EFFICIENT AND STRONG MUTUAL AUTHENTICATION SCHEME USING CSTA FOR SECURE DATA TRANSMISSION

Brinta Babu[1] | T.B Dharmaraj[2]
[1](UG Student, Christ the King Engineering College, brintababu142@gmail.com)
[2](Professor, Christ the King Engineering College, bellidharmaraj@gmail.com)

Abstract— *Among all type of computations, the polynomial function evaluation is a fundamental, yet an important one due to its wide usage in engineering and scientific problems. In this paper, we investigate publicly verifiable outsourced computation for polynomial evaluation with the support of multiple data sources. Our proposed scheme is universally applicable to all types of polynomial computation and allows the clients to outsource new data at any time. While the existing solutions only support the verification of polynomial evaluation over a single data source, i.e, all the inputs of the polynomial function are outsourced and signed by a single entity, our solutions support polynomial evaluations over multiple different data sources, which are more common and have wider applications.,e.g., to assess the city air pollution one needs to evaluate the environmental data uploaded from the multiple environmental monitoring centers. The security issue has become an important concern of networks. To prevent the grid resources from being illegally visited, the strong mutual authentication is needed for both user and the server. In recent periods, many password based user authentication schemes are proposed for solving authentication issues. However, most of them are not ideal for networking, since they do not provide the strong mutual authentication. It is proposed to introduce an efficient user Cyclic Shift Transposition Algorithm(CSTA) scheme and model for secure routing mechanism of sharing messages between the source and the destinations in networking by adding information about sender and preserves the message content, which prevents not only known attack but also maintains the integrity of data*

Keywords— Polynomial Verification; Authentication; Secured Outsourcing

1. INTRODUCTION

As cloud computing provide affordable and scalable computation and storage resources, outsourcing computation to the cloud become an unavoidable trend nowadays. Among all types of computations , the polynomial evaluation is considered to be fundamental, yet an important one, and it has been widely used in the engineering and scientific problems

In cloud computing the cloud server executes the polynomials over the personal health data uploaded from various wearable devices to evaluate the personal health and make suggestions for people to keep healthy. However, while enjoying all the benefits of cloud the users also lose the control of their computation in cloud. Due to the cloud's possible misbehaviors motivated by the monetary reasons or caused by the hacking or system failure, the client would like to verify the correctness of the computation result output by the cloud. According to a verifiable computation scheme for the delegated polynomial should satisfy the following requirements: (1) security, meaning that the cloud server can prove the correctness of the delegated computation of the polynomial function f, and the client can correctly verify the result of the function f by checking the proof message(2) efficiency, meaning that the client should be able to verify the result with communication and computation costs significantly lower than the cost of computing f locally; (3) input-independent efficiency, meaning that the verifying time is independent of the size of the inputs of the delegated polynomial function; (4) unbound storage, meaning that the client is able to outsource new input data to the cloud for polynomial evaluations; (5) not fixed polynomial functions, meaning that neither the delegated functions should be fixed nor the client is required to know the functions before outsourcing the data. To the best of our knowledge, the authors in proposed the first protocol which satisfies the above requirements. From the practical perspective, however, it still has some limitations which make it not practical in real-world application scenarios. First of all, the scheme of does not support public verification. It has an implicit assumption that the data contributor and the computation requestor are the same entity (or the data owner has to share the private key with the computation requestor). But, it is not always true in practice, where the computation requestor is most likely to be a third-party entity. The second limitation is that the inputs of the polynomial must come from one single data source. In real-world applications, however, the input data of the polynomial may come from multiple contributors. Finally, the communication cost of the solution in depends on the polynomial size, which raises the scalability concern in practice. Take the computation of the air pollution level as an example, there are many air quality monitor sites collecting the environmental pollution data and uploading these data to the cloud in the fixed time interval.
Thus, the computation is expected to be conducted by a third party agency over the environmental pollution data uploaded from multiple monitors (different data sources). At last, the air pollution information will be released to the public by the agency. To verify the correctness of the result of the delegated computation over the outsourced data

