IPHV8I2003X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .II
ISSN Online: 2319 – 9253
Print: 2319 – 9245

AN EFFICIENT AND SECURE DYNAMIC PROVABLE DATA POSSESSION FOR CLOUD STORAGE

[#1]PEDDI NAVITHA, M.Tech Student,

[#2]K.SADANANDAM, Assistant Professor,

Department Of CSE,

SREE CHAITANYA INSTITUTE OF TECHNOLOGICAL SCIENCES,KARIMNAGAR,T.S, INDIA.

ABSTRACT: Gradually more and more organizations are opting for outsourcing data to remote cloud service providers (CSPs). Clients can rent the CSPs storage infrastructure to store and get back almost infinite amount of data by paying amount per month. On behalf of an improved level of scalability, availability, and durability, some clients may want their data to be virtual on multiple servers across multiple data centers. The more copies the CSP is asked to store, the more amounts the clients are charged. As a result, clients need to have a strong assurance that the CSP is storing all data copies that are decided upon in the service contract, and all these copies are reliable with the most recent modifications issued by the clients. Map-based provable multi copy dynamic data possession (MB-PMDDP) method is being proposed in this paper and consists of the following features: 1) it affords an proof to the clients that the CSP is not corrupt by storing less copies; 2) it supports outsourcing of dynamic data, i.e., it supports block-level functions, such as block alteration, addition, deletion, and append; and 3) it permits official users to effortlessly access the file copies stored by the CSP. In addition, we discuss the security against colluding servers, and discuss how to recognize corrupted copies by a little revising the projected scheme.

Key words: *Cloud computing, dynamic environment, data duplication, outsourcing data storage.*

I.INTRODUCTION

Outsourcing data to a remote cloud service provider (CSP) permits society to store additional data on the CSP than on private computer systems. Such Out sourcing of data storage allows society to focus on improvement and relieves the load of constant server updates and other computing matter. On one occasion the data has been outsourced to a remote CSP which may not be dependable, the data owners drop the direct control over their confidential data. This need of control raises new difficult and demanding tasks connected to data confidentiality and integrity protection in cloud computing. The confidentiality issue can be feeling by encrypting confidential data before outsourcing to remote servers. As such, it is a vital demand of customers to have strong proofs that the cloud servers still have their data and it is not being corrupt with or partially deleted over time. As a result, many researchers have payed attention on the problem of provable data possession (PDP) and proposed different systems to review the data stored on remote servers. PDP is a method for authenticating data integrity over remote servers. In a typical PDP model, the data owners produce some metadata for a data file to be used later for verification purposes through a challenge-response protocol with the remote/cloud server. The owner sends the file to be stored on a remote server which may be untrusted, and erases the local copy of the file. One of the core design ethics of outsourcing data is to provide dynamic behavior of data for a variety of applications. This means that the slightly stored data can be not only accessed by the authorized users, but also efficient and scaled Examples of PDP constructions that deal with dynamic data [10]-[14]. The final are how-ever for a single copy of the data file. PDP method has been obtainable for multiple copies of static data [15]–[17]. PDP system directly deals with multiple copies of dynamic data. When proving multiple data copies, generally system integrity check fails if there is one or more corrupted copies were present. To deal with this issue and recognize which copies have been corrupted, a slight modification has been applied to the proposed scheme.

