Searchable Encryption

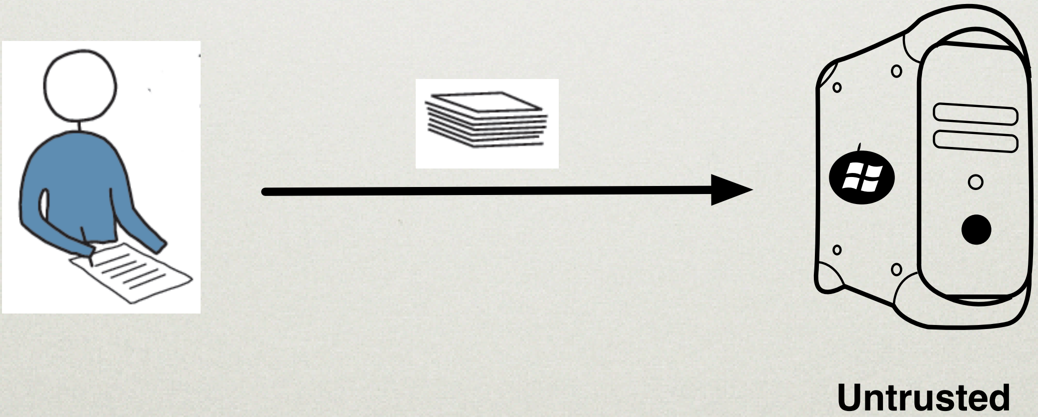
Prepared for 600.624
February 9, 2006

Outline

- Motivation of Searchable Encryption
- Searchable Encryption
- Constructions of Song, Wagner and Perrig
- Discussion
- Related Work
- Conjunctive Keyword Searches

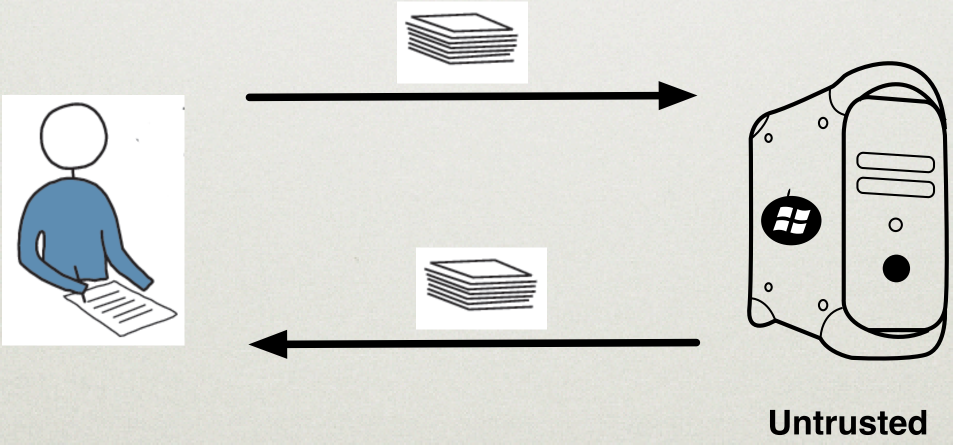
Motivation

- Proliferation of computing from different machines
- Want to store sensitive data remotely
  - e.g., email, audit logs, backups



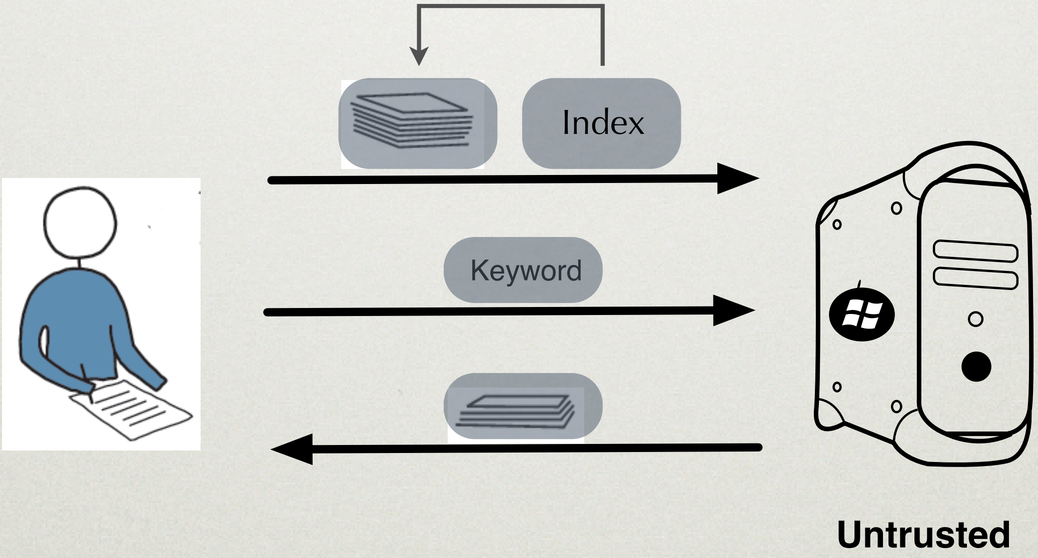
Motivation (2)

- Data must be encrypted
- Encryption prevents delegated searches
- Naive approach:



Searchable Encryption

- Combine an indexing scheme with trapdoors to allow server to search...



Searchable Encryption

- Goals:
  - Security
  - Correctness
  - Efficiency

Today's Paper

- Proposes the idea of Searchable Encryption
- Provides construction
  - basic idea: embed information in the ciphertext

Preliminaries (1)

- n , m -- block length, system parameter
- $G : \mathcal{K} \to S^l , |S_i| = n - m$
  - pseudo-random number generator
- $F : \mathcal{K} \times \{0,1\}^{n-m} \to \{0,1\}^m$
  - pseudo-random function

Preliminaries (2)

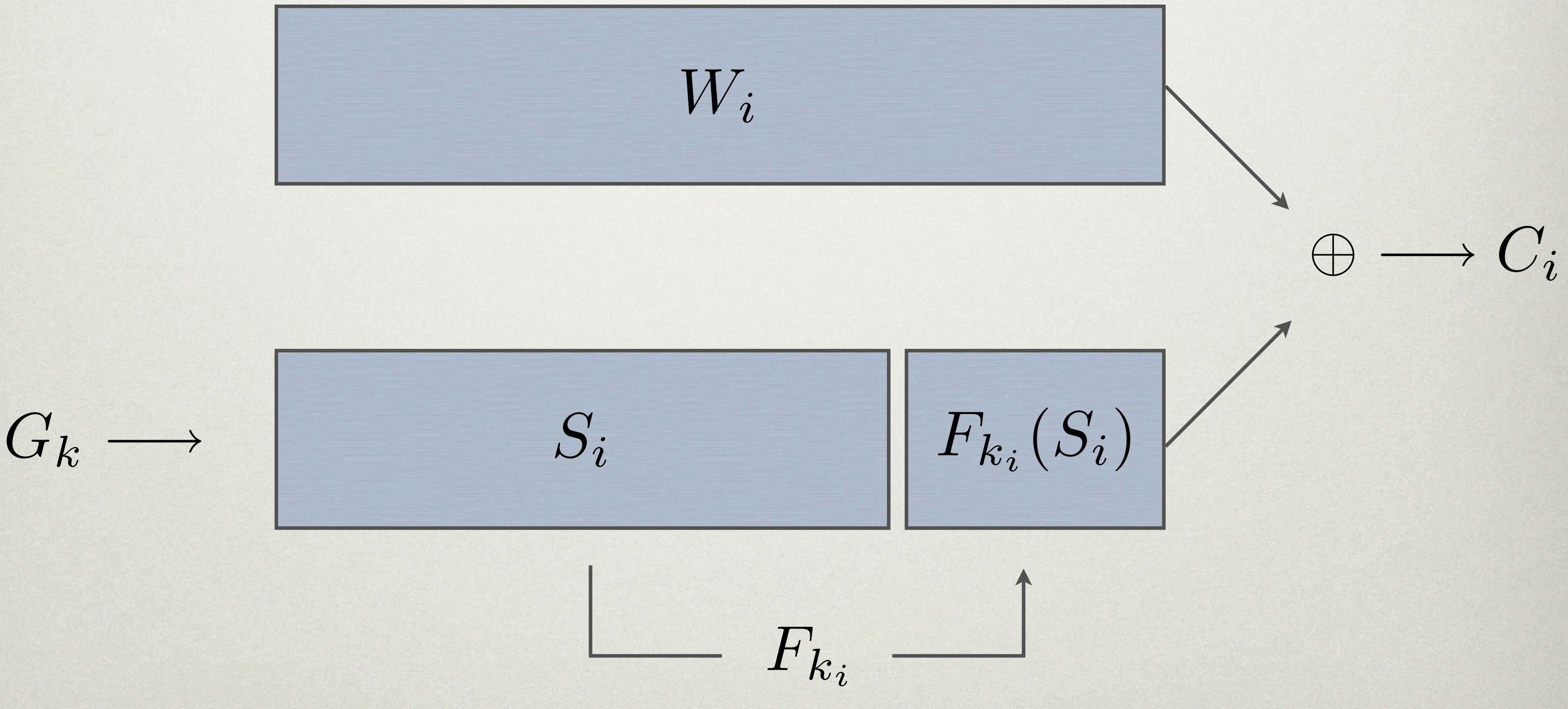
- $f : \mathcal{K} \times \{0,1\}^* \rightarrow \mathcal{K}$
  - pseudo-random function
- $E : \mathcal{K} \times \{0,1\}^n \rightarrow \{0,1\}^n$
  - pseudo-random permutation

