CMSC 711 Computer Networks **Spring 2016**

Mid-term Exam

Open book and notes; Take home *Due: Tuesday, Mar. 29th*

- I cannot stress this point enough: **Be precise**. If you have written something incorrect along with the correct answer, you should **not** expect to get all the points. I will grade based upon what you **wrote**, not what you **meant**.
- Typeset your answers, ideally using LaTeX. Use 10pt font, with reasonable margins (1 – 1.25 inches). Constrain the text of each answer to one page. Figures and diagrams do not count towards the page limit.
- Please cite every external reference you quote or derive an idea from.
- I will grade your solutions both on correctness and presentation.
- Maximum possible points: 8000 ($1000 \times 2 + 1500 + 2000 + 2500$).

Problem	Points
Search	
Untrusted Storage	
CRL Analysis	
Ephemeral Data	
Trinc BlockChain	
Total	

Search

- Assume that Alice stores her data on Bob's server. She wants to be able to search her data without retrieving all of her content, but she does not want Bob to be able to gather information about her content.

  Analyze the following protocols for "search over encrypted data". State your adversarial model(s) (i.e. what can Bob do), how much (extra) state Alice and Bob have to maintain, and what guarantees, if any, the protocols give to Alice. In particular, state whether protocol is correct (in that it allows Alice to search and also retrieve her document) and what attacks each protocol is susceptible to. Finally, describe what types of searches (word occurs in document, phrase occurs in document, and so on) each protocol allows.

Protocol 0 Alice stores the document encrypted using symmetric encryption and also a search digest. The search digest is a Bloom Filter into which she hashes each word in the document.

During a search, Alice states the search term(s) to Bob, who then checks the corresponding Bloom Filters and returns matching documents.

Protocol 0a Same as Protocol 0, but Alice hashes each word in the digest (and search term) using a secret hash function.

Protocol 1 Alice uses a hash function (that she keeps secret) to map each word in the document to its "encrypted" form. In order to search, she translates the search term(s) using the same hash, and asks Bob to do a string match.

Protocol 1a Same as Protocol 1, but Alice encrypts each word instead of hashing them.

Protocol 2 Alice encrypts her files using symmetric encryption. She then constructs a inverted index that stores (word, document-ID) pairs for each word she wishes to search for. For example, if Documents 1 and 8 had the word "cat", the inverted index would contain two entries (cat, 1) and (cat , 8). Instead of simply putting "cat" in the index, Alice stores a hash of "cat"; otherwise, Bob could simply reconstruct the documents reading the indices.

Alice stores both the encrypted document and the inverted index at Bob. She sends the search terms appropriately hashed, and Bob returns all the documents that contain the term(s) in the index.

Protocol 2a Same as Protocol 2, but Alice adds "chaff", i.e. a set of extra words, hashed as before, chosen uniformly at random pointing to document IDs chosen uniformly at random.

- Can you reduce the processing required in Schemes III and IV in the "Practical Techniques for Searches on Encrypted Data" paper by Song et al. (available at `http://www.cs.berkeley.edu/~dawnsong/papers/se.pdf`) without affecting security?

Bonus Suggest a new algorithm or describe a technique from a paper not listed here. For either, compare to the Song et al. paper.

Untrusted Storage Suppose Alice and Bob want to replicate 1 MB of data for each other. Alice is honest, but suspects Bob may discard (some of) her data. Alice wants to ensure that Bob is faithfully storing her data.

She devises a family of protocols whereby she challenges Bob to produce a function of the data. The protocol is secure iff Bob must store the data in order to answer the challenge.

For each, state if they are secure. If not, show how Bob can break them. For each protocol, state how much state Bob and Alice have to store, how much precomputation Alice has to do, and how much work Bob has to do during the verification. Finally, give an estimate of how much data the protocol must send on the network during the verification.

Protocol 0

1. Give data to Bob
2. Periodically challenge Bob to produce the data and compare against stored copy

Protocol 1

1. Compute Fletcher checksum sum of data
2. Give data to Bob
3. Periodically challenge Bob to produce the checksum

Protocol 2

1. Seed a PRNG and generate 1024 permutations of the numbers $[0 \ldots 2^{20} - 1]$
2. 1024 times
   (a) Permute data bytes using one of the permutations
   (b) Compute the Fletcher checksum of the permuted data
   (c) Store the sum
3. Give data to Bob
4. Periodically send one of the permutations to Bob and challenge him to produce the corresponding checksum

Protocol 2.efficient

1. Alice decides to use the more efficient 2-s complement instead of Fletcher's checksum

Protocol Schwarz and Miller, Shacham and Waters

1. Alice finally finds two papers
   – Schwarz and Miller, "Store, Forget, and Check..." paper; `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1648799&tag=1`, and
   – Shacham and Waters, "Compact Proofs of Retrievability", `http://cseweb.ucsd.edu/~hovav/dist/verstore.pdf`.

   Explain the technique described in either paper. (Note that I will ask you questions during a one-on-one session on the paper that you choose; your grade will depend on your written explanation and answers to questions.)

CRL Analysis Obtain a set of certificate revocation lists from certificate authorities. Parse these lists to obtain times when certificates are revoked. Analyze the revocation times and present your analysis. You could, for instance, analyze how often and when CAs revoke certificates; if some CAs behave differently than others, whether the rate of revocations has changed over time, whether certain types of certificates are more likely to be revoked than others.

Your work will be graded on the size of the list you work on, the method you undertook for the analysis, the accuracy of your analysis, and on *interesting* observations you are able to validate.

Ephemeral Data Design a protocol whereby Alice can provide a data reference (a capability/URL/...) to Bob. The reference identifies a data item and an *expiry time* t. The semantics of the reference are:

- No one without a reference can read the document.
- Alice should not have to be online for Bob to obtain the document
- Bob can obtain and read the document prior to t
- Bob may make copies of the document if he obtained it prior to t
- After t or later, Bob can no longer obtain/read the document

Design a system that provides these properties (to the extent possible). Explain your assumptions and the vulnerabilities of your protocol.

Acceptable solutions should not use a trusted third party. In particular, the trivial solution where Alice puts the document on a single trusted server that checks for expiry will not receive any credit.

- Secure Blockchain

  Read the "TrInc: Small trusted hardware for large distributed systems" paper.
  `http://research.microsoft.com/apps/pubs/default.aspx?id=78369`

  Describe how you could use TrInc to create a more secure version of the BitCoin Blockchain that makes double spending impossible.

  State your assumptions, the security/protocol properties you provide, and the problems with your solution, if any.