II.RELATED WORK

Our contributions can be review as follows: i) We propose a map-based provable multi-copy dynamic data possession (MB-PMDDP) method. This method provides an sufficient guarantee that the CSP stores all copies that are agreed upon in the service contract. Additionally, the method supports outsourcing of dynamic data, i.e., it supports blocklevel functions such as block alteration, insertion, removal, and append. The certified users, who have the right to access the owner’s file, can effortlessly access the copies received from the CSP. ii)A thorough comparison of MB-PMDDP with a reference scheme, which one can obtain by expanding existing PDP models for dynamic single-copy data. iii)We show the security of our system against colluding servers, and talk about a slight alteration of the proposed scheme to identify corrupted copies. comment 1: Proof of retrievability (POR) is a balancing approach to PDP, and is stronger than PDP in the sense that the verifier can rebuild the entire file from answers that are consistently transmitted from the server. This is due to encoding of the data file, for example using erasure codes, before outsourcing to remote servers. A range of POR systems can be found in the journalism, for example [18]–[23], which focuses on static data . In this work, we do not instruct the data to be outsourced for the following reasons. Primarily, we are dealing with dynamic data, and therefore if the data file is encoded before outsourcing, modifying a portion of the file needs re-encoding the data file which may not be suitable in practical applications due to high calculation transparency. Secondly, we are allowing for economically-motivated CSPs that may challenge to use less storage than essential by the service agreement through deletion of a few copies of the file. The CSPs have approximately no economic benefit by removing only a small portion of a copy of the file. Thirdly, and more significantly, unlike removal codes, duplicating data files transversely multiple servers attains scalability which is a basic client constraint in CC systems. A file that is duplicated and stored deliberately on multiple servers – situated at various geographic locations – can help decrease access time and communication cost for users. In addition, a server's copy can be rebuilt even from a whole damage using duplicated copies on other servers.

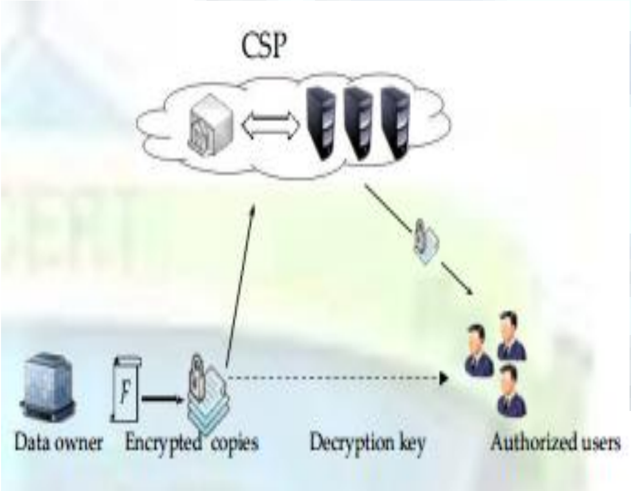

Figure: 1 System Architecture

A. System Components

The cloud computing storage model measured in this work includes three main components as illustrated in Fig. 1:

(i) A data owner that can be an organization initially possessing confidential data to be stored in the cloud.

(ii) A CSP who handles cloud servers (CSs) and offers paid storage space on its infrastructure to store the owner's files.

(iii) Authorized users — a set of owner's clients who have the right to access the remote data. The storage model used in this work can be assumed by much practical requests. For example, e-Health applications can be predicted by this model where the patients' database that includes large and confidential information can be stored on the cloud servers. In these types of applications, the e-Health organization can be measured as the data owner, and the physicians as the approved users who have the right to access the patients' medical history. Many other practical applications like financial, scientific, and educational applications can be observed in similar settings.

B. Outsourcing, Updating, and Accessing

The data owner has a file F consisting of m blocks and the CSP offers to store n copies { F1,F2, ………..,Fn} of the Owner's file on different servers — to prevent simultaneous failure of all copies — in exchange of pre-specified fees in the form of GB/month. The number of copies depends on the nature of data; more copies are desired for critical data that cannot easily be replicated, and to attain a higher level of scalability. This critical data be supposed to be replicated on multiple servers across multiple data centers. On the other hand, non-critical, reproducible data are stored at compact levels of redundancy. The CSP cost model is linked to the number of data copies. For data privacy, the owner encrypts their data before outsourcing to CSP. After outsourcing all n copies of the file, the owner may work together with the CSP to carry out block-level functions on all copies. These functions contains alter, insert, append, and remove specific blocks of the outsourced data copies. An authorized user of the outsourced data throws a data-access request to the CSP and accepts a file copy in an encrypted form that can be decrypted using a secret key shared with the owner. According to the load balancing device used by the CSP to arrange the work of the servers, the data-access demand is directed to the server with the lowest jamming, and as a result the user is not conscious of which copy has been received. We imagine that the communication between the owner and the official users to authenticate their identities and share the secret key has previously been completed.