Intuition

- Add structure to cipher-stream
    - Still secure
- Knowledge of word allows server to test for this structure

Construction #1

$$k_i \leftarrow f_{k'}(W_i)$$

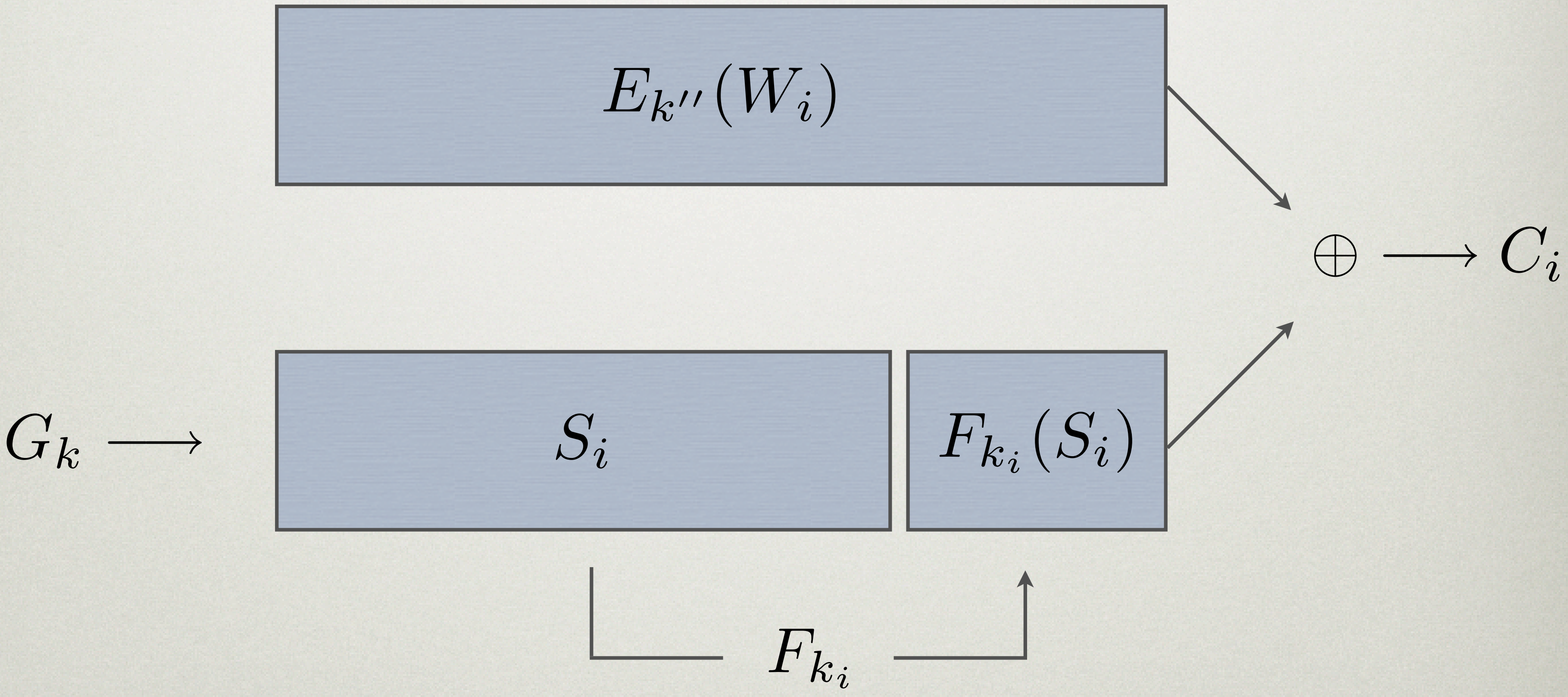


Limitations of #1

- Reveals the word we are searching
    - Fix this by encrypting the word
    - Must be a deterministic encryption!
- Who needs to decrypt anyway?

Construction #2

$$k_i \leftarrow f_{k'}(E_{k''}(W_i))$$

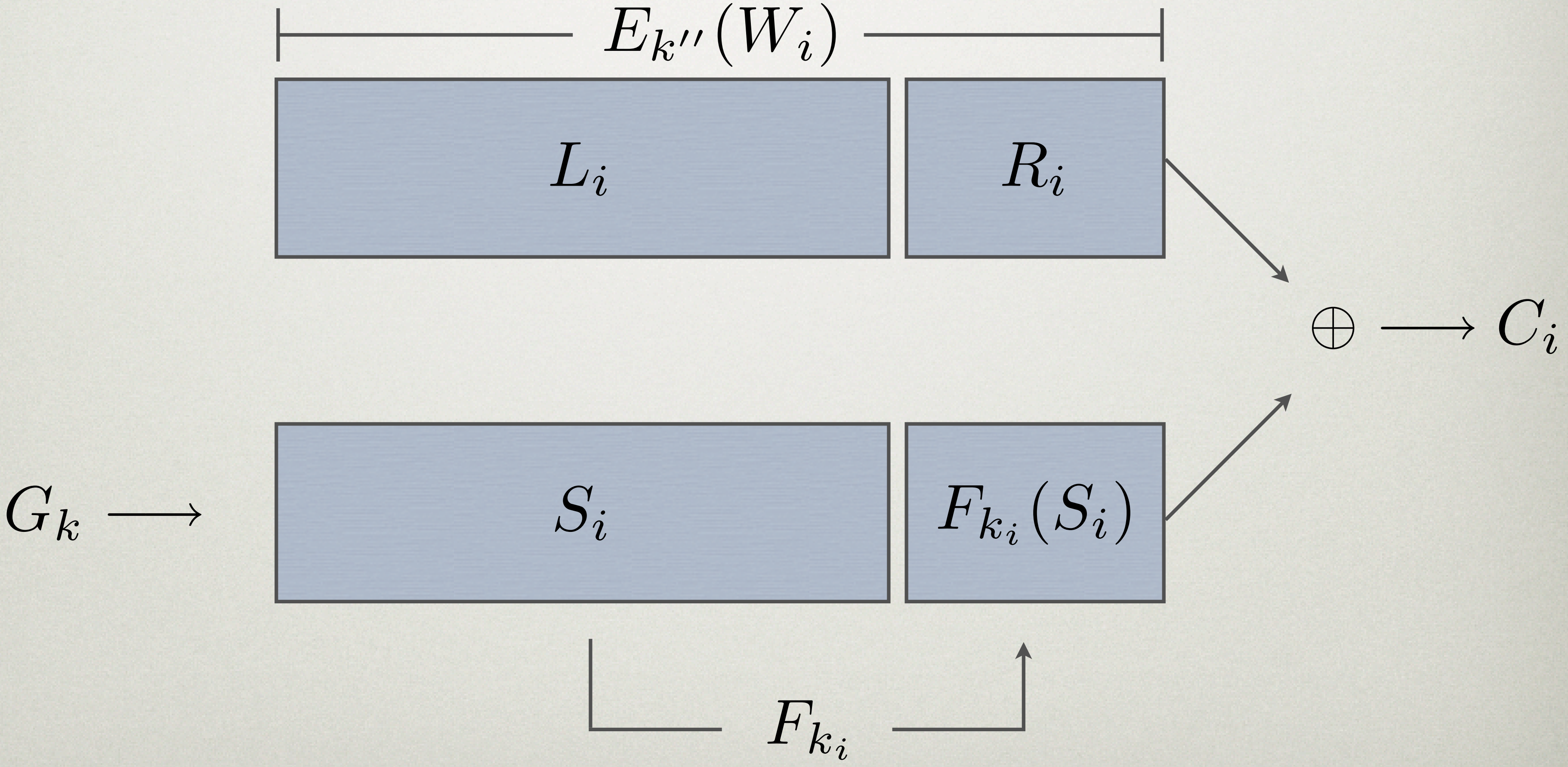


Limitations of #2

- ~~Reveals the word we are searching~~
- Who needs to decrypt anyway?
    - Problem: cipher-stream is a function of the plaintext---which we don’t know!
    - Solution: make it a function of the plaintext that we can actually derive!