from multiple contributors, a publicly verifiable computation scheme for polynomial functions is highly

required. To address this problem, we define three key requirements for a practically verifiable computation design, which are lacking in existing solutions: 1) Public verification. It means that any entity can verify the correctness of the result of the delegated polynomial function even if the inputs are not uploaded or signed by itself; 2) The Support of multiple data sources. It means that the inputs of the polynomial can be contributed by multiple independent data sources; 3) Function-independent bandwidth efficiency. It means that the communication cost should be independent of the complexity of the delegated function, i.e., the overhead of communication for a verification task is constant. In this paper, we propose a novel and efficient publicly verifiable computation scheme for the delegated polynomial evaluation. The key characteristic of our design is that the proposed scheme is homomorphically verifiable for any general polynomial function. We formally prove the security of our scheme based on the Computational Diffie-Hellman (CDH) assumption and evaluate its efficiency in terms of the computation and the communication costs through extensive experiments. The main contributions in this paper can be summarized as follows.

We, for the first time, model the problem of publicly verifiable delegated computation for the polynomial functions and identify three new practical requirements: public verification, the support of multiple data contributors, and function-independent efficiency. We propose an efficient and publicly verifiable outsourcing computation scheme to meet the above key requirements

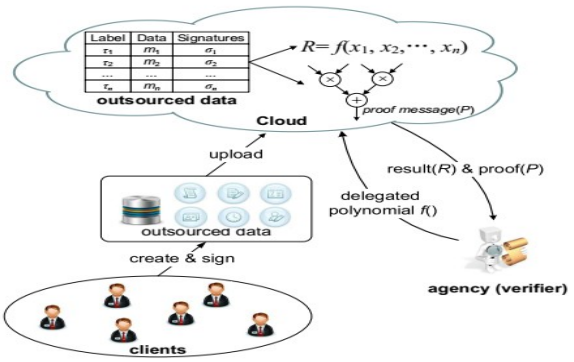
.To achieve the high bandwidth efficiency, we design a homomorphic verifiable computation tag structure for the delegated polynomial function, by which our verifiable computation scheme can achieve the constant communication cost and scales well in practice. We formally prove the correctness and the soundness of our scheme under the well-defined CDH assumption. We also implement a prototype and carry out a series of evaluation studies. The evaluation results further validate the effectiveness and efficiency of our scheme.

2. RELATED WORKS

2.1.VERIABLE DELEGATION OF COMPUTATION ON OUTSOURCED DATA

We address the problem in which a client stores a large amount of data with an untrusted server in such a way that, at any moment, the client can ask the server to compute a function on some portion of its outsourced data. In this scenario, the client must be able to efficiently verify the correctness of the result despite no longer knowing the inputs of the delegated computation, it must be able to keep adding elements to its remote storage, and it does not have to fix in advance (i.e., at data outsourcing time) the functions that it will delegate. Even more ambitiously, clients should be able to verify in time independent of the input-size – a very appealing property for computations over huge amounts of data. In this work we propose novel cryptographic techniques that solve the above problem for the class of computations of quadratic polynomials over a large number of variables. This class covers a wide range of significant arithmetic computations – notably, many important statistics. To confirm the efficiency of our solution, we show encouraging performance results, e.g., correctness proofs have size below 1 kB and are verifiable by clients in less than 10 milliseconds. The only approach that comes close to achieving requirements (1)–(5) is the work by Chung et al. on memory delegation. The authors propose a scheme based on techniques from which exploit the power of the PCP theorem. With this scheme, a client can delegate a broad class of computations over its outsourced memory fulfilling the requirements from above (except for verification efficiency, which requires time log n, instead of constant time). While providing a satisfying solution in theory, this approach suffers from the usual impracticality issues of genera lpurpose PCP techniques and hence does not lead to truly practical solutions to the problem. In summary, in this work we focus on the important problem of delegating computations over data which continuously grows and is outsourced to remote servers. This specific problem has not received much attention so far: the only existing solution, though very general, does not seem to lead to efficient implementations. In contrast, we propose a protocol that achieves all the desired requirements for a restricted, yet practical and useful, class of computations, and has the advantage of achieving performances that are promising for a practically efficient solution