C. Threat Model

The integrity of customers' data in the cloud may be at danger due to the following reasons. Firstly, the CSP — whose goal is probable to make a profit and sustain a reputation — has an reason to hide data loss (due to hardware failure, management errors, various attacks) or get back storage by removing data that has not been or is rarely accessed. Secondly, a dishonest CSP may store less copies than what has been decided upon in the service contact with the data owner, and try to induce the owner that all copies are correctly stored intact. Thirdly, to save the computational resources, the CSP may totally pay no attention to the data update requests concerned by the owner, or not execute them on all copies leading to inconsistency between the file copies. The objective of the proposed scheme is to identify (with high probability) the CSP misconduct by validating the number and integrity of file copies.

2.1 MB-PMDDP SCHEME

A. Overview and Rationale

produce unique differentiable copies of the data file is the core to design a provable multi-copy data possession scheme. Identical copies enable the CSP to simply deceive the owner by storing only one copy and pretending that it stores multiple copies. Using a simple yet efficient way, the proposed scheme generates distinct copies utilizing the diffusion property of any secure encryption scheme. The diffusion property ensures that the output bits of the cipher text depend on the input bits of the plaintext in a very complex way, i.e., there will be an unpredictable complete change in the cipher text, if there is a single bit change in the plaintext [24]. The interaction between the authorized users and the CSP is considered through this methodology of generating distinct copies, where the former can decrypt/access a file copy received from the CSP. In the proposed scheme, the authorized users need only to keep a single secret key (shared with the data owner) to decrypt the file copy, and it is not necessarily to recognize the index of the received copy. In this work, we propose a MB-PMDDP scheme allowing the data owner to update and scale the blocks of file copies outsourced to cloud servers which may be untrusted. Validating such copies of dynamic data requires the knowledge of the block versions to ensure that the data blocks in all copies are consistent with the most recent modifications issued by the owner. Furthermore, the verifier should be aware of the block indices to guarantee that the CSP has inserted or added the new blocks at the requested positions in all copies. To this end, the proposed scheme is based on using a small data structure (metadata), which we call a map version table.

B. Map-Version Table

The map-version table (MVT) is a small dynamic data structure accumulates on the verifier side to authenticate the reliability and uniformity of all file copies outsourced to the CSP. The MVT consists of three columns: serial number (SN), block number (BN), and block version (BV). The SN is an indexing to the file blocks. It point out the physical position of a block in a data file. The BN is a counter used to make a logical numbering/indexing to the file blocks. Therefore, the relation between BN and SN can be observed as a mapping between the logical number BN and the physical position SN. The BV specifies the current version of file blocks. When a data file is originally created the BV of each block is 1. If a specific block is being updated, its BV is incremented by 1. comment 2: It is significant to note that the verifier remain only one table for infinite number of file copies, i.e., the storage condition on the verifier side does not depend on the number of file copies on cloud servers. For n copies of a data file of size | G|, the storage condition on the CSP side is O(n| G|), while the verifier's overhead is O(m) for all file copies (m is the number of file blocks). Comment 3: The MVT is applied as a linked list to make simpler the insertion deletion of table entries. For actual achievement, the SN is not needed to be stored in the table; SN is considered to be the entry/table index, i.e., each table entry contains just two integers BN and BV (8 bytes). As a result, the total table size is 8m bytes for all file copies. We additionally note that even if the table size is linear to the file size, in practice the previous would be smaller by several orders of magnitude. For instance, outsourcing infinite number of file copies of a 1GB-file with 16KB block size requires a verifier to keep MVT of only 512KB (< 0.05% of the file size).

III. PROPOSED WORK

The above PDP schemes need client initiations for data integrity checking. Also, in case of company oriented environment, it will be useful if the system is tracking the data integrity verification transactions for future enhancements. The model can be designed to achieve self initiated approach of PDP and log-based approach can be used for administration purpose. According to above PDP schemes, the proposed system will help client to maintain the data integrity verification records for further proceedings and also client will be freed from initiating the data integrity checking process. In detail, we will design a system where there will be a timer which will keep generating interrupts and due to these interrupts the data integrity verification request will get generated on behalf of the client. The request then will be served by cloud server as normal PDP approach and will return the proof back to client. The figure (Fig.2) depicts an architecture of proposed system where the client is equipped with timer and has access to the log file generated by verifier module. Our scheme will compare this proof to verify and the result will be saved on permanent storage as like a log file. The client can then periodically check the log file to analyse the request responses made and can assure data integrity.

