**New Algorithms for Secure Outsourcing of**

**Large-Scale Systems of Linear Equations**

**ABSTRACT**

**With the rapid development in availability of cloudservices, the techniques for securely outsourcing the prohibitivelyexpensive computations to untrusted servers are getting moreand more attentions in the scientific community. In this paper,we investigate secure outsourcing for large-scale systems of linearequations, which are the most popular problems in various engineeringdisciplines. For the first time, we utilize the sparse matrixto propose a new secure outsourcing algorithm of large-scalelinear equations in the fully malicious model. Compared with thestate-of-the-art algorithm, the proposed algorithm only requires(optimal) one round communication (while the algorithm requiresL rounds of interactions between the client and cloud server,where L denotes the number of iteration in iterative methods).Furthermore, the client in our algorithm can detect the misbehaviorof cloud server with the (optimal) probability 1. Therefore,our proposed algorithm is superior in both efficiency andcheckability. We also provide the experimental evaluation thatdemonstrates the efficiency and effectiveness of our algorithm.**

**EXISTING SYSTEM**

**We argue that the traditional encryption technique can onlyprovide a partial solution to this problem since it is very difficultto perform meaningful computations over the ciphertext.Though the fully homomorphic encryption could be a potentialsolution, the existing schemes are not practical yet. Secondly,the semi-trusted cloud servers may return an invalid result. Forexample, the servers might contain a software bug that willfail on a constant number of invocation. Moreover, the serversmight decrease the amount of the computation due to financialincentives and then return a computationally indistinguishable(invalid) result. Therefore, the second security challenge is thecheckability of the outsourcing computation: the outsourcershould have the ability to detect any failures if the cloudservers misbehave.**

**PROPOSED SYSTEM:**

**we propose an efficient outsource-securealgorithm for large-scale systems of linear equations, whichare the most basic and expensive operations in many engineeringdisciplines. The proposed algorithm is suitable forany nonsingular dense matrix even in the strongest adversarialmodel (a.k.a. fully malicious model). Furthermore, our algorithmis superior in both efficiency and checkability than thestate-of-the-art algorithm**

**MODULE DESCRIPTION:**

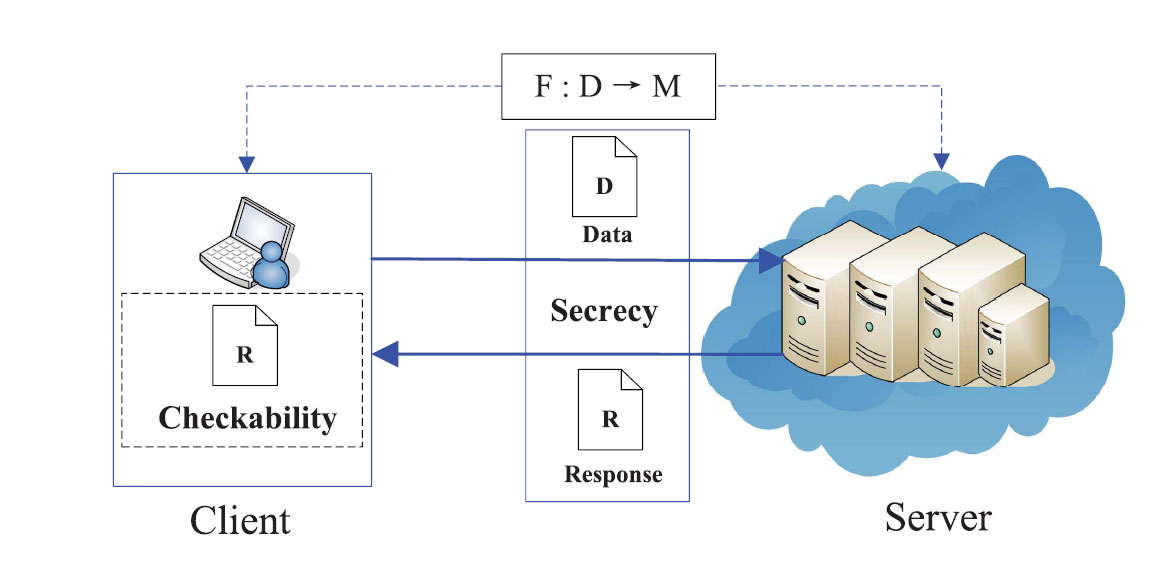
# Number of Modules:

After careful analysis the system has been identified to have the following modules:

1. Cloud computing
2. Outsource-secure algorithms**,**
3. System of linear equations.
4. Security Model

# Cloud computing.

# cloud computing is a computing term or metaphor that evolved in the late 2000s, based on utility and consumption of computer resources. Cloud computing involves deploying groups of remote servers and software networks that allow different kinds of data sources be uploaded for real time processing to generate computing results without the need to store processed data on the cloud.



