Secure Outsourcing of Data and Arbitrary Computations with Lower Latency Token-Based Cloud Computing

Asha Rani M[1], Kavana M D[2], Soumya C T[3], Pruthvi P[4]
[1, 2, 3, 4] Department of CSE
[1, 2, 3, 4] GSSSIETW,Mysuru

Abstract- *Secure outsourcing of computation to an untrusted service provider is becoming more and more important. Pure cryptographic solutions based on fully homomorphic and verifiable encryption, recently proposed, are promising but suffer from very high latency. Other proposals perform the whole computation on tamper-proof hardware and usually suffer from the the same problem. Trusted computing (TC) is another promising approach that uses trusted software and hardware components on computing platforms to provide useful mechanisms such as attestation allowing the data owner to verify the integrity of the cloud and its computation. However, on the one hand these solutions require trust in hardware (CPU, trusted computing modules) that are under the physical control of the cloud provider, and on the other hand they still have to face the challenge of run-time attestation. In this paper we focus on applications where the latency of the computation should be minimized, i.e., the time from submitting the query until receiving the outcome of the computation should be as small as possible. To achieve this we show how to combine a trusted hardware token with Secure Function Evaluation to compute arbitrary functions on secret data where the computation leaks no information and is verifiable. The token is used in the setup phase only whereas in the time-critical online phase the cloud computes the encrypted function on encrypted data using symmetric encryption primitives only and without any interaction with other entities.*

Keywords- Cloud Computing, Hardware Token, Outsourcing.

I. INTRODUCTION

Enterprises and other organizations often have to store and operate on a huge amount of data. Cloud computing offers infrastructure and computational services on demand for various customers on shared resources. Services that are offered range from infrastructure services such as Amazon EC2. While sharing IT infrastructure in cloud computing is cost-efficient and provides more flexibility for the clients, it introduces security risks organizations have to deal with in order to isolate their data from other cloud clients and to fulfill confidentiality and integrity demands. Moreover, since the IT infrastructure is now under control of the cloud provider, the customer has not only to trust the security mechanisms and configuration of the cloud provider, but also the cloud provider itself. When data and computation is outsourced to the cloud, prominent security risks are: malicious code that is running on the cloud infrastructure could manipulate computation and force wrong results or steal data; personnel of the cloud provider could misuse their capabilities and leak data; and vulnerabilities in the shared resources could lead to data leakage or manipulated computation. Secure outsourcing of arbitrary computation and data storage is particularly difficult to fulfill if a cloud client does not trust the cloud provider at all. There are proposals for cryptographic methods which allow to perform specific computations on encrypted data [3], or to securely and verifiably outsource storage. Arbitrary computation on confidential data can be achieved with fully homomorphic encryption [12], in combination with garbled circuits for verifiability[11].. The use of trusted computing based remote attestation in the cloud .Trusted Virtual Domains [5,6] are one approach that combines trusted computing, secure hypervisors, and policy enforcement of information flow within and between domains of virtual machines. However, those approaches require trust in a non-negligible amount of hardware (e.g., CPU, Trusted Platform Module which are under the physical control of the cloud provider. According to the specification of the Trusted Computing Group, the TPM is not designed to protect against hardware attacks, but provides a shielded location to protect keys. However, the TPM cannot perform arbitrary secure computations on data. It can protect cryptographic keys and perform only pre-defined cryptographic operations like encryption, decryption, and signature creation. In particular, if data should be encrypted it must be provided in plaintext to the TPM, and if data should be decrypted it will be given in plaintext as output. Unfortunately, the TPM cannot be instructed to decrypt data internally, perform computations on the data, and encrypt it again before returning the output. A virtualized TPM [4] that is executed in software could be