Construction #3

$$k_i \leftarrow f_{k'}(L_i)$$



Recap

- Achieved secure keyword searches
  - Sequential scan through ciphertext
  - Extract stream structure using PRF and knowledge of the word
  - Protect word using PRP/PRF
- Questions?

Extensions (1)

- Boolean searches
  - everyone buy this?
- Regular expressions
- Searching for the nth occurrence of a word
  - thwarts statistical attacks?

Extensions (2)

- Variable-length words
  - what does this do to search time and false-positive rate?
- A Searchable Index
  - Advantages: can limit statistical information
  - Disadvantage: Difficult to update

N & M?

- Parameters of the System
- n --- word length
  - e.g., $n = 32$ “hi there” ⇒ [hi--] [_---] [ther] [e---]
  - Ciphertext expansion increases with n
  - Search speed increases with n
- m --- “check” length
  - Number of false matches ($\ell 2^{-m}$) are inversely proportional to m ... is this the only factor?
  - m cannot be too small... why?

Realizing N and M

- Implemented the system
- Downloaded english text from Project Gutenberg
- Measured performance under different loads
- Showed best tradeoffs results when

$$n = 32 \text{ bits},\ m = 8 \text{ bits}$$

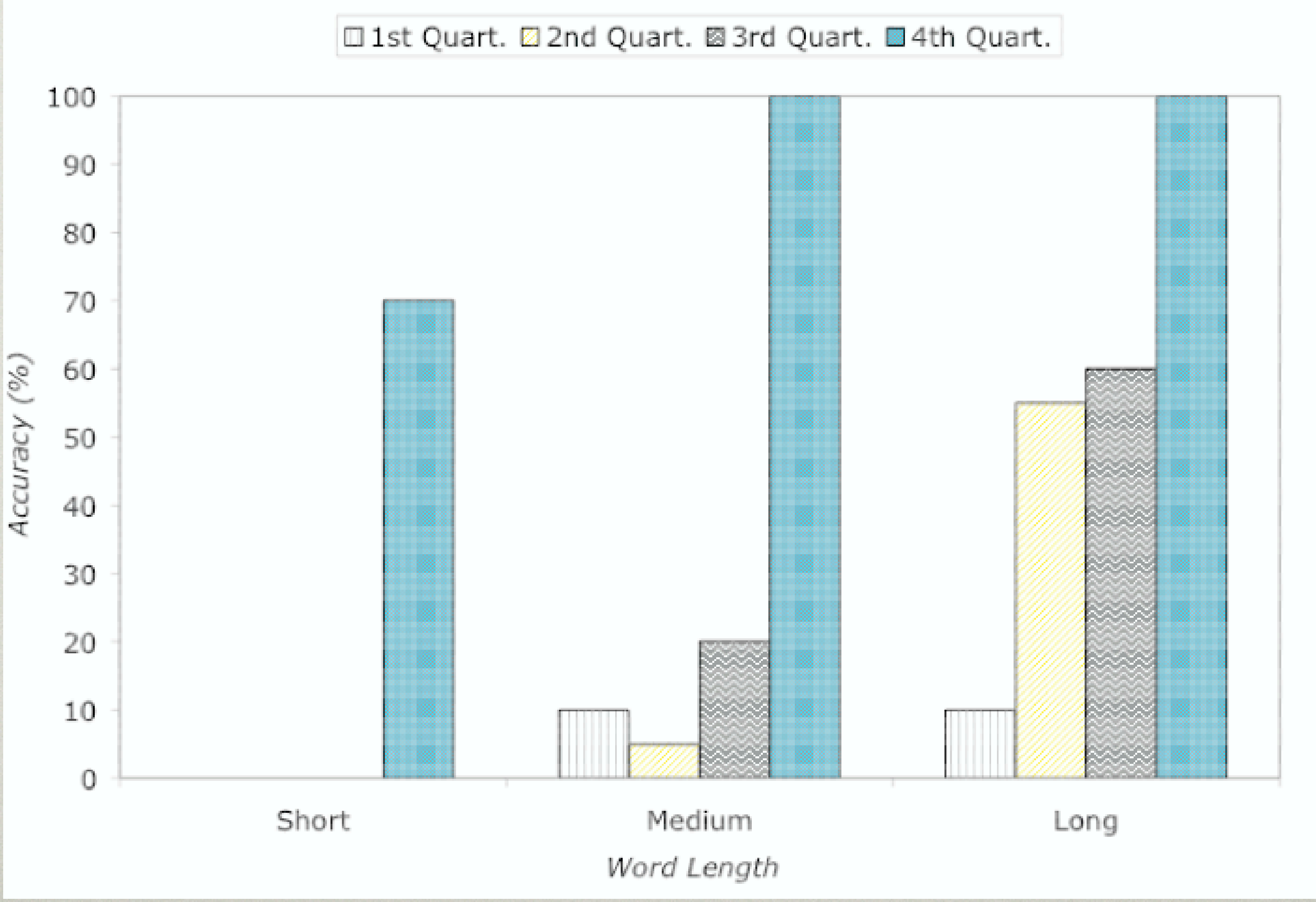
Implications of N and M

- Words are partitioned to have length 4
    - e.g., “Fabian” --> [Fabi] [an--]
- Searching of words spanning k partitions in a document of ℓ partitions has a false positive rate of $(\ell + 1 - k)/2^{8k}$

