Poster: Zero-Knowledge Authenticated Order Queries and Applications

Esha Ghosh
Dept. Computer Science
Brown University
esha_ghosh@brown.edu

Michael T. Goodrich
Dept. Computer Science
U. California, Irvine
goodrich@uci.edu

Olga Ohrimenko
Microsoft Research
oohrim@microsoft.com

Roberto Tamassia
Dept. Computer Science
Brown University
roberto_tamassia@brown.edu

***Abstract*—Releasing verifiable partial information while maintaining privacy of the rest of the data is a requirement in many practical scenarios. In particular, maintaining an ordered list of elements in a trustworthy and privacy-preserving manner has a number of applications in management of network information and health care data. In this poster, we discuss our efficient, fully dynamic, secure and privacy-preserving mechanisms that allow querying of member and order information on the data stored in lists, trees, or posets of bounded dimension.**

I. Dynamic Privacy-Preserving Authenticated Data Structure Model

Motivated by networking and cloud computing applications, we introduce a formal model of a dynamic privacy-preserving authenticated data structure (DPPADS). It is a three party model where the owner outsources his data structure to a server who answers queries issued by a set of distributed clients. The owner can at any point update the data structure. The server answers queries in such a way that the clients (1) can verify the correctness of the answers but (2) do not learn anything about the data structure besides what can be inferred from the query answers. This poster is based on [1] (to appear in ACNS 2015) and [2].

II. Applications

Our model generalizes data structures where privacy and integrity should be maintained simultaneously. Here, we consider applications, such as the following, that use ordered lists and trees, as well as bounded-dimensional paritally-ordered sets (posets), to store and efficiently query data, where the sensitive nature of data items and potentially malicious behavior of the involved parties require appropriate security measures.

Network Information Management

- *Firewall policies* are often expressed as an ordered list of rule-action pairs [3], [4], $((r_1, a_1), \ldots, (r_n, a_n))$, where if a network packet, p, matches two rules, r_i and r_j, with $i < j$, then the action a_i should be applied, rather than the action a_j. A broad class of *firewall policies* can also be expressed in terms of posets of bounded dimension [5]. The contents and ordering of such firewall policy lists are potentially sensitive from a security perspective, so it is desirable that external rule-comparison queries to such a list are answered without revealing other rules in the list/DAG or even the number, n, of rules.
- In *collaborative filtering* and *reputation management* systems, one maintains an ordered preference list for a set of items (e.g., products), based on popularity or feedback scores. Due to the potential for feedback extortion [6], answers to queries on such lists should be limited to reporting the preference order between two items without revealing relative orderings between other items.
- In *wireless networking* applications, access control can be defined by geo-spatial location, where access policies are defined in terms of rectangular regions [7], [8]. Since rectangle inclusion is a poset of bounded dimension, and access control involves sensitive policies, this work motivates the need for secure, verifiable, private methods for querying partial orders of bounded dimension.

Health Care Information

XML is a common format for managing information including sensitive medical data [9], [10]. Since XML format is human-readable, it has security and privacy concerns and it is desirable to perform verifiable queries on the tree structure of an XML document so that the answer reveals no more information than can be inferred from the answer itself.

Order Statistics

- In *distributed grid computing*, such as folding@home, incentives are provided to the top-k most productive participants. Due to the prevalence of cheating [11], however, the incentive service should ideally prove to a participant that she is the kth most productive without revealing the ranking or relative ordering of the other participants.
- In an auction with a single winner (e.g., online ad auction for a single ad spot) every participant submits her secret bid to the auction organizer. After the top bidder is announced a participant wishes to verify that her bid was inferior. The organizer would then provide a proof without revealing the amount of the top bid, the rank of the participant's bid, or any information about other bids.
- Lenders often require an individual or a couple to prove eligibility for a loan by providing a bank statement and a pay stub. Such documents contain sensitive information beyond what the lender is looking for, such as whether

the bank account balance and salary are above given thresholds. A desirable alternative would be to provide a proof from the bank and employer that these thresholds are met without revealing exact figures and even hiding who of the two spouses earns more.

III. Our Contributions

- Our proposed model for DPPADS is general enough to support queries on any dynamic data structure.
- We give an efficient and provably secure constructions of fully dynamic authenticated lists and trees that support order queries and updates. The latter construction extends to any partial order of bounded dimension. The constructions are based on standard cryptographic assumptions.
- Our constructions have optimal performance in all cost measures, except for a logarithmic overhead on the query time (see Table I).
- We use lightweight cryptographic primitives: hash functions, bilinear map and group operations (exponentiation, multiplication) in prime order groups.

IV. Comparison with Related Work

We compare privacy properties and the asymptotic complexity of our constructions with the existing constructions that are privacy-preserving in Table I. We note that [10], [12], [13] address similar problems but do not satisfy our notion of zero-knowledge privacy. In Table I, where our space efficient implementation (SE-DPPADS) differs from the DPPADS implementation, we denote the complexity of SE-DPPADS with brackets, "[ ]." We provide the only construction that supports fully dynamic zero-knowledge updates (inserts and deletes) and zero-knowledge queries (order and positive membership) with near optimal proof size and complexities for all three parties.