2.2 SYSTEM ARCHITECHTURE



2.3 CERTIFICATE REVOCATION AND CERTIFICATE UPDATE

A new solution is suggested for the problem of certificate revocation. This solution represents Certificate Revocation Lists by an authenticated search data structure. The process of verifying whether a certificate is in the list or not, as well l as updating the list, is made very efficient. The suggested solution gains in scalability, communication costs, robustness to parameter changes and update rate. Comparisons to the following solutions are included: `traditional' CRLs (Corticated Revocation Lists), Micali's Certificate Revocation System (CRS) and Kocher's Certificate Revocation Trees (CRT). Finally, a scenario in which certificates are not revoked, but frequently issued for short-term periods is considered. Based on the authenticated search data structure scheme, a certificate

update scheme is presented in which all certificates are updated by a com- mon message. The suggested solutions for certificate revocation and certificate update problems is better than current solutions with respect to communication costs, update rate, and robustness to changes in parameters and is compatible e.g. with X.500 certificates.Certification authority (CA): A trusted party, already having a certified public key, responsible for establishing and vouching for the authenticity of public keys, including the binding of public keys to users through certificates and certificate revocation. A CA does not provide on-line certificate information services to users. Instead, It updates a directory on a periodic basis. A CA issues certificates for users by signing a message containing the certificate serial number, relevant data and an expiration date. The certificate is sent to a directory and/or given to the user himself. The CA may revoke a certificate prior to its expiration date. 2. Directory: One or more non-trusted parties that get updated certificate revocation information from the CA and serve as a certificate database accessible by the users. User: A non-trusted party that receives its certificate from the CA, and issues queries for certificate information. A user may either query the validity of other users' certificates (we denote users that query other users' certificates as merchants) or, get a proof of the validity of his certificate in order to present it with his certificate (for the latter, the proof must be transferable).

2.4AUTHENTICATED DATA STRUCTURES FOR GRAPH AND GEOMATRIC SEARCHING

Following in the spirit of data structure and algorithm correctness checking, authenticated data structures provide cryptographic proofs that their answers are as accurate as the author intended, even if the data structure is being maintained by a remote host. We present techniques for authenticating data structures that represent graphs and collection of geometric objects. We use a model where a data structure maintained by a trusted source is mirrored at distributed directories, with the directories answering queries made by users. When a user queries a directory, it receives a cryptographic proof in addition to the answer, where the proof contains statements signed by the source. The user verifies the proof trusting only the statements signed by the source. We show how to efficiently authenticate data structures for fundamental problems on networks, such as path and connectivity queries, and on geometric objects, such as intersection and containment queries. Our work has applications to the authentication of network management systems and geographic information systems. In this paper we are interested in studying a new dimension in data structure and algorithm checking— how can we design sophisticated data structures and algorithms so that their responses can be verified as accurately as if they were coming from their author, even when the response is coming from an untrusted host? Examples of the kind of information we want to authenticate include dynamic documents, online catalog entries, and the responses to queries in geographic information systems, financial databases, medical information systems, and scientific databases. Digital signatures can be used to verify simple static documents, but are inefficient for dynamic data structures. We therefore need new techniques for authenticating data structures. The main challenge in providing an integrity service in such contexts is that the space of possible answers is much larger than the data size itself. For example, there are O(n2) different paths in a tree of n nodes, and each of these paths can have O(n) edges. Requiring an authenticator to digitally sign every possible response is therefore prohibative, especially when the data is changing due to the insertion or deletion of elements in the set. Ideally, we would like our authenticator to sign just a single digest of our data structure, with that digest being built from the careful combination of cryptographic hashes of subsets of our data. If we can achieve such a scheme, then verifying the answer to a query in our data base can be reduced to the problem of collecting the appropriate partial hashes for a user to recompute the digest of the entire structure and compare that to the digest signed by the authenticator. Even when we follow this approach, however, we are faced with the challenge of how to subdivide the data in a way that allows for efficient assembly for any possible query. For simple data structures, such as dictionaries, this subdivision is fairly straightforward (say using a linear ordering and a Merkle hash tree.