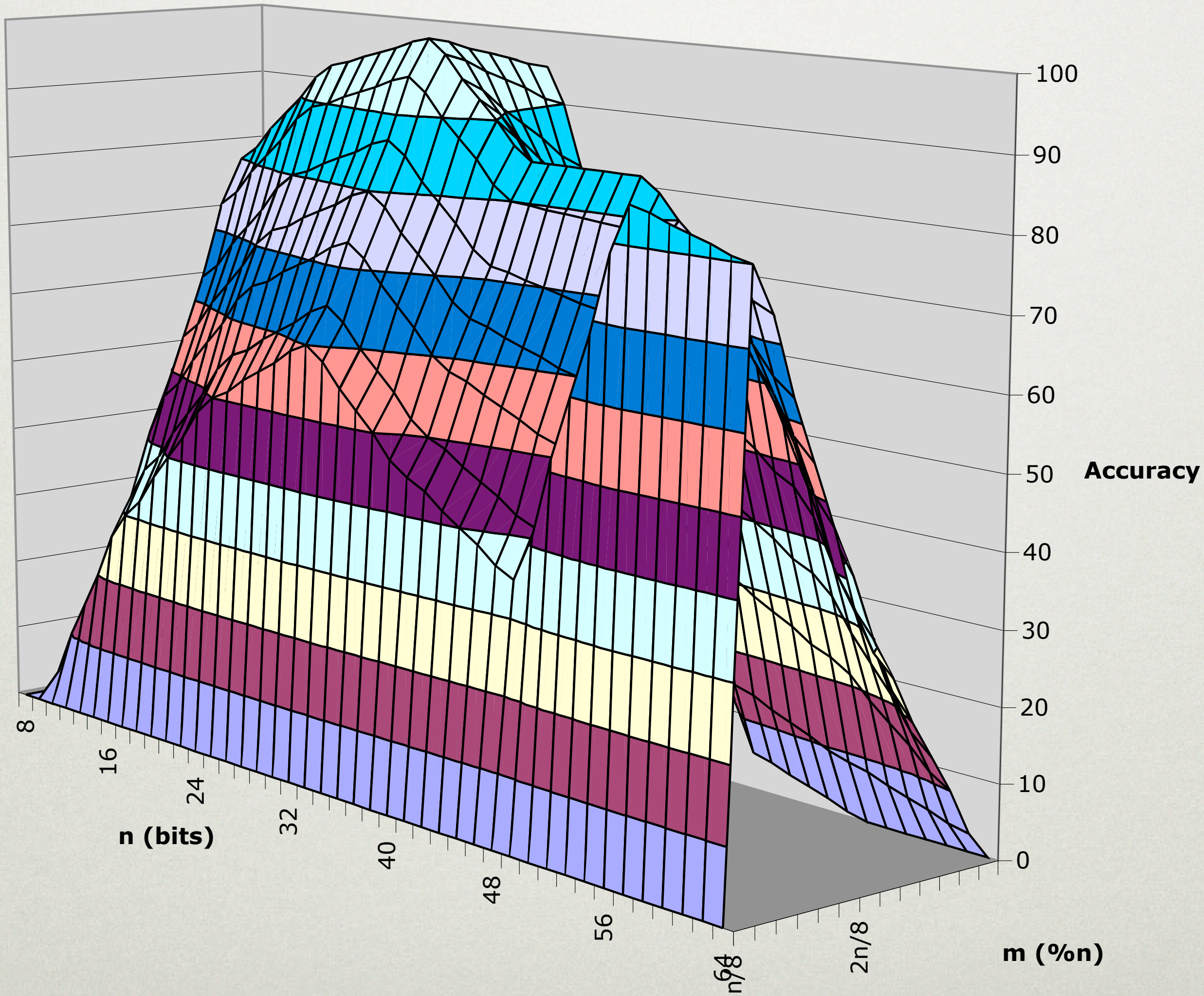
Statistical Attacks

- ECB mode encryption!!!
- Assumption: Malicious server has knowledge of plaintext distribution
- Records how many times a given query matches
  - Note: only considered ONE search

Statistical Attacks (2)



Statistical Attacks (3)



The Problem?

- Designed a new “encryption algorithm”
  - Revealed patterns in the plaintext
  - Perhaps we should consider alternate constructions

Security?

- Is this construction secure?
- There are proofs...
  - What did they prove?
- More on that tomorrow.

Related Work (see references)

- Private Information Retrieval [CGKS95]
- Oblivious RAMs [KO97]
- Secure Indexes [G03]
- **Keyword Search over Asymmetric Encryption [BdCOP04]**
    - **w/ applications to audit logs [WBDS04]**
- Boolean Keyword Search [GSW04, PKL04, BKM05]

Secure Audit Log Properties

- Tamper Resistant/verifiable
  - May need to offload to other machines
- Private
  - Contents are generally sensitive
- Searchable
  - Perhaps outsourced to an auditor

Applications: Secure Audit Logs

- Associate keywords with each log entry
  - e.g., “Failed login attempt”
- Encryption provides privacy
- Searchable Encryption allows auditors to do their job
- Problem: who encrypts the logs
  - the machine generating them?

Identity-Based Encryption

- Asymmetric Encryption
  - public key is a function of a string!!!
- Secret key (corresponding to a string) is created by TTP
  - has a master secret
- Greatly reduces PKI

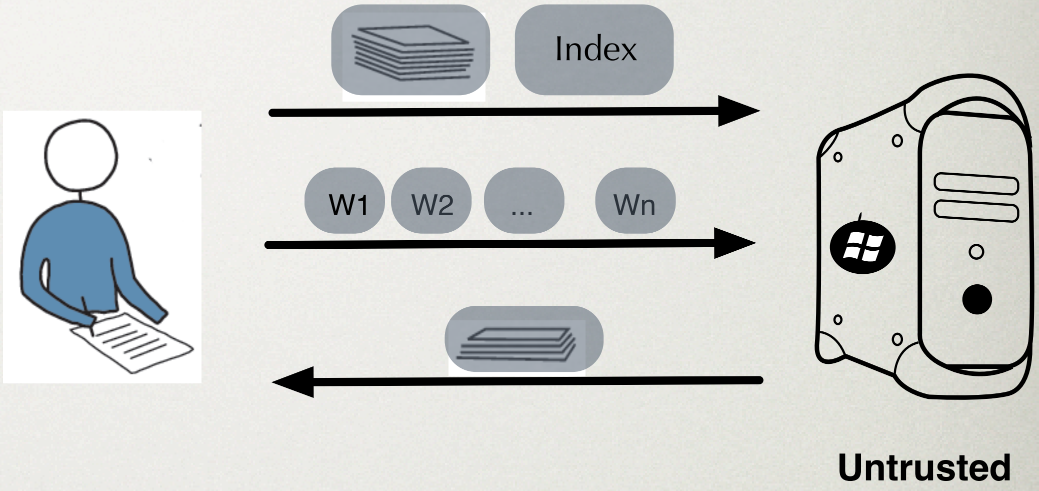
A need for Asymmetric Searchable Encryption

- Log entries encrypted with IBE
  - public key corresponds to keyword
- Escrow Agent knows IBE master secret
  - Can delegate secret-keys corresponding to any keyword to any auditor

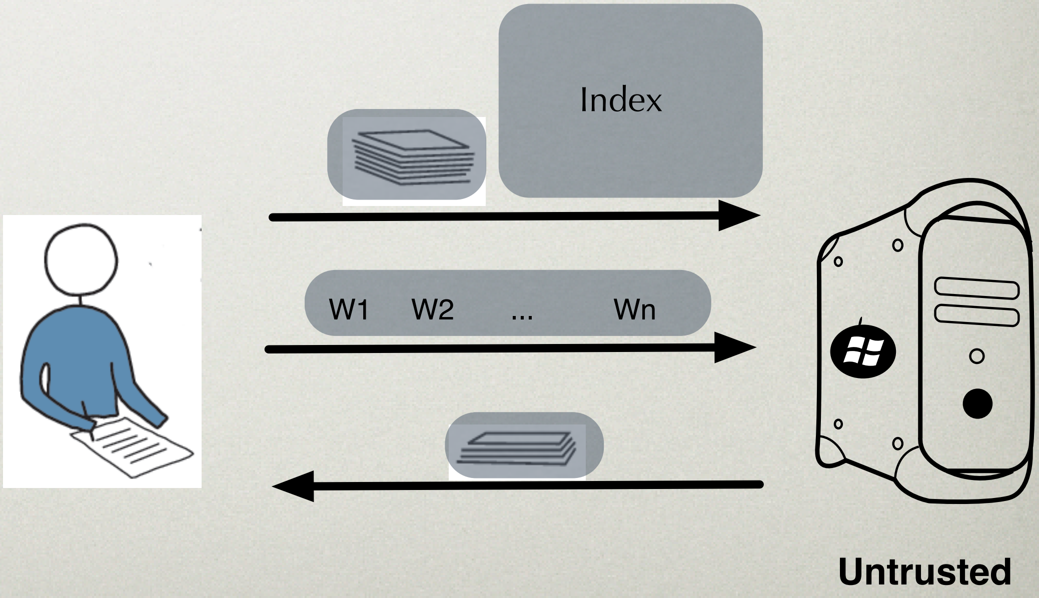
Back to Boolean Searches

Conjunctive Keyword Searches

- Send a trapdoor for each conjunct

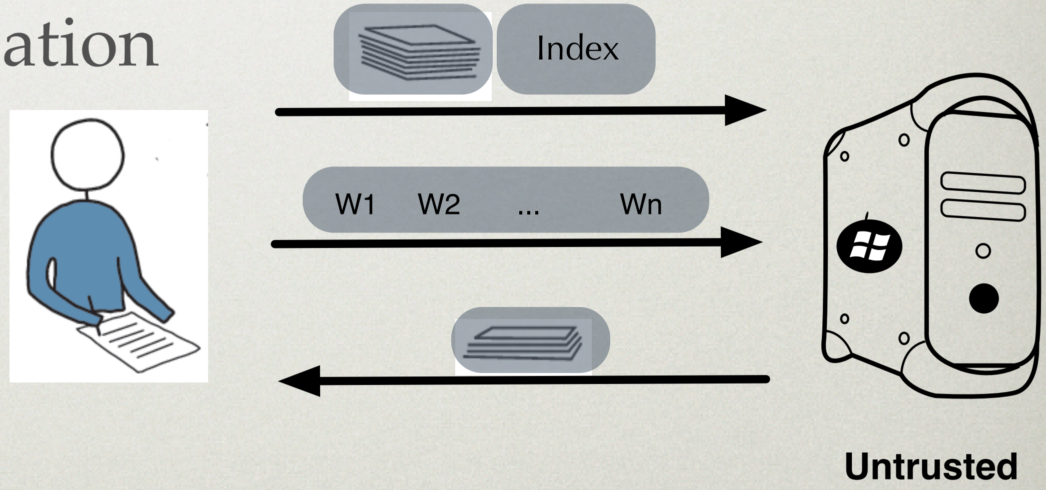


- Add every keyword combination to the index



Requirements of SCKS

- Security!
- Reasonable Index Size
- Small trapdoors
- Efficient Index Generation
- Efficient trapdoor generation
- Efficient search



