Have Your Cake and Eat It Too: Searchable Encryption

Srinivasan Narayananmurthy (Srini), NetApp

naras@netapp.com
About

- Me
  - Engineer at Advanced Technology Group since 2011
  - Spent a decade working on security before joining NetApp
    - Mostly on PhD & a few years at RSA Security
  - @NetApp: Distributed systems, NoSQL, Erasure codes
- My Involvement in SNIA
  - SDC talk on Modern Erasure Codes
  - SNIA Security TWG
    - Tries very hard to stay awake until 1:30AM (IST) to attend the weekly meeting! 😊
- Talk
  - Non-mathematical; non-algorithmic
The Problem

Semi-trusted (Honest-but-but-curious) server
Classification

Unstructured

Structured
Solution

Encrypt!
But…
Or...
Searchable Encryption

Encrypted Data-at-rest & Data-in-motion
Symmetric Searchable Enc. (SSE)

Inverted Index

Symmetric encrypt. (Eg. AES)

Trapdoor

SSE (Eg. Hash)

Access pattern leakage!
Oblivious RAM (oRAM)

Hides all information, including access pattern
Many rounds of communication; Large storage cost
SSE + oRAM

Hides all information, including access pattern;
Many rounds of communication, Large storage cost
Private Stream Searching (PSS)

Partial (Additive) homomorphism!
Order-preserving Encryption (OPE)

Symmetric encryption over integers

Range Query e.g. age > 3
Structured Data

Social networks, Web crawlers, Maps, Network routing, Communication (email headers, phone logs), Research papers (citations)
Structured Encryption (STE)
Fully Homomorphic Encryption (FHE)

2 + 5 = ?

4 7 3
2 0 5

2 + 5

Computationally expensive, high storage overhead
Search time is linear in the length of the dataset

Somewhat Homomorphic (SWHE):
Efficient; restricted number of additions and multiplications
Other Encryption Schemes

- **PKEET** (Public Key Encryption with Equality Test)
  - Equality tests of plaintexts encrypted under different public keys

- **PE (Predicate)** & **IPE (Inner Product)**
  - Access-control & (originally) equality tests
  - **IBE (Identity)**, **AIBE (Anonymous IBE)**, **HIBE (Hierarchical)**
  - **ABE (Attribute)**

- **HVE (Hidden Vector)**
  - Wild card characters inside a key
  - Supports: conjunctive, subset, range queries, disjunctions, polynomials, inner products
Summary

- Symmetric
  - Searchable Symmetric Encryption (SSE)
  - IND-CKA2 security
  - Efficient (sub-linear) SE schemes

- Asymmetric
  - Public key Encryption with Keyword search (PEKS)
  - Efficiency and security?
  - Lack of query expressiveness
Tradeoffs

Functionality

E.g. Efficiency
(subset, fuzzy, range queries, inner products)

Security

E.g. IND-CKA2

Query expressiveness
E.g. Equality, conjunctive, extended search

Computation & Communication

Leak: Index, search & access pattern
E.g. IND-CKA2

E.g. sub-linear index

Efficiency

19
Efficiency vs. Security

Efficiency

Leakage

ORAM
FHE
HVE (PEKS)
PE (IBE, ABE)
IPE (AIBE)
SSE
PKEET
PPE
STE
PEKS
Functionality vs. Efficiency

- ORAM
- FHE
- HVE (PEKS)
- PE (IBE, ABE)
- IPE (AIBE)
- SSE
- PEKS
- STE
- PKEET
- PPE
Applications

- Secure search
- Secure storage
  - Outsourced, Backup
- Secure Data management
  - Deduplication, email forwarding, etc.
- Security tiers for analytics
- Private data with “enough” privacy
  - Call logs, map queries, image search, data classification
In Practice

- **Systems**
  - CryptDB, MIT CSAIL
  - Cipherbase, Microsoft
  - Google’s Encrypted BigQuery Demo
  - Microsoft SQL Server 2016 Always Encrypted

- **Implementations**
  - CS2, Microsoft & UCB (2012); C++; Keyword search
  - IARPA, IBM & UCI (2013); C++; Conjunctive
  - BlindSeer, Bell Labs & Columbia (2014); Boolean
  - GRECS, Microsoft, Boston & Harvard (2015); C++; Graph
  - Clusion, Brown & Colorado (2016); Java; Boolean
Conclusion

- Tradeoffs: Security vs. Efficiency vs. Functionality
- Unclear security model
- Not-so-good asymmetric schemes
- Limited set of (academic) implementations

- But ...

This could be as big a wave as public-key crypto!
Thank you!

Download this presentation and others from SNIA’s Data Storage Security Summit at:

http://www.snia.org/dss-summit