2.5 A GENERAL MODEL FOR AUTHENTICATED DATA STRUCTURE

Query answers from on-line databases can easily be corrupted by hackers or malicious database publishers. Thus it is important to provide mechanisms which allow clients to trust the results from on-line queries. Authentic publication allows untrusted publishers to answer securely queries from clients on behalf of trusted off-line data owners. Publishers validate answers using hard-to-forge verification objects VOs), which clients can check efficiently. This approach provides greater scalability, by making it easy to add more publishers, and better security, since on-line publishers do not need to be trusted. To make authentic publication attractive, it is important for the VOs to be small, efficient to compute, and efficient to verify. This has lead researchers to develop independently several different schemes for efficient VO computation based on specific data structures. Our goal is to develop a unifying framework for these disparate results, leading to a generalized security result. In this paper we characterize a broad class of data structures which we call Search DAGs, and we develop a generalized algorithm for the construction of VOs for Search DAGs. We prove that the VOs thus constructed are secure, and that they are efficient to compute and verify. We demonstrate how this approach easily captures existing work on simple structures such as binary trees, multi-dimensional range trees, tries, and skip lists. Once these are shown to be Search DAGs, the requisite security and efficiency results immediately follow from our general theorems. Going further, we also use Search DAGs to produce and prove the security of authenticated versions of two complex data models for efficient multi-dimensional range searches. This allows

efficient VOs to be computed (size O(log N + T)) for typical one- and two-dimensional range queries, where the query answer is of size T and the database is of size N. We also show I/O-efficient schemes to construct the VOs. For a system with disk blocks of size B, we answer one-dimensional and three-sided range queries and compute the VOs with O(logB N + T/B) I/O operations using linear size data structures. The use of an untrusted publisher reduces the risks of operating a secure on-line system: an attacker who gains control of a specific publisher would not be able to fool clients, who would reliably detect incorrect answers and simply find another publisher. It also allows graceful scaling by adding additional publishers to meet increasing demand from clients. Note that there are no secrets in this scheme: an adversary trying to fool the client is assumed to know all the data, the hash function, and all the digest values. Thus there is no privileged information to be compromised. Note, however, that this approach is currently only practical when the data is relatively static.

2.6 AUTHENTICATED RELATIONALTABLES AND AUTHENTICATED SKIP LISTS

We present a general method, based on the usage of typical DBMS primitives, for maintaining authenticated relational tables. The authentication process is managed by an application external to the DBMS, that stores just one hash information of the authentication structure. The method exploits techniques to represent hierarchical data structures into relational tables and queries that allow an efficient selection of the elements needed for authentication. We have described methods that allow a user to verify the authenticity and completeness of simple queries results, even if the database system is not trusted. The overhead for the user is limited at storing only a single hash value. Our work is the first to design and evaluate techniques for authenticated skip list that are appropriate to a relational database, and the first to prove the feasibility of authenticated skip list for integrity of databases. The security of the presented method is based on the reliability of ADSes. There are many works in the literature that demonstrate that the security of ADS is based on the difficulty to find useful collisions in a cryptographic hash function. So all the security relies on the effectiveness of hash functions. The prototype used for the experiments uses commutative hashing. In it is demonstrated that commutative hashing does not augment the possibility to find a collision in the used hash function. In the future we would like to investigate how to authenticate more complex queries making use of a larger set of relational operations. Further, we would like to study models to build integrity verification services in peer to peer systems