Work with Seny & Fabian

- Two constructions:
  - SCKS-SS and SCKS-XDH
- Symmetric conjunctive searchable encryption
- Use formal definitions from Goh (2003)
- constructions more efficient than Golle et al. (2004)

Standard Assumptions

- For efficiency documents are associated with a list of keywords
- Trapdoors specify which elements of the index to search on
- Keywords are distinct
    - add field name such as SUBJECT: or FROM:
- Each document has a fixed number of keywords
    - add NULL keywords to pad

SCKS-SS

- Most computationally-efficient construction known to date
- Based on
  - Shamir Secret Sharing
  - PRFs

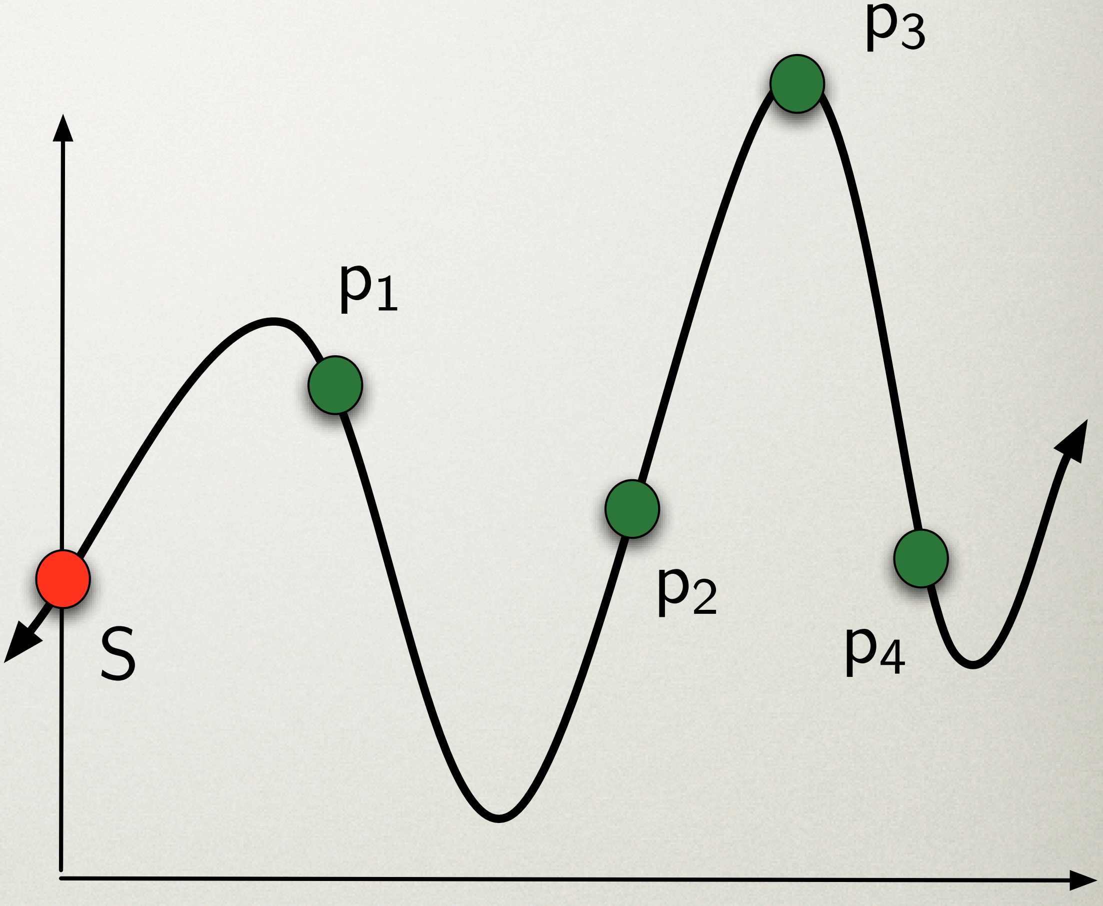
Shamir Secret Sharing

$S \in \mathbb{Z}_p$

$\mathcal{P} \overset{R}{\leftarrow} \mathbb{Z}_p[x], \ \deg = k - 1$

$\mathrm{share}(S) \to p_1, \ldots, p_n$

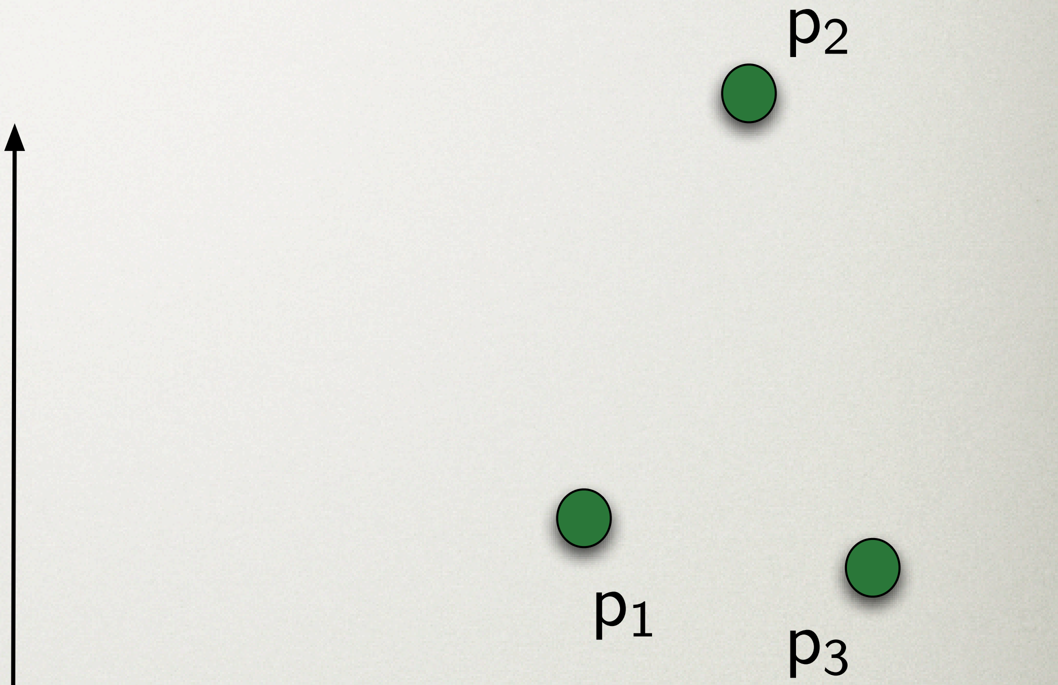
$\mathrm{recover}(p_1, \ldots, p_k) \to S$