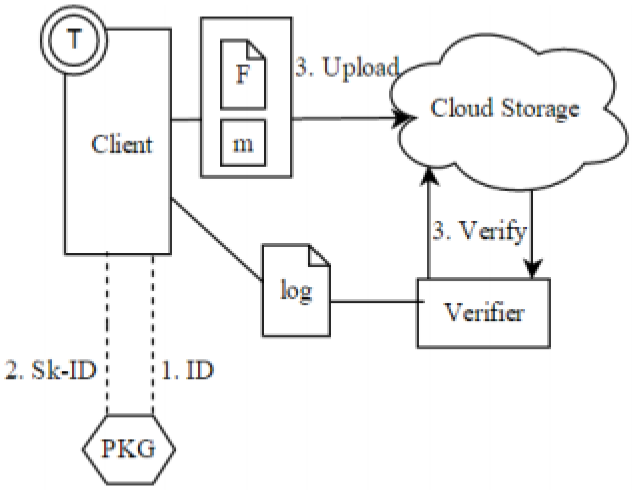

Fig.2 Proposed System Architecture

IV. VARIOUS PDP MODELS

A. Cooperative Provable Data Possession

The parallel computing can be implemented in several ways of computing like instruction level, task and data parallelism. The data verification techniques like PDP perform slower in case of large volume of data. In such situations, the data integrity verification can be done in parallel and data storages can be on multiple clouds. The YanZhu, et al. [2] proposed Cooperative PDP model, which is based on zero knowledge proof mechanism and interactive proof system to prove the integrity of data stored in a multi cloud. A CPDP is a collection of two main algorithms (Key Gen, Tag Gen) and interactive proof system Proof [2]. Key Gen: It takes a security parameter as input and returns a secret key. Tag Gen: It takes a secret key, file and set of cloud storage providers as input and returns triplet. GenProof: A protocol to generate a proof of data possession among the CSP's and data verifier. The CPDP approach allows parallel computing which enhances performance and also provides support for large file storage on cloud.

B. Designated-Verifier Provable Data Possession in Public Cloud Storage

In public clouds, it data integrity is a matter of crucial importance when the client cannot perform the remote data possession checking. The normal PDP approach increases overhead for clients where client needs to calculate tags and hash values for the data. The Yongjun Ren, et al. [3] proposed the designated data verification model for the clients with less recourses and computational power. The authors have proposed to use ECC (Elliptic Curve Cryptography)-based homomorphism authenticator to design PDP scheme, which does not compute expensive and time consuming bilinear and consume small amount of calculation and communications. This scheme is best suited for mobile clouds. In terms of complexities, compared to RSA, elliptic curves cryptography (ECC)[10] provides shorter key length based on the same level of security. It has been shown by authors that 160-bit ECC provides comparable security to 1024-bit RSA. The communication overhead caused mostly comes from the DV-PDP response.

C. Dynamic Provable Data Possession (DPDP)

The Data dynamics plays important role in data integrity checking techniques. The PDP provides best suited scheme for only static outsourced data files, where as in general, client may wish to alter the outsourced data occasionally. To address and solve this issue C. Chris Erway, et al. provided Dynamic Provable Data Possession (DPDP) to allow data dynamics in outsourced data. They present a framework and efficient structures for DPDP approach, which extends the PDP model to support provable updates to stored data by introducing new version of authenticated dictionaries based on rank information. They provided two approaches of DPDP [4], where a rank-based authenticated dictionary was built over a skip list. This construction provides a DPDP scheme with log computation and communication and the same detection probability as the original PDP scheme; and other is an alternative construction of a rank-based authenticated dictionary using an RSA tree [4]. This construction results in a DPDP scheme with improved detection probability but increases server computation.