2.7 TIME AND SPACE EFFICIENT ALGORITHMS FOR TWO-PARTY AUTHENTICATED DATA STRUCTURES

Authentication is increasingly relevant to data management. Data is being outsourced to untrusted servers and clients want to securely update and query their data. For example, in database outsourcing, a client's database is stored and maintained by an untrusted server. Also, in simple storage systems, clients can store very large amounts of data but at the same time, they want to assure their integrity when they retrieve them. In this paper, we present a model and protocol for two party authentication of data structures. Namely, a client outsources its data structure and verifies that the answers to the queries have not been tampered with. We provide efficient algorithms to securely outsource a skip list with logarithmic time overhead at the server and client and logarithmic communication cost, thus providing an efficient authentication primitive for outsourced data, both structured (e.g., relational databases) and semi-structured (e.g., XML documents). In our technique, the client stores only a constant amount of space, which is optimal. Our two-party authentication framework can be deployed on top of existing storage applications, thus providing an efficient authentication service. Finally, we present experimental results that demonstrate the practical efficiency and scalability of our scheme. In this paper, we have presented an efficient protocol for two-party authentication based on cryptographic hash functions. Namely, we have given efficient, lightweight and provably secure algorithms that ensure the validity of the answers returned by an outsourced dictionary. We have implemented our protocol and we have provided experimental results that confirm the scalability of our approach. As future work, we envision developing a general authentication framework that can be applied to other data structures, such as dynamic trees . Additionally, we could investigate realizations of two-party authenticated protocols based on other cryptographic primitives, (for example cryptographic accumulators.

"Optimal verification of operatioNs on dynamic sets

We study the verification of set operations in the model of authenticated data structures, namely the problem of cryptographically checking the correctness of outsourced set operations performed by an untrusted server over a dynamic collection of sets that are owned (and updated) by a trusted source. We present a new authenticated data structure scheme that allows any entity to publicly verify the correctness of primitive sets operations such as intersection, union, subset and set difference. Based on a novel extension of the security properties of bilinear-map accumulators as well as on a primitive called accumulation tree, our authenticated data structure is the first to achieve optimal verification and proof complexity (i.e., only proportional to the size of the query parameters and the answer), as well as optimal update complexity (i.e., constant), and without bearing any extra asymptotic space overhead. Queries (i.e., constructing the proof) are also efficient, adding a logarithmic overhead to the complexity needed to compute the actual answer. In contrast, existing schemes entail high communication and verification costs or high storage costs as they recompute the query over authentic data or precompute answers to all possible queries. Applications of interest include efficient verification of keyword search and database queries. We base the security of our constructions on the bilinear q-strong Diffie-Hellman assumption.In this paper, we presented an authenticated data structure for the optimal verification of set operations. The achieved efficiency is

mainly due to new, extended security properties of accumulators based on pairing-based cryptography. Our solution provides two important properties, namely public verifiability and efficiency, as opposed to the outsourced verifiable computation model. A natural question to ask is whether outsourced verifiable computation with secrecy, public verifiability and efficiency exists. Analogously, which other specific functionalities can be optimally and publicly verified? Finally, according to a recently proposed definition of optimality , our construction is nearly optimal—only verification costs (hence the title) and updates are optimal. It is interesting to explore whether an optimal authenticated sets collection data structure is possible, i.e., one that asymptotically matches the bounds of the plain sets collection data structure, reducing the query time from O(N log2 N) to O(N).

3. PUBLICLY VERIFIABLE COMPUTATION

SetUp : is run by the cloud server to initialize the system. It takes a security parameter λ as input and outputs the global security parameters for the system.
KeyGen(1κ) (pk; sk) is run by a new client. It takes the security parameter κ as input and returns the public key pk and private key sk for the new client.