All the time and space complexities in the table are asymptotic. Notation: n is the list size, m is the query size, L is the number of insertions/deletions in a batch, M is the number of distinct elements that have been queried since the last update (insertion/deletion) k is the security parameter. W.l.o.g., we assume list elements are k bits long. Following the standard convention, we omit a (constant) multiplicative factor of $O(k)$ for element size in every cell.

V. Extensions

Our scheme for order queries can be used as a building block to answer efficiently and in zero knowledge (i.e., the returned proofs should be simulatable) many interesting statistical queries. Let the server hold a list $\mathcal{L}$ and $\mathcal{S}$ is a (unordered) subset of $\mathcal{L}$ to which the client has access. The client can request the following queries w.r.t. the order of the elements of $\mathcal{S}$ in the list $\mathcal{L}$ (without querying the order explicitly):

1) *Maximum, Minimum, Median* element of $\mathcal{S}$;
2) *Top* t elements of $\mathcal{S}$;
3) *Elements* in $\mathcal{S}$ that are *above/below threshold* value a.

To reply to these queries in a privacy-preserving manner, the server can simply use our construction for zero-knowledge order proofs. Moreover, the size of the proof returned for each query is proportional to the query size and is optimal for the threshold query.

TABLE I
Comparison of the efficiency of our construction with existing static and dynamic constructions that support privacy-preserving queries in the three party model.

	[14]	[15]	[16]	[1], [2]
ZK Query	✓	✓	✓	✓
ZK Update			✓	✓
Owner's State Size			n	n[1]
Setup time	n^2	n^2	n	n
Storage Space	n^2	n^2	n	n
Order Query time	mn	m		$m \log n$
(Positive) Member Query time	mn	m	m	$m \log n$
Order Verification time	m^2	m^2		m
(Positive) Member Verification time	m^2	m^2	m	m
Proof size	m^2	m^2	m	m
Insertion Time			L	$L + M[L \log n + M]$
Deletion Time				$L + M[L \log n + M]$

References

[1] E. Ghosh, O. Ohrimenko, and R. Tamassia, "Verifiable order queries and order statistics on a list in zero-knowledge," ePrint Report 2014/632 (To appear in ACNS 2015).

[2] E. Ghosh, M. T. Goodrich, O. Ohrimenko, and R. Tamassia, "Fully-dynamic verifiable zero-knowledge order queries for network data," ePrint Report 2015/283.

[3] F. Bukhatwa and A. Patel, "Effects of ordered access lists in firewalls," in *IADIS WWW/Internet International Conference*, 2004.

[4] H. Hamed and E. Al-Shaer, "Dynamic rule-ordering optimization for high-speed firewall filtering," in *ASIACCS*, 2006.

[5] A. Tapdiya and E. Fulp, "Towards optimal firewall rule ordering utilizing directed acyclical graphs," in *ICCCN*, 2009.

[6] M. T. Goodrich and F. Kerschbaum, "Privacy-enhanced reputation-feedback methods to reduce feedback extortion in online auctions," in *CODASPY*, 2011.

[7] M. J. Atallah, M. Blanton, and K. B. Frikken, "Efficient techniques for realizing geo-spatial access control," in *ASIACCS*, 2007.

[8] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, "Dynamic and efficient key management for access hierarchies," *ACM Trans. Inf. Syst. Secur.*, 2009.

[9] J. Brown and D. M. Blough, "Verifiable and redactable medical documents," *AMIA Annu Symp Proc*, 2012.

[10] A. Kundu and E. Bertino, "Structural signatures for tree data structures," *PVLDB*, 2008.

[11] M. T. Goodrich, "Pipelined algorithms to detect cheating in long-term grid computations," *Theoretical Computer Science*, 2008.

[12] E.-C. Chang, C. L. Lim, and J. Xu, "Short redactable signatures using random trees," in *CT-RSA*, 2009.

[13] A. Kundu, M. J. Atallah, and E. Bertino, "Leakage-free redactable signatures," in *CODASPY*, 2012.

[14] C. Brzuska, H. Busch, Ö. Dagdelen, M. Fischlin, M. Franz, S. Katzenbeisser, M. Manulis, C. Onete, A. Peter, B. Poettering, and D. Schröder, "Redactable signatures for tree-structured data: Definitions and constructions," in *ACNS*, 2010.

[15] K. Samelin, H. C. Poehls, A. Bilzhause, J. Posegga, and H. De Meer, "Redactable signatures for independent removal of structure and content," in *ISPEC*, 2012.

[16] H. C. Pöhls and K. Samelin, "On updatable redactable signatures," in *ACNS*, 2014.