Authenticated Data Structures

Roberto Tamassia

Department of Computer Science
Brown University
Providence, RI 02912–1910, USA
rt@cs.brown.edu
http://www.cs.brown.edu/~rt/

Abstract. Authenticated data structures are a model of computation where untrusted responders answer queries on a data structure on behalf of a trusted source and provide a proof of the validity of the answer to the user. We present a survey of techniques for designing authenticated data structures and overview their computational efficiency. We also discuss implementation issues and practical applications.

1 Introduction

Data replication applications achieve computational efficiency by caching data at servers near users but present a major security challenge. Namely, how can a user verify that the data items replicated at a server are the same as the original ones from the data source? For example, stock quotes from the New York Stock Exchange are distributed to brokerages and financial portals that provide quote services to their customers. An investor that gets a stock quote from a web site would like to have a secure and efficient mechanism to verify that this quote is identical to the one that would be obtained by querying directly the New York Stock Exchange.

A simple mechanism to achieve the authentication of replicated data consists of having the source digitally sign each data item and replicating the signatures in addition to the data items themselves. However, when data evolves rapidly over time, as is the case for the stock quote application, this solution is inefficient.

Authenticated data structures are a model of computation where an untrusted responder answer queries on a data structure on behalf of a trusted source and provides a proof of the validity of the answer to the user.

In this paper, we present a survey of techniques for designing authenticated data structures and overview bounds on their computational efficiency. We also discuss implementation issues and practical applications.

2 Model

The *authenticated data structure* model involves a structured collection S of objects (e.g., a set or a graph) and three parties: the *source*, the *responder*, and the *user*. A repertoire of *query operations* and optional *update operations* are assumed to be defined over S. The role of each party is as follows:

G. Di Battista and U. Zwick (Eds.): ESA 2003, LNCS 2832, pp. 2–5, 2003.

- The *source* holds the original version of S. Whenever an update is performed on S, the source produces *structure authentication information*, which consists of a signed time-stamped statement about the current version of S.
- The *responder* maintains a copy of S. It interacts with the source by receiving from the source the updates performed on S together with the associated structure authentication information. The responder also interacts with the user by answering queries on S posed by the user. In addition to the answer to a query, the responder returns *answer authentication information*, which consists of (*i*) the latest structure authentication information issued by the source; and (*ii*) a *proof* of the answer.
- The *user* poses queries on S, but instead of contacting the source directly, it contacts the responder. However, the user trusts the source and not the responder about S. Hence, it verifies the answer from the responder using the associated answer authentication information.

The data structures used by the source and the responder to store collection S, together with the algorithms for queries, updates, and verifications executed by the various parties, form what is called an *authenticated data structure*.

In a practical deployment of an authenticated data structure, there would be several geographically distributed responders. Such a distribution scheme reduces latency, allows for load balancing, and reduces the risk of denial-of-service attacks. Scalability is achieved by increasing the number of responders, which do not require physical security since they are not trusted parties.

3 Overview of Authenticated Data Structures

Throughout this section, we denote with n the size of the collection S maintained by an authenticated data structure.

Early work on authenticated data structures was motivated by the *certificate revocation* problem in public key infrastructure and focused on *authenticated dictionaries*, on which membership queries are performed.

The *hash tree* scheme introduced by Merkle [17,18] can be used to implement a static authenticated dictionary. A hash tree T for a set S stores cryptographic hashes of the elements of S at the leaves of T and a value at each internal node, which is the result of computing a cryptographic hash function on the values of its children. The hash tree uses linear space and has $O(\log n)$ proof size, query time and verification time. A dynamic authenticated dictionary based on hash trees that achieves $O(\log n)$ update time is described in [19]. A dynamic authenticated dictionary that uses a hierarchical hashing technique over skip lists is presented in [9]. This data structure also achieves $O(\log n)$ proof size, query time, update time and verification time. Other schemes based on variations of hash trees have been proposed in [2,6,13].

A detailed analysis of the efficiency of authenticated dictionary schemes based on hierarchical cryptographic hashing is conducted in [22], where precise measures of the computational overhead due to authentication are introduced. Using

this model, lower bounds on the authentication cost are given, existing authentication schemes are analyzed and a new authentication scheme is presented that achieve performance very close to the theoretical optimal.

An alternative approach to the design of authenticated dictionary, based on the *RSA accumulator*, is presented in [10]. This technique achieves constant proof size and verification time and provides a tradeoff between the query and update times. For example, one can achieve $O(\sqrt{n})$ query time and update time.

In [1], the notion of a *persistent authenticated dictionary* is introduced, where the user can issue historical queries of the type "was element e in set S at time t".

A first step towards the design of more general authenticated data structures (beyond dictionaries) is made in [5] with the authentication of relational database operations and multidimensional orthogonal range queries.

In [16], a general method for designing authenticated data structures using hierarchical hashing over a search graph is presented. This technique is applied to the design of static authenticated data structures for pattern matching in tries and for orthogonal range searching in a multidimensional set of points.

Efficient authenticated data structures supporting a variety of fundamental search problems on graphs (e.g., path queries and biconnectivity queries) and geometric objects (e.g., point location queries and segment intersection queries) are presented in [12]. This paper also provides a general technique for authenticating data structures that follow the *fractional cascading* paradigm.