enhanced with additional functionality. However, such software running on the CPU has access to unencrypted data at some point to compute on it. Hence, if the cloud provider is malicious and uses specifically manipulated hardware, confidentiality and verifiability cannot be guaranteed by using trusted computing. A hardware token which is tamper-proof against physical attacks but can perform arbitrary computations would enable the cloud client to perform confidential and verifiable computation on the cloud provider's site, given that the client trust the manufacturer of the token that it does not leak any information to the provider. For example, secure coprocessors are tamper-proof active programmable devices that are attached to an untrusted computer in order to perform security-critical operations or to allow to establish a trusted channel through untrusted networks and hardware devices to a trusted software program running inside the secure coprocessor. Data can be stored encrypted outside the token in cloud storage while decryption keys are stored in shielded locations of the trusted tokens. The token based approach is reasonable because both, cryptographic coprocessors and standardized interfaces .exist that can be used for such tokens. Of course, for trust reasons, the token vendor should not be the same as the cloud provider. However, the whole security-critical computation takes place in the token. Hence, such computation is not really outsourced to the cloud because the function is computed within the token. Some applications, however, require fast replies to queries which cannot be computed online within the tamper-proof token. For example, queries in personal health records or payroll databases may occur not very frequently, but need to be processed very fast while privacy of the data should be preserved. In this paper, we focus on cloud application scenarios where private queries to the outsourced data have to be processed and answered with low latency.

II. MODEL FOR SECURE OUTSOURCING OF DATA AND ARBITRARY COMPUTATIONS

We consider the model shown in Fig. 1 that allows a client C to verifiably and securely outsource a database D and computations thereon to an untrusted service provider S.

A client C (e.g., a company) wants to securely outsource data D and computation of a function there on to an untrusted service provider S who offers access to storage services and to computation services. Example applications include outsourcing of medical data, log files or payrolls and computing arbitrary statistics or searches on the outsourced data. In addition, the evaluation of f can depend on a session-specific private query xi of C resulting in the response yi = f(xi,D). However, S should be prevented from learning or modifying D) or to compute f incorrectly .1 Any cheating attempts of a malicious S who tries to deviate from the protocol should be detected by C with overwhelming probability where C outputs the special failure symbol !.2

While this scenario can be easily solved for a restricted class of functions private search of a keyword xi using searchable encryption we consider the general case of arbitrary functions f. Due to the large size of D and or the computational complexity of f, it is not possible to securely outsource D to S only and let C compute f after retrieving D from S. Instead, the confidentiality and integrity of the outsourced data D has to be protected while at the same time secure computations on D need to be performed at S without interaction with C

III. ENCRYPTION AND AUTHENTICATION

Confidentiality and authenticity of messages can be guaranteed either symmetrical key or asymmetrical keys. The symmetric case can be instantiated with a combination of symmetric encryption and a message authentication code . These schemes use a respective symmetric key for encryption/ authentication and the same key for decryption/verification. Alternatively, public-key cryptography allows usage of separate keys for encryption/authentication and other keys for decryption/ verification. This could be used for example to construct an outsourced database to which new entries can be appended by multiple parties without using shared symmetric keys . encrypted and authenticated Data x.

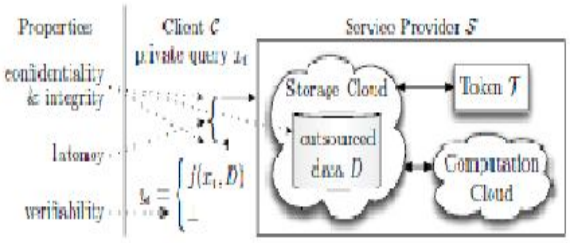

Fig. 1. Model for Secure Outsourcing of Data and Computation

Figure 1.

1. Fully Homomorphic Encryption

Fully homomorphic encryption is semantically secure public-key encryption that additionally allows computing an arbitrary function on encrypted data using the public-key only, i.e., given a cipher text !x", a function f and the public-key , it is possible to compute !y" = EVAL pk(f, !x") = !f(x)". Constructing a homomorphic encryption scheme with polynomial overhead was a longstanding open problem.

Recently, there are several proposals starting with [12] and subsequent extensions and improvements of . Still, all these schemes employ computationally expensive public-key operations for each gate of the evaluated function and hence are capable of evaluating only very small functions on today's hardware.

IV. ARCHITECTURES FOR SECURE OUTSOURCING OF DATA AND ARBITRARY COMPUTATION

Symmetric keys for encryption and verification. The encrypted and authenticated database D and the authenticated function is stored within the storage cloud of service provider S. In the online phase, C sends the encrypted and authenticated query xi to T and the storage cloud provides D and ! f one-by-one. T decrypts and verifies these inputs and evaluates yi = f(xi,D) using secure external memory. If T detects any inconsistencies, it continues evaluation substituting the inconsistent value with a random value ,and finally sets yi to the failure symbol !. Finally, T sends the authenticated and encrypted response yi back to C who decrypts, verifies and obtains the output yi. Performance. In this approach, the latency of the online phase, i.e., the time from sending the query xi to receiving the response yi, depends on the performance and cannot be improved by using the computation cloud services offered by S.

