Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 34 (2014) 474 – 483

Procedia Computer Science

The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC-2014)

A Practical, Secure, and Verifiable Cloud Computing for Mobile Systems

Sriram N. Premnath[a], Zygmunt J. Haas[a,b,*]

[a] *Cornell University, 324 Frank Rhodes Hall, Ithaca, NY 14853, U.S.A.*
[b] *University of Texas at Dallas, 800 W. Campbell Road, Richardson, TX 75080, U.S.A.*

Abstract

Cloud computing systems, in which clients rent and share computing resources of third party platforms, have gained widespread use in recent years. Furthermore, cloud computing for mobile systems (i.e., systems in which the clients are mobile devices) have too been receiving considerable attention in technical literature. We propose a new method of delegating computations of resource-constrained mobile clients, in which multiple servers interact to construct an encrypted program known as garbled circuit. Next, using garbled inputs from a mobile client, another server executes this garbled circuit and returns the resulting garbled outputs. Our system assures privacy of the mobile client's data, even if the executing server chooses to collude with all but one of the other servers. We adapt the garbled circuit design of Beaver et al. and the secure multiparty computation protocol of Goldreich et al. for the purpose of building a secure cloud computing for mobile systems. Our method incorporates the novel use of the cryptographically secure pseudo random number generator of Blum et al. that enables the mobile client to efficiently retrieve the result of the computation, as well as to verify that the evaluator actually performed the computation. We analyze the server-side and client-side complexity of our system. Using real-world data, we evaluate our system for a privacy preserving search application that locates the nearest bank/ATM from the mobile client. We also measure the time taken to construct and evaluate the garbled circuit for varying number of servers, demonstrating the feasibility of our secure and verifiable cloud computing for mobile systems.

Selection and peer-review under responsibility of Conference Program Chairs
Keywords: Secure Cloud Computing; Garbled Circuits, Secure Multiparty Computation

1. Introduction

Cloud computing systems, in which the clients rent and share computing resources of third party platforms such as Amazon Elastic Cloud, Microsoft Azure, etc., have gained widespread use in recent years. Provisioned with a large pool of hardware and software resources, these cloud computing systems enable clients to perform computations on a vast amount of data without setting up their own infrastructure [1]. However, providing the cloud service provider with the client data in *plaintext form* to carry out the computations will result in complete loss of data privacy.

Homomorphic encryption [2] is an approach to tackle the problem of preserving data privacy, which can allow the cloud service providers to perform specific computations directly on the encrypted client data, without requiring

* Corresponding author. Tel.: +1-607-255-3454 ; fax: +1-607-255-9072.
E-mail address: haas@ece.cornell.edu

Selection and peer-review under responsibility of Conference Program Chairs
doi:10.1016/j.procs.2014.07.033

private decryption keys. Recently, fully homomorphic encryption (FHE) schemes (e.g., Gentry et al.[3]) have been proposed, which enable performing any arbitrary computation on encrypted data. *However, FHE schemes are currently impractical for mobile cloud computing applications due to extremely large cipher text size*. For instance, to achieve 128-bit security, the client is required to exchange a *few Giga bytes of ciphertext* with the cloud server, for each bit of the plain text message[3]. *Thus, there is a need for a more efficient alternative, which is suitable for mobile systems.*

Yao's garbled circuits approach[4,5], which we consider in our work, is a potential alternative to FHE schemes that can drastically reduce the ciphertext size. Any computation can be represented using a Boolean circuit, for which, there exists a corresponding garbled circuit[4,5,6,7]. Each gate in a garbled circuit can be unlocked using a pair of input *wire keys* that correspond to the underlying plaintext bits; and the association between the wire keys and the plaintext bits is kept secret from the cloud server that performs the computation. Unlocking a gate using a pair of input wire keys reveals an output wire key, which, in turn, serves as an input wire key for unlocking the subsequent gate in the next level of the circuit. Thus, garbled circuits can enable *oblivious evaluation* of any arbitrary function, expressible as a Boolean circuit, on a third-party cloud server.

While garbled circuits preserve the privacy of client data, they are, however, one time programs – using the same version of the circuit more than once compromises the garbled circuit and reveals to an adversarial evaluator whether the semantics have changed or remained the same for a set of input and output wires between successive evaluations. Expecting the client to create a new version of the garbled circuit for each evaluation, however, is an unreasonable solution, since creating a garbled circuit is at least as expensive as evaluating the underlying Boolean circuit! *Thus, in contrast to FHE schemes such as that of Gentry[3], that can directly delegate the desired computation to the cloud servers, a scheme using garbled circuits, presents the additional challenge of efficiently delegating to the cloud servers the creation of garbled circuit.*

We propose a new method, in which whenever the client needs to perform a computation, it employs a number of cloud servers to create a new version of the garbled circuit in a distributed manner. Each server generates a set of private input bits using unique seed value from the client and interacts with all the other servers to create a new garbled circuit, which is a function of the private input bits of all the servers. Essentially, the servers engage in a secure multiparty computation protocol (e.g., Goldreich et al.[6,7]) to construct the desired garbled circuit without revealing their private inputs to one another. Once a new version of the garbled circuit is created using multiple servers, the client delegates the evaluation to an arbitrary server in the cloud. The resulting version of the garbled circuit, the garbled inputs that can unlock the circuit, and the corresponding garbled outputs, remain unrecognizable to the evaluator, even if it chooses to collude with any strict-subset of servers that participated in the creation of the garbled circuit.

Our proposed system is designed to readily exploit the real-world asymmetry that exists between typical mobile clients and cloud servers – while the mobile clients are *resource-constrained*, the cloud servers, on the other hand, are sufficiently provisioned to perform numerous intensive computation and communication tasks. To achieve secure and verifiable computing capability, our system requires very little computation and communication involvement from the mobile client beyond the generation and exchange of *compact cipher text messages*. However, using significantly larger resources, the cloud servers can efficiently generate and exchange a large volume of random bits necessary for carrying out the delegated computation. *Thus, our proposed scheme is very suitable for mobile environments.*

We adapt the garbled circuit design of Beaver, Micali, Rogaway (BMR[8,9]), and the secure multiparty computation protocol of Goldreich et al.[6,7] to suit them for the purpose of building a secure cloud computing system. To facilitate the construction of the garbled circuit, and also to enable the client to *efficiently retrieve and verify the result of the computation*, our method incorporates the novel use of the cryptographically secure pseudo random number generator of Blum, Blum, Shub[10,11], whose strength relies on the computational difficulty of factorizing large numbers into primes. *Our proposed system enables the client to efficiently verify that the evaluator actually and fully performed the requested computation.*

Our major contributions in this work include the following: (i) we design a secure mobile cloud computing system using multiple servers that enables the client to delegate any arbitrary computation, (ii) our system assures the privacy of the client input and the result of the computation, even if the evaluating server colludes with all but one of the servers that created the garbled circuit, (iii) our system enables the client to efficiently recover the result of the computation and to verify whether the evaluator actually performed the computation, (iv) we present an analysis of the server-side and client-side complexity of our proposed scheme. Our findings show that in comparison to Gentry's FHE scheme, our scheme uses very small cipher text messages suitable for mobile clients, (v) using real-world data, we evaluate our