D. Robust DPDP

A robust DPDP scheme implements mechanisms to mitigate arbitrary amounts of data corruption. The protection against small corruptions (i.e., bytes or even bits) ensures that attacks that modify a few bits do not destroy an encrypted file or invalidate authentication information. As updating a small portion of the file may require retrieving the entire file, the PDP scheme must be robust enough to perform dynamic updates. The author Bo Chen, et al. [5] proposed two approaches towards Robust DPDP, the first construction provides efficient encoding, but causes high communication cost for updates. The second construction overcomes this drawback through a combination of techniques that consists RS codes based on Cauchy matrices, separating the encoding for robustness from the symbol position in the file, and reducing add/remove operations to append/modify operations when updating the RS-encoded parity data. Robustness is a vital property for all PDP schemes that rely on spot checking, which includes the majority of static and dynamic PDP protocols.

E. Identity based Remote Data Possession Checking

The existing PDP protocols have been designed in the PKI (public key Infrastructure) setting. In PDP approach, the cloud server has to authenticate the users’ certificates before storing the data uploaded by the users in order to prevent spam. This incurs considerable costs as many users may frequently upload data to the cloud server. The author Huaqun Wang addressed this problem with a new model of identity-based RDPC (ID-RDPC) protocols [6]. They provided first ID based PDP protocol to be secure assuming the hardness of the standard computational Diffie-Hellman (CDH) problem. In addition to the structural advantage of elimination of certificate management [11] and verification, the ID-RDPC protocol also outperforms existing PDP protocols in the PKI setting in terms of computation and communication. Firstly, the PKG (Private Key Generator) generates the system public key and the master secret key along with the private keys for the clients of an organization [6]. The main challenge to design the ID-RDPC protocol was that it requires the client to generate aggregatable ID-based signatures like tags for blocks without applying the hashand-sign paradigm to the original data. The authors addressed this with a variation of the well-known Schnorr signature [11].

F. Identity Based Distributed PDP

In some scenarios, the clients have to store their data on multi-cloud servers to allow parallelism and huge data storage. So, the integrity checking protocol must be efficient to save the verifier's cost. The author Wang, H. proposed a novel PDP model as ID-DPDP (identity-based distributed provable data possession) in multi-cloud storage. Based on the bilinear pairing concept, the complete IDDPDP protocol is designed [7]. The proposed ID-DPDP protocol is provably secure under the hardness assumption of the standard CDH (computational Diffie- Hellman) problem as tested by author. In addition to the structural advantage of elimination of managing certificate, the IDDPDP approach is efficient and flexible. Depending on the client's authorization, the proposed ID-DPDP protocol can identify private verification and public verification.

V. IMPLEMENTATION

Our implementation of the presented schemes consists of three modules: OModule (owner module), CModule (CSP module), and VModule (verifier module). OModule, which runs on the owner side, is a library that includes KeyGen, CopyGen, TagGen, algorithms. CModule is a library that runs on Amazon EC2 and includes Execute Update and Prove algorithms. VModule is a library to be run at the verifier side and includes the Verify algorithm. In the experiments, we do not believe the system preprocessing time to arrange the different file copies and produce the tags set. This pre-processing is complete only once during the life time of the scheme which may be for tens of years. Furthermore, in the implementation we do not think the time to access the file blocks, as the state-of-the-art hard drive Deletion: When one block is deleted all following blocks is motivated one step forward. To delete a specific data block at position j from all copies, the owner deletes the entry at position j from the MVT and sends a delete request I DF , BD, j, null, null_ to the CSP.Technology permits as much as 1MB to be read in just few nanoseconds [5]. Therefore, the total access time is improbable to have substantial impact on the overall system performance. We utilize the BarretoNaehrig (BN) [33] curve defined over prime field G F(p) with | p| = 256 bits and embedding degree = 12 nanoseconds [5]. Hence, the total access time is unlikely to have considerable force on the overall system presentation. In addition, it enables clients to specify geographic locations for storing their data.