# Outsource-secure algorithms,

**Despite the tremendous benefits, outsourcing computationalso inevitably suffers from some new security challenges. Firstly, the computation tasks often containsome sensitive information that should not be exposed tothe (semi-trusted) cloud servers. Therefore, the first securitychallenge is the privacy of the outsourcing computation: thecurious cloud servers should not learn anything about what it isactually computing (including the secret inputs and outputs).We argue that the traditional encryption technique can onlyprovide a partial solution to this problem since it is very difficultto perform meaningful computations over the ciphertext.Though the fully homomorphic encryption could be a potentialsolution, the existing schemes are not practical yet. Secondly,the semi-trusted cloud servers may return an invalid result. Forexample, the servers might contain a software bug that willfail on a constant number of invocation. Moreover, the serversmight decrease the amount of the computation due to financialincentives and then return a computationally the test procedure should neverbe involved in some other complicated computations sincethe computationally limited devices may be incapable toaccomplish a complicated test. At the very least, it must be farmore efficient than accomplishing the computation task itself(recall the motivation for outsourcing computations.**

**Security Model**

**consider an outsourcing computation architecture inwhich an honest while resources-constrained client C wantsto outsource an expensive implements F if F(x) = CS(x) for ∀ x ∈ D, whereCS We means that C is given oracle access to S that records allof its computation over time.Golle and Mironov first introduced the “lazy- of work it needs to performin order to retrieve the payment, while it will honestly providethe computation results to C. It is reasonable to assume thatS is honest in this kind of outsourcing computation model buthonest”model for the computation task F : D → M to acloud server S, as illustrated in Fig. 1. Trivially, S is not fullytrusted by C. Without loss of generality, we say that (C, S)correctly inversion of one-way function classof outsourcing computations. As a rational economic agent,S will try to minimize the amount.Actually, given an invalid computation result, C can detect itimmediately with probability 1 since the verification of theresult is equivalent to compute the one-way functions.Another well-known model is the “honest-but-curious”model (originally called the semi-honest model), firstly introducedby Goldreich et al. . In this model, both the partiesare guaranteed to properly execute a prescribed protocol, but,at the end of it, one party can use its own view of the executionto infer about the other’s input. Therefore, S will also honestlysend the computation results to C. However, S will try hisbest to retrieve some sensitive information such as the secretinput/output of C. Note that the computation results (i.e., theoutput of S) are different from the output of C (i.e., the realcomputation aim of C).We can view S in the above two models as a passiveadversary. In many applications, we must consider the case thatS is an active attacker. For example, S can intentionally send acomputationally indistinguishable (invalid) result to C. That is,S may be not only lazy and curious, but also dishonest. Thisis the strongest adversarial model, and we name it the “fullymalicious model.” I**

**System of linear equations.**

**Compared with the state-of-the-artalgorithm , the proposed algorithm is superior in bothefficiency and checkability. Our contributions are three folds:**

**1) For the first time, we utilize the sparse matrix toinvestigate securely outsourcing for large-scale systemsof linear equations. Our algorithm is suitable for anynonsingular dense matrix A. However, in algorithm ,A must be a strictly diagonally dominant matrix forconvergence.**

**2) Our proposed algorithm only requires (optimal) 1 roundcommunication between the client C and server S, whilealgorithm requires L interactions due to the iterativemethod.**

**3) In the proposed algorithm, the client C can detect themisbehavior of server S with the (optimal) probability 1.Surprisingly, we use neither the complicated knowledgeproof techniques nor the boolean garbled circuits.**

**System Configuration:**

HARDWARE REQUIREMENTS:

Hardware - Pentium

Speed - 1.1 GHz

RAM - 1GB

Hard Disk - 20 GB

Key Board - Standard Windows Keyboard

SOFTWARE REQUIREMENTS:

Operating System : Windows

Technology : Java and J2EE

Web Technologies : Html, JavaScript, CSS

IDE : My Eclipse

Web Server : Tomcat

Database : My SQL

Java Version : J2SDK1.5

**CONCLUSION**

In this paper, we propose an efficient outsource-securealgorithm for large-scale systems of linear equations, whichare the most basic and expensive operations in many engineeringdisciplines. The proposed algorithm is suitable forany nonsingular dense matrix even in the strongest adversarialmodel (a.k.a. fully malicious model). Furthermore, our algorithmis superior in both efficiency and checkability than thestate-of-the-art algorithm