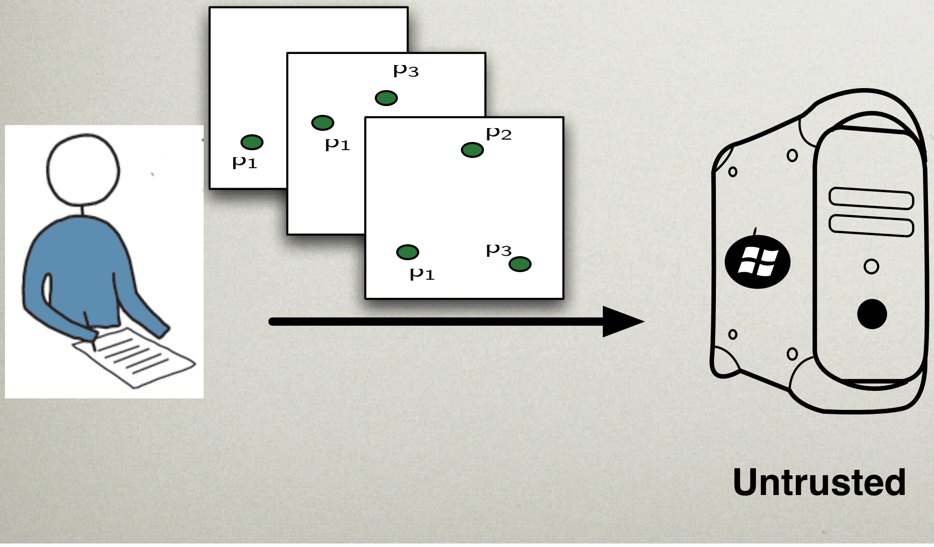


Build Index

Generate Index (for each document ID)

$$\mathrm{BuildIndex}(w_1, w_2, w_3) \rightarrow p_1, p_2, p_3$$

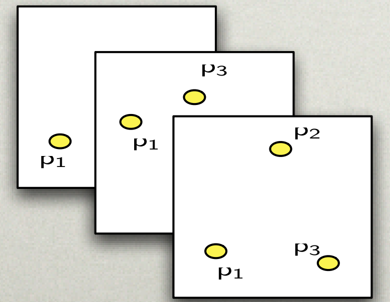


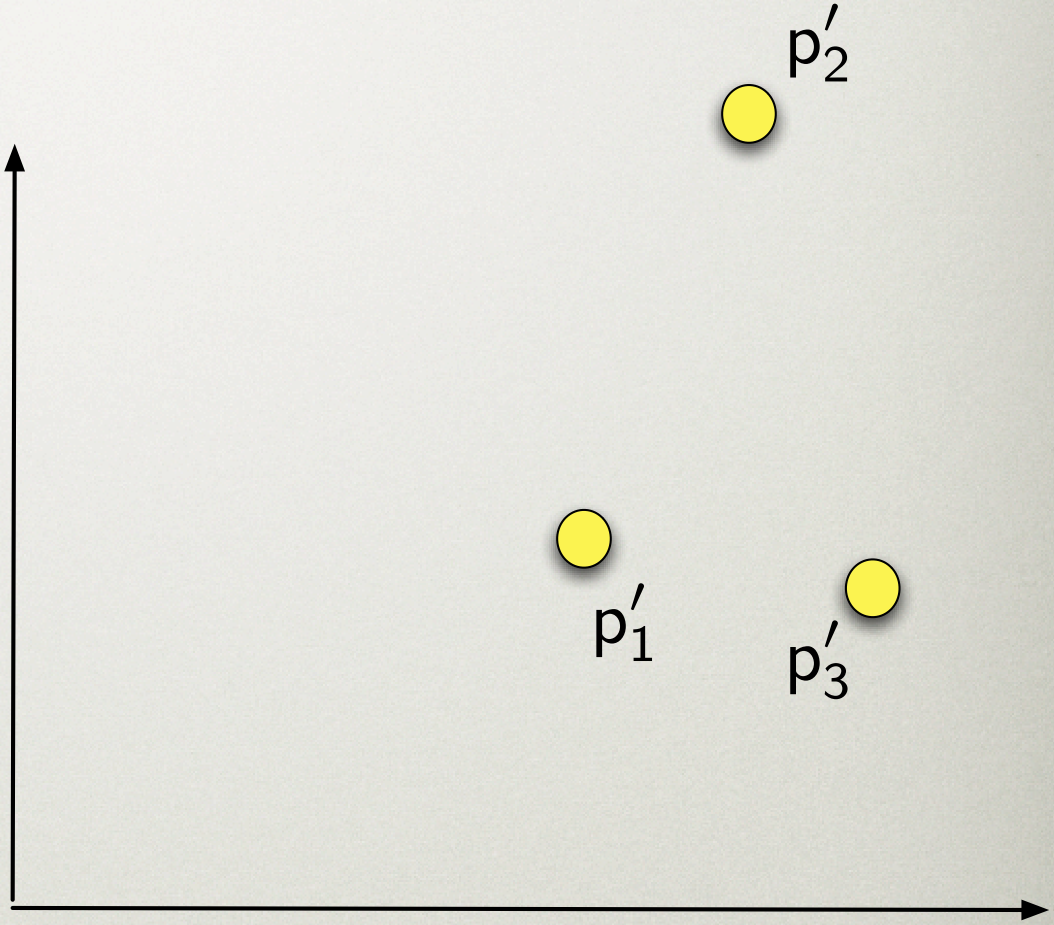


Trapdoor (1/1)

Generate Trapdoor (for each document ID)

