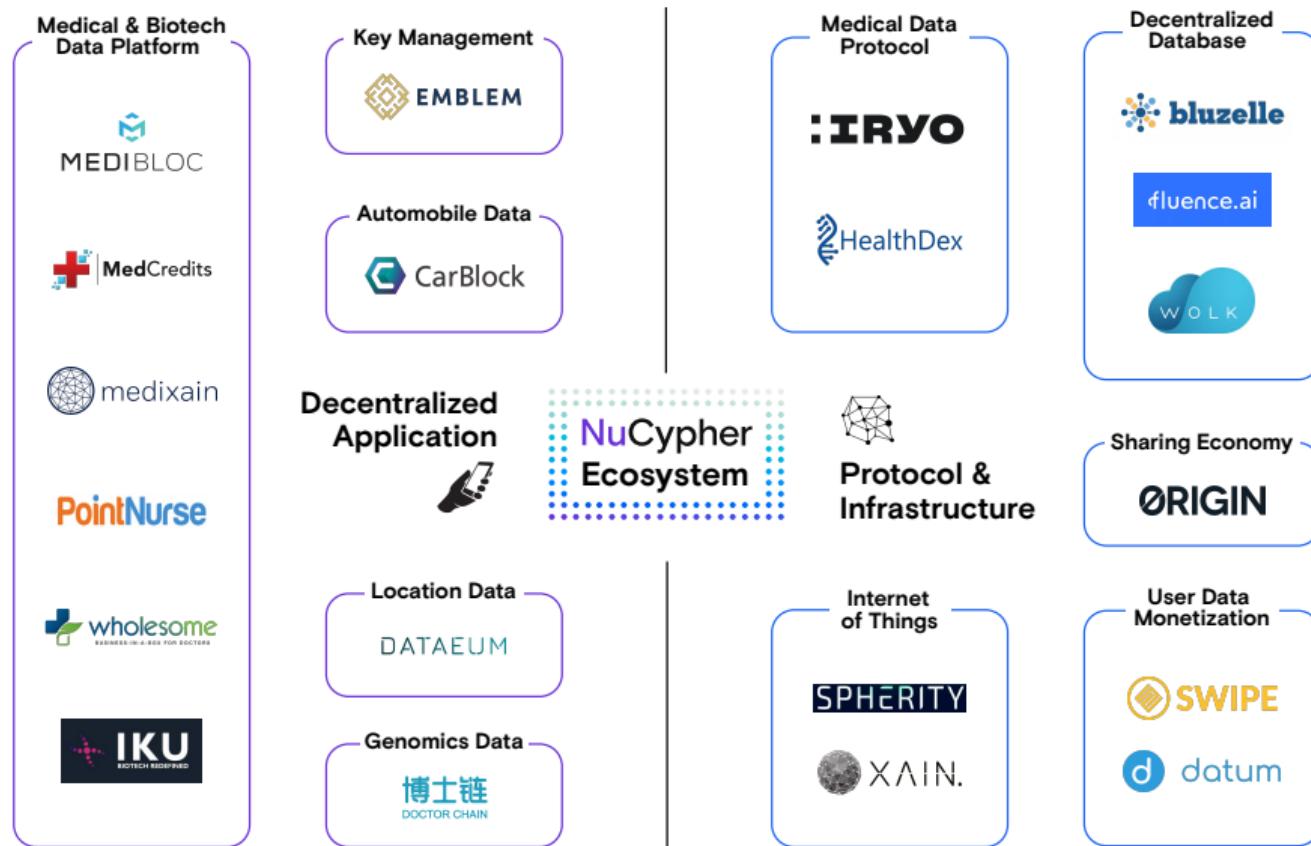


Use Cases

Decentralized Access-Controlled Content



Early Users



Competing Technology

Data Masking and Tokenization

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

Multi-Party Computation

- Early Research Stage
- Slow Performance

Fully Homomorphic Encryption

- Early Research Stage
- Slow Performance
 - ▶ NuCypher has invested efforts in this area

Investors

>\$15M in Venture Funding



AMINO Capital

BASE



Blockchain Partners Korea

CoinFund

compound



DHVC



F BIG
CAPITAL

FIRST MATTER



Kenetic
Capital



POLYCHAIN
CAPITAL

Satoshi•Fund

semantic
capital



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 Zотов
Engineer



Kieran Prasch
Engineer



Bogdan Opanchuk, PhD
Engineer



Ryan Caruso
Community



Derek Pierre
Business Development



Arjun Hassard
Product & Partnerships

More Information



NuCypher

Website: <https://nucypher.com>

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

Github: <https://github.com/nucypher>

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

Email: <fname>@nucypher.com

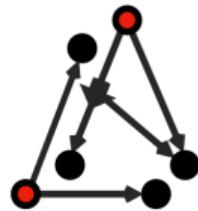
Email: hello@nucypher.com

Appendix: Umbral – Threshold Proxy Re-Encryption

Designed by: David Nuñez, University of Malaga, NICS Lab

- “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
 - ▶ 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>

Appendix: Security Audits



Least Authority
Freedom Matters

Appendix: Fully Homomorphic Encryption