4. DATA FORWARDING

The sender affords the input text for encryption in the text operation or file operation. Initially, the algorithm accomplishes the partition operation by splitting the sentences into a variable sized word (16 bit) and then into four partition that is to say; in other words into four character. Then, arrange them in a matrix format to perform various shifting operation. Following this, it maps the sequence into a block having four columns and N/4 rows.

5. ROW AND DIAGONAL SHIFT

It performs the shift operation 1(Column shift) in a certain specified order to the resulting symbol block needed. Subsequently, it performs the Shift Operation 2 (Row Shift). Each Row has to be shifted from left to right into a certain number of times. Later on, it perform shift operation3 (Primary Diagonal Shift Operation) Each element of the primary diagonal must be shifted to a definite number of times from top to bottom. Then, perform the shift operation 4(Secondary Diagonal Shift Operation) in which each element of the secondary diagonal must be shifted to a certain number of times from top to bottom. This encrypted form of text is the cipher text which should be sent to the receiver by the sender.

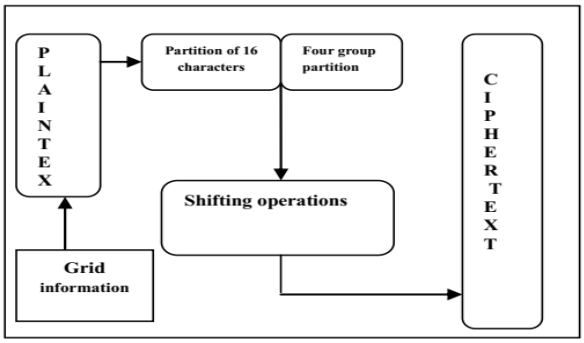
6. DATA ENCRYPTION

The encryption process consists of partition operation and then it is subjected to shifting operation and thus it obtains the cipher text. The cipher text is converted to the original plain text by the receiver by performing the identical reverse process. In partition operation the block of the plain text is subjected to splitting with variable sized word and then subjected to shifting operation.

7. CSTA ENCRYPTIONN AND DECRYPTION

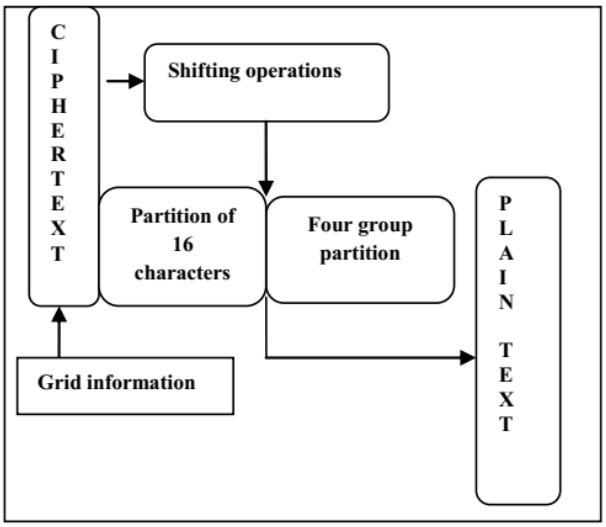
7.1 ENCRYPTION

Step1: Map the Text sequence S into blocks of size NxN.
Step2: Perform shift column in a certain order specified (CS [xxxx]).
Step3: Perform shift row in a certain order specified (RS [xxxx]).
Step4: Perform prime Diagonal shift in a certain order specified D1[x].
Step5: Perform secondary Diagonal shift in a certain order specified D2[x].
Step6: Represent the outcome in a linear order to get the encrypted text.



7.2 DECRYPTION

Split the cipher text into blocks of size NxN.
Step1: Perform Secondary Diagonal shift in an order carried out.
Step2: Perform prime Diagonal shift in an order carried out in a certain order specified.
Step3: Perform Row shift in a certain order specified.
Step4: Perform Column shift in an certain order specified.
Step5: Formulate the outcome in a linear order to get the decrypted text.