$w'_1 \wedge w'_2 \wedge w'_3$



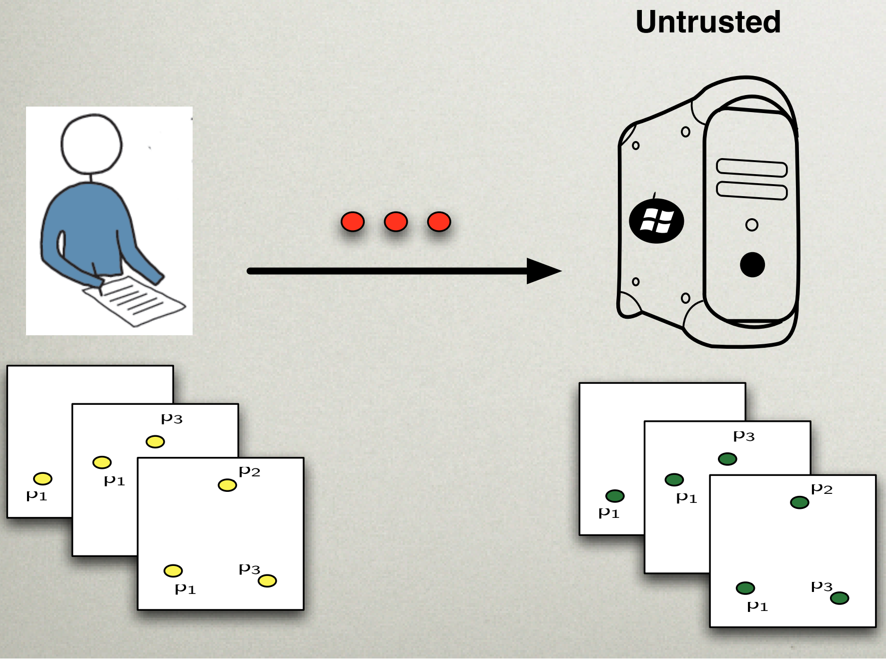


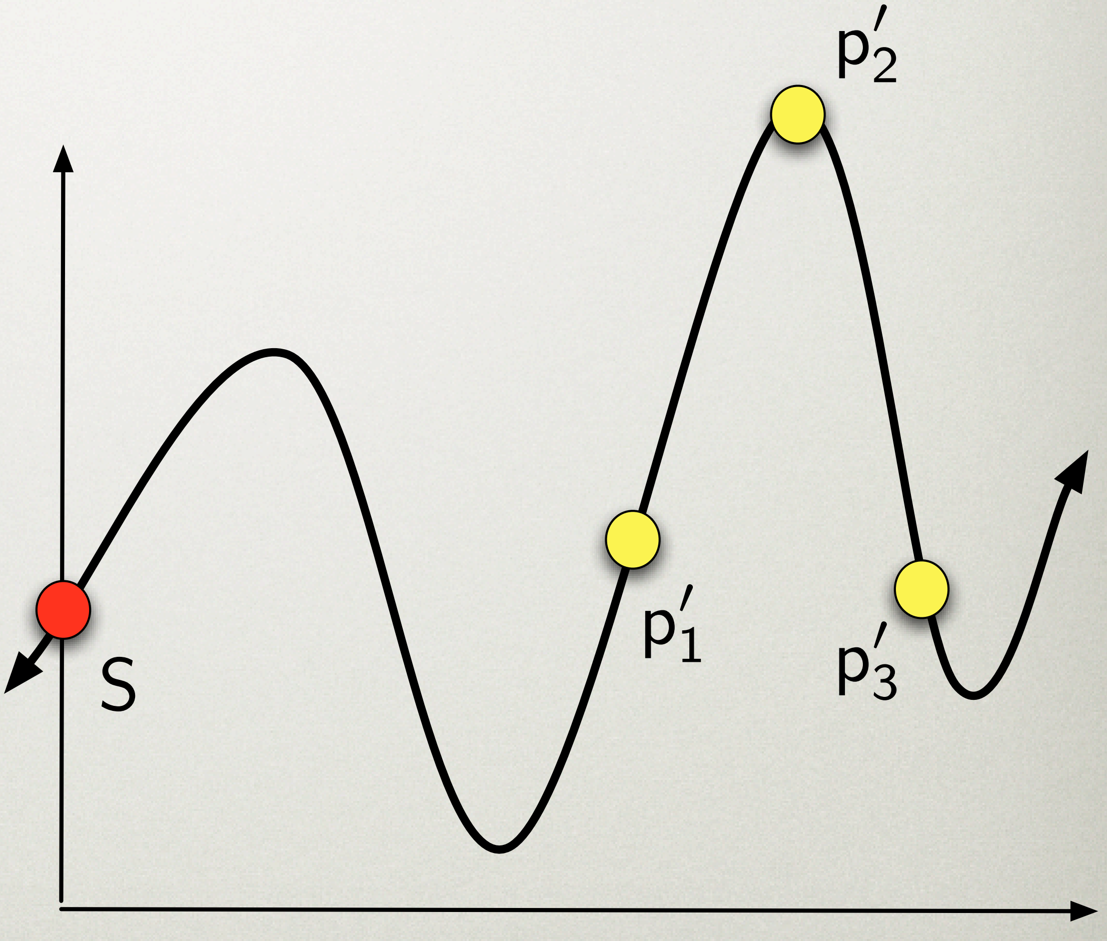
Trapdoor (2/2)

Generate Trapdoor (for each document ID)

$w'_1 \wedge w'_2 \wedge w'_3$

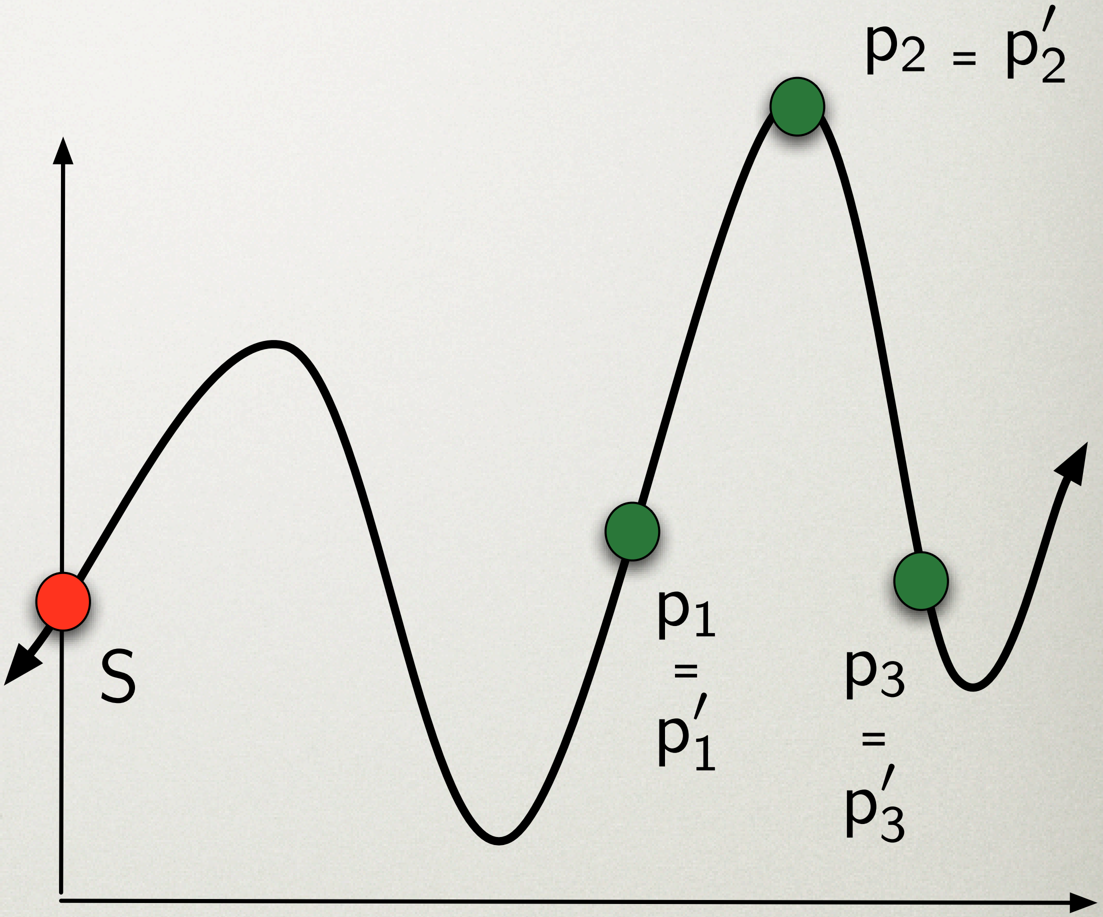
$\text{Trapdoor}(w'_1, w'_2, w'_3) \rightarrow S$

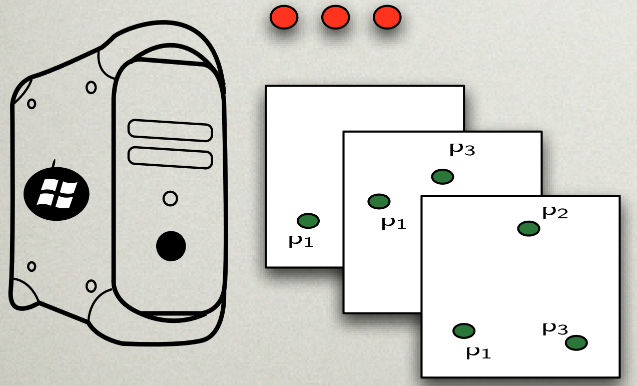




Successful Search

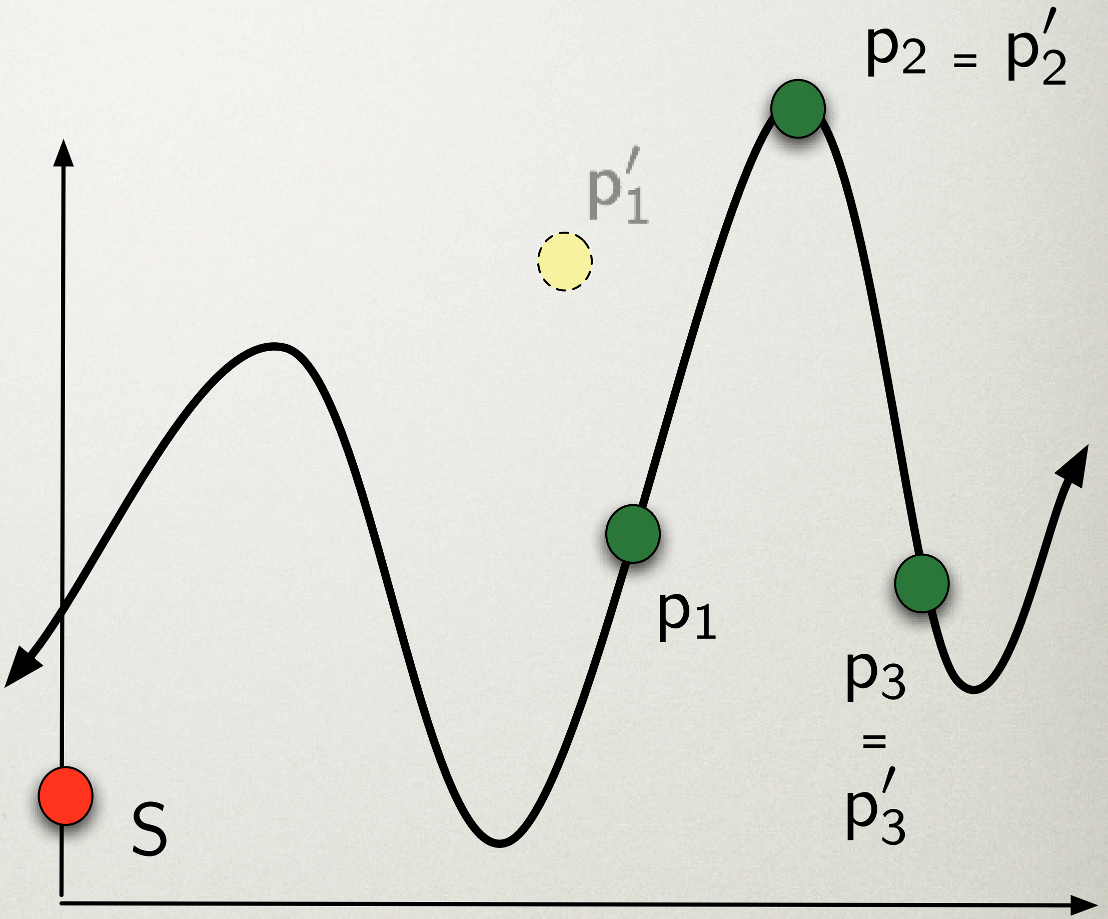
Successful search (for each document)