VI. SUMMARY AND CONCLUDING REMARKS

Outsourcing data to remote servers has turn into a growing trend for many organizations to ease the burden of local data storage and protection. In this work we have considered the difficulty of creating multiple copies of dynamic data file and confirm those copies stored on untrusted cloud servers.We have proposed a new PDP scheme (referred to as MBPMDDP), which supports outsourcing of multi-copy dynamic data, where the data owner is skilled of not only archiving and accessing the data copies stored by the CSP, but also updating and scaling these copies on the remote servers. The proposed scheme is the first to address multiple copies of dynamic data. The communication between the authorized users and the CSP is measured in our system, where the authorized users can effortlessly access a data copy received from the CSP using a single secret key shared with the data owner. Furthermore, the proposed scheme supports public verifiability, allows arbitrary number of auditing, and allows possession-free verification where the verifier has the capability to verify the data integrity even though they neither possesses nor retrieves the file blocks from the server. From side to side performance analysis and experimental results, we have established that the proposed MB-PMDDP scheme outperforms the TBPMDDP come near derived from a class of dynamic single-copy PDP models. The TB-PMDDP leads to high storage transparency on the remote servers and high computations on both the CSP and the verifier sides. The MB-PMDDP scheme considerably decreases the computation time during the challenge-response stage which makes it more practical for request where a large number of verifiers are connected to the CSP causing a huge computation overhead on the servers. A slight alteration can be done on the proposed scheme to hold up the feature of recognizing the indices of corrupted copies. The corrupted data copy can be rebuild even from a complete damage using duplicated copies on other servers. Through algorithms, we have shown that the proposed system is probably safe.

REFERENCES

[1] Ayad F. Barsoum and M. Anwar Hasan, " Provable Multicopy Dynamic Data Possession in Cloud computing systems", in IEEE Transactions On Information Forensics And Security, Vol. 10, No. 3, March 2015

[2] G. Ateniese et al., "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, 2007, pp. 598–609.

[3]K. Zeng, "Publicly verifiable remote data integrity," in Proc. 10th Int. Conf. Inf. Commun. Secur. (ICICS), 2008, pp. 419–434.

[4] Y. Deswarte, J.-J. Quisquater, and A. Saïdane, "Remote integrity checking," in Proc. 6th Working Conf. Integr. Internal Control Inf. Syst. (IICIS), 2003, pp. 1–11.

[5] F. Sebé, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," IEEE Trans. Knowl. Data Eng., vol. 20, no. 8, pp. 1034–1038, Aug. 2008.

[6] P. Golle, S. Jarecki, and I. Mironov, “Cryptographic primitives enforcing communication and storage complexity,” in Proc. 6th Int. Conf. Finan-cial Cryptograph. (FC), Berlin, Germany, 2003, pp. 120–135.

[7] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, “Auditing to keep online storage services honest,” in Proc. 11th USENIX Workshop Hot Topics Oper. Syst. (HOTOS), Berkeley, CA, USA, 2007, pp. 1–6.

[8] M. A. Shah, R. Swaminathan, and M. Baker, “Privacy-preserving audit and extraction of digital contents,” IACR Cryptology ePrint Archive, Tech. Rep. 2008/186, 2008.

[9] E. Mykletun, M. Narasimha, and G. Tsudik, “Authentication and integrity in outsourced databases,” ACM Trans. Storage, vol. 2, no. 2, pp. 107–138, 2006.

[10] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, “Scalable and efficient provable data possession,” in Proc. 4th Int. Conf. Secur. Privacy Commun. Netw. (SecureComm), New York, NY, USA, 2008, Art. ID 9.

[11] C. Wang, Q. Wang, K. Ren, and W. Lou. (2009). “Ensuring data storage security in cloud computing,” IACR Cryptology ePrint Archive, Tech. Rep. 2009/081. [Online]. Available: http://eprint.iacr.org/

[12] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, “Dynamic provable data possession,” in Proc. 16th ACM Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, 2009, pp. 213–222.

[13] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling public verifiability and data dynamics for storage security in cloud computing,” in Proc. 14th Eur. Symp. Res. Comput. Secur. (ESORICS), Berlin, Germany, 2009, pp. 355–370.

[14] Z. Hao, S. Zhong, and N. Yu, “A privacypreserving remote data integrity checking protocol with data dynamics and public verifiability,” IEEE Trans. Knowl. Data Eng., vol. 23, no. 9, pp. 1432–1437, Sep. 2011.

[15] A. F. Barsoum and M. A. Hasan. (2010). “Provable possession and replication of data over cloud servers,” Centre Appl. Cryptograph. Res., Univ. Waterloo, Waterloo, ON, USA, Tech. Rep. 2010/32. [Online]. Available: http://www.cacr.math.uwaterloo.ca/techreports/2010/ cacr2010-32.pdf