8. CONCLUSION

The proposed CSTA algorithm is implemented over text file based on word and group of words. The file format consists of two processes. The

sender affords the input text for encryption in the text operation. Initially, the algorithm accomplishes the partition operation by splitting the variable sized word (16bit) into four partition that is to say; in other words into four character. Then, arrange them in a matrix format to perform various shifting operation. Following this, it maps the sequence into a block having four columns and N/4 rows. Afterwards, perform the shift operation 1(Colum shift) in a certain specified order to the resulting symbol block needed. Subsequently, it performs the Shift Operation 2 (Row Shift). Each row has to be shifted from left to right into a certain number of times. Later on it perform shift operation3 (Primary Diagonal Shift Operation) Each element of the primary diagonal must be shifted to a definite number of times from top to bottom. Then, perform the shift operation 4(Secondary Diagonal Shift Operation) in which each element of the secondary diagonal must be shifted to a certain number of times from top to bottom. This encrypted form of text is the cipher text which should be sent to the receiver by the sender. The receiver affords the cipher text for decryption in the text operation. This decryption is performed using the identical shifting operation and then supposed to perform the partition operation to perform the identical cyclic shift transposition algorithm. The decryption is the turn back process encryption and hence it obtains the plain text. In the file format, the sender side uploads the demanded files. The sender can upload n‟‟ number of files. Then the sender side executes the encryption operation by employing the cyclic shift transposition algorithm and hence forms the cipher text. This algorithm carries out the partition operation by splitting the sentence in the files into variable sized word (16 bit), the empty space is also counted. Following this, it is subjected to shifting operation and therefore it forms the cipher text. This cipher text is received by the receiver. The cipher text received by the receiver requires being decrypted to incur the original plain text. This decryption is performed using the identical shifting operation and then supposed to perform the partition operation to perform the identical cyclic shift transposition algorithm. The decryption is the turn back process of encryption and hence it obtains the plain text.

REFERENCES

[1] M. Backes, D. Fiore, and R. M. Reischuk, “Veriable delegation of computation on outsourced data,” in Proc. of CCS’13. ACM, 2013, pp. 863–874.

[2] M. Naor and K. Nissim, “Certificate revocation and certificate update,” IEEE Journal on Selected Areas in Communications, vol. 18, no. 4, pp. 561–570, 2000.

[3] M. T. Goodrich, R. Tamassia, N. Triandopoulos, and R. Cohen, “Authenticated data structures for graph and geometric searching,” in Proc. of RSA Conference on the Cryptographers’ Track. Springer, 2003, pp. 295–313.

[4] C. Martel, G. Nuckolls, P. Devanbu, M. Gertz, A. Kwong, and S.G. Stubblebine, “A general model for authenticated data structures,” Algorithmica, vol. 39, no. 1, pp. 2004.

[5] G. D. Battista and B. Palazzi, “Authenticated relational tables and authenticated skip lists,” in Proc. of DBSec. Springer, 2007, pp. 31–46.

[6] C. Papamanthou and R. Tamassia, “Time and space efficient algorithm for two-party authenticated data structures,” in Proc. of ICICS. Springer, 2007, pp. 1–15.

[7] C. Papamanthou, R. Tamassia, and N. Triandopoulos, “Optimal verification of operatios on dynamic sets,” in Proc. of CRYPTO. Springer, 2011, pp. 91–110.

[8] R. Gennaro, C. Gentry, and B. Parno, “Non-interactive verifiable computing: Outsourcing computation to untrusted workers,” in Proc. Of CRYPTO. Springer, 2010, pp. 465–482.

[9] B. Parno, M. Raykova, and V. Vaikuntanathan, “How to delegate and verify in public: Verifiable computation from attribute-based encryption,” in Proc. of TCC. Springer, 2012, pp. 422–439.