The software architecture and implementation of an authenticated dictionary based on skip lists is presented in [11]. A distributed system realizing an authenticated dictionary, is described in [7]. This paper also provides an empirical analysis of the performance of the system in various deployment scenarios. The authentication of distributed data using web services and XML signatures is investigated in [20]. *Prooflets*, a scalable architecture for authenticating web content based on authenticated dictionaries, are introduced in [21].

Work related to authenticated data structures includes [3,4,8,14,15].

Acknowledgements. I would like to thank Michael Goodrich for his research collaboration on authenticated data structures. This work was supported in part by NSF Grant CCR–0098068.

References

1. A. Anagnostopoulos, M. T. Goodrich, and R. Tamassia. Persistent authenticated dictionaries and their applications. In *Proc. Information Security Conference (ISC 2001)*, volume 2200 of *LNCS*, pages 379–393. Springer-Verlag, 2001.
2. A. Buldas, P. Laud, and H. Lipmaa. Accountable certificate management using undeniable attestations. In *ACM Conference on Computer and Communications Security*, pages 9–18. ACM Press, 2000.
3. J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Proc. CRYPTO*, 2002.

4. P. Devanbu, M. Gertz, A. Kwong, C. Martel, G. Nuckolls, and S. Stubblebine. Flexible authentication of XML documents. In *Proc. ACM Conference on Computer and Communications Security*, 2001.
5. P. Devanbu, M. Gertz, C. Martel, and S. Stubblebine. Authentic third-party data publication. In *Fourteenth IFIP 11.3 Conference on Database Security*, 2000.
6. I. Gassko, P. S. Gemmell, and P. MacKenzie. Efficient and fresh certification. In *Int. Workshop on Practice and Theory in Public Key Cryptography (PKC '2000)*, volume 1751 of *LNCS*, pages 342–353. Springer-Verlag, 2000.
7. M. T. Goodrich, J. Lentini, M. Shin, R. Tamassia, and R. Cohen. Design and implementation of a distributed authenticated dictionary and its applications. Technical report, Center for Geometric Computing, Brown University, 2002. http://www.cs.brown.edu/cgc/stms/papers/stms.pdf.
8. M. T. Goodrich, M. Shin, R. Tamassia, and W. H. Winsborough. Authenticated dictionaries for fresh attribute credentials. In *Proc. Trust Management Conference*, volume 2692 of *LNCS*, pages 332–347. Springer, 2003.
9. M. T. Goodrich and R. Tamassia. Efficient authenticated dictionaries with skip lists and commutative hashing. Technical report, Johns Hopkins Information Security Institute, 2000. http://www.cs.brown.edu/cgc/stms/papers/hashskip.pdf.
10. M. T. Goodrich, R. Tamassia, and J. Hasic. An efficient dynamic and distributed cryptographic accumulator. In *Proc. Int. Security Conference (ISC 2002)*, volume 2433 of *LNCS*, pages 372–388. Springer-Verlag, 2002.
11. M. T. Goodrich, R. Tamassia, and A. Schwerin. Implementation of an authenticated dictionary with skip lists and commutative hashing. In *Proc. 2001 DARPA Information Survivability Conference and Exposition*, volume 2, pages 68–82, 2001.
12. M. T. Goodrich, R. Tamassia, N. Triandopoulos, and R. Cohen. Authenticated data structures for graph and geometric searching. In *Proc. RSA Conference—Cryptographers'Track*, pages 295–313. Springer, LNCS 2612, 2003.
13. P. C. Kocher. On certificate revocation and validation. In *Proc. Int. Conf. on Financial Cryptography*, volume 1465 of *LNCS*. Springer-Verlag, 1998.
14. P. Maniatis and M. Baker. Enabling the archival storage of signed documents. In *Proc. USENIX Conf. on File and Storage Technologies (FAST 2002)*, Monterey, CA, USA, 2002.
15. P. Maniatis and M. Baker. Secure history preservation through timeline entanglement. In *Proc. USENIX Security Symposium*, 2002.
16. C. Martel, G. Nuckolls, P. Devanbu, M. Gertz, A. Kwong, and S. Stubblebine. A general model for authentic data publication, 2001. http://www.cs.ucdavis.edu/~devanbu/files/model-paper.pdf.
17. R. C. Merkle. Protocols for public key cryptosystems. In *Proc. Symp. on Security and Privacy*, pages 122–134. IEEE Computer Society Press, 1980.
18. R. C. Merkle. A certified digital signature. In G. Brassard, editor, *Proc. CRYPTO '89*, volume 435 of *LNCS*, pages 218–238. Springer-Verlag, 1990.
19. M. Naor and K. Nissim. Certificate revocation and certificate update. In *Proc. 7th USENIX Security Symposium*, pages 217–228, Berkeley, 1998.
20. D. J. Polivy and R. Tamassia. Authenticating distributed data using Web services and XML signatures. In *Proc. ACM Workshop on XML Security*, 2002.
21. M. Shin, C. Straub, R. Tamassia, and D. J. Polivy. Authenticating Web content with prooflets. Technical report, Center for Geometric Computing, Brown University, 2002. http://www.cs.brown.edu/cgc/stms/papers/prooflets.pdf.
22. R. Tamassia and N. Triandopoulos. On the cost of authenticated data structures. Technical report, Center for Geometric Computing, Brown University, 2003. http://www.cs.brown.edu/cgc/stms/papers/costauth.pdf.