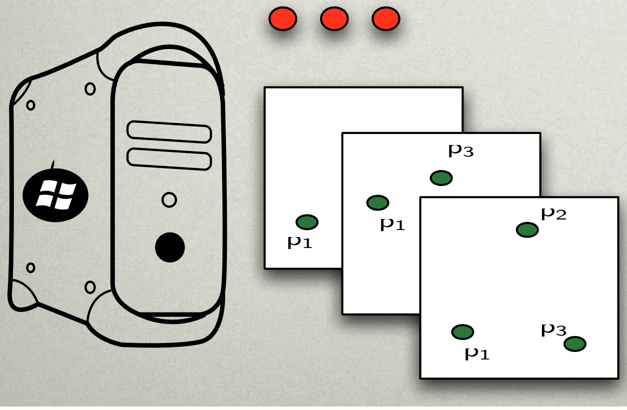




Failed Search

Failed search





Asymptotic Performance

	Linear Trapdoors		Constant Trapdoors	
	GSW-1	SCKS-SS	GSW-2	SCKS-XDH
Search	2m exp, m hash	m interpolations	m(2n+1) Pairings	2m Pairings

m: number of documents

n: number of keywords

Empirical Evaluation

- Ran tests on 3.0 GHz P4
- Implemented constructions with C++
  - OpenSSL (PRF)
  - MIRACL (curve operations, mod arithmetic)
- Measured time to process 10,000 documents with ≤ 10 keywords each
  - BuildIndex, Trapdoor, SearchIndex

SCKS-SS

Computation

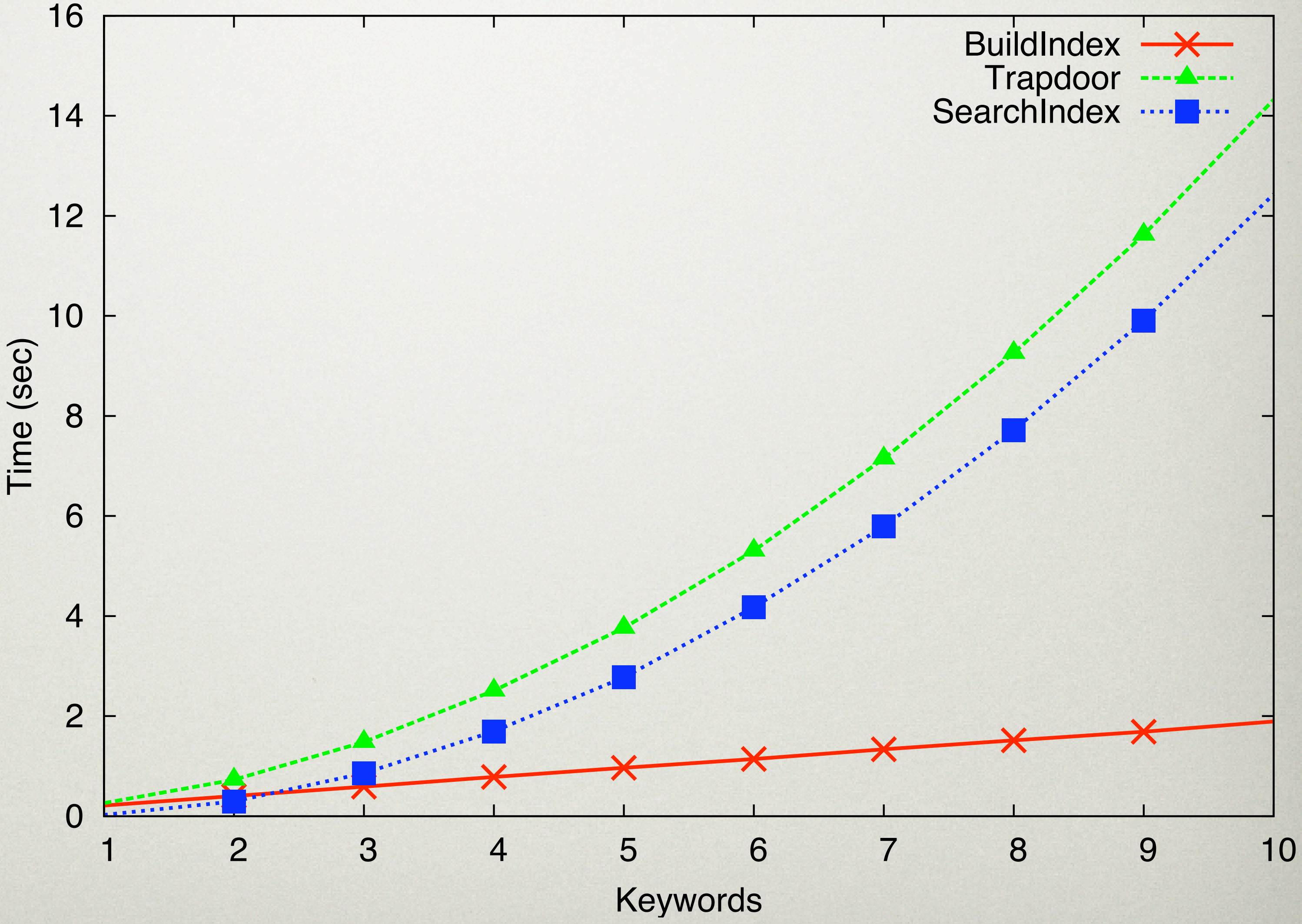
10 000 documents

Storage

10 Keywords

Index: 3.1 MB

Trap: 156 KB



- Time for SCKS-XDH?

Conclusion

- Searchable Encryption
- Excellent Idea, area is gaining momentum
- Lots of interesting problems:
    - Work on adequate security models
    - Boolean Searches
    - Regular Expression Matching

Questions?

References (1)

- M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, H. Shi, “Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions,” CRYPTO 2005.
- L. Ballard, S. Kamara, F. Monrose, “Achieving Efficient Conjunctive Keyword Searches over Encrypted Data,” ICICS 2005.
- D. Boneh, G. Di Crescenzo, R. Ostrovsky, G. Persiano, “Public Key Encryption with Keyword Search,” EUROCRYPT 2004.
- Y.C. Chang, M. Mitzenmacher, “Privacy Preserving Keyword Searches on Remote Encrypted Data,” ACNS 2005.
- B. Chor, O. Goldreich, E. Kushilevitz, M. Sudan, “Private Information Retrieval,” FOCS 1995.
- D. Davis, F. Monrose, M. Reiter, “Time Scoped Searching of Encrypted Audit Logs,” ICICS 2004.

References (2)

- E. Goh, “Secure Indexes”, Cryptology ePrint Archive, Report 2003/216, 2003.
- P. Golle, J. Staddon, B. Waters, “Secure Conjunctive Keyword Searches over Encrypted Data,” ACNS 2004.
- E. Kusilevitz, R. Ostrovsky, “Replication is not needed: Single Database, Computationally-Private Information Retrieval,” FOCS 1997.
- D. Park, K. Kim, P. Lee, “Public Key Encryption with Conjunctive Field Keyword Search,” WISA 2004.
- **D. Song, D. Wagner, A. Perrig, “Practical Techniques for searches on Encrypted Data,” S&P 2000.**
- B. Waters, D. Balfanz, G. Durfee, D. Smetters, “Building an Encrypted and Searchable Audit Log,” NDSS 2004.