Token Sets Up and Cloud Computes:

Our approach combines a tamper-proof hardware token T used in the setup phase only with efficient computations performed in parallel in the computation cloud as shown in The basic idea is that T generates a garbled circuit during the setup phase and in the time-critical online phase the garbled circuit is evaluated in parallel by the computation cloud.

In detail, our architecture consists of the following three phases: During System Initialization, client C and the tamper-proof hardware token T agree on a symmetric key. Additionally, C provides with the authenticated function and the authenticated and encrypted data D who stores them in the storage cloud.

V. CONCLUSION

Our entire architecture is based solely on symmetric cryptographic primitives and hence is very efficient In the Setup Phase, T generates for protocol invocation i an internal session key ki derived from the key k and i. Using ki, T generates a garbled circuit from the function of and a corresponding garbled re-encryption "Di of the database D which are stored in the storage cloud: According to the construction of [16], the GC can be generated gate-by-gate using a constant amount of memory only. For each gate of ! f, S provides T with the description of the gate. T uses the session key ki to derive the gate's garbled input values and the garbled output value and returns the corresponding garbled table to S. Summary. We discussed a model and several possible architectures for outsourcing data and arbitrary computations that provide confidentiality, integrity, and verifiability. The first architecture is based on a tamper-proof hardware token, the second on evaluation of a garbled circuit under fully homomorphic encryption, and the third is a combination of both approaches. complexity of the presented architectures is the same: the client C performs work linear in the size of the inputs.online latency, i.e., the time between C submitting the encrypted query xi to the service provider S until obtaining the result differs substantially in practice. For the token-based architecture of , the online latency depends on the performance of the token T that evaluates f and hence is hard to parallelize and might become a bottleneck in particular when f is large and T must resort to secure external memory in the storage cloud. The homomorphic encryption-based architecture does not use a token and hence can exploit the parallelism offered by the computation cloud. However, this architecture is not ready for deployment in practical applications yet, as fully homomorphic encryption schemes are not yet sufficiently fast enough for evaluating a large functionality such as a garbled circuit under fully homomorphic encryption.

REFERENCES

[1] Amazon Elastic Compute Cloud (EC2), http://aws.amazon.com/ec2.

[2] Amazon Simple Storage Service (S3), http://aws.amazon.com/s3

[3] Atallah, M.J., Pantazopoulos, K.N., Rice, J.R., Spafford, E.H.: Secure outsourcing of scientific computations. Advances in Computers 54, 216–272 (2001)

[4] Berger, S., Caceres, R., Goldman, K.A., Perez, R., Sailer, R., Doorn, L.v.: vTPM: Virtualizing the Trusted Platform Module. In: USENIX Security Symposium(USENIX 2006), pp. 305–320. USENIX Association (2006)

[5] Bussani, A., Griffin, J.L., Jasen, B., Julisch, K., Karjoth, G., Maruyama, H., Nakamura,M., Perez, R., Schunter, M., Tanner, A., Van Doorn, L., Herreweghen, E.V.,Waidner, M., Yoshihama, S.: Trusted Virtual Domains: Secure Foundations for Business and IT

Services. Technical Report Research Report RC23792, IBM Research (November 2005)

[6] Cabuk, S., Dalton, C.I., Eriksson, K., Kuhlmann, D., Ramasamy, H.G.V., Ramunno,G., Sadeghi, A.-R., Schunter, M., St¨uble, C.: Towards automated security policy enforcement in multi-tenant virtual data centers. Journal of Computer Security 18, 89–121 (2010)

[7] Chow, R., Golle, P., Jakobsson, M., Shi, E., Staddon, J., Masuoka, R., Molina, J.: Controlling data in the cloud: outsourcing computation without outsourcing control. In: ACM Workshop on Cloud Computing Security (CCSW 2009), pp. 85– 90. ACM, New York (2009)

[8] Cloud Security Alliance (CSA). Top threats to cloud computing, version 1.0 (March 2010), http://www.cloudsecurityalliance.org/topthreats/csathreats .v1.0.pdf

[9] Dijk, M.v., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. Cryptology ePrint Archive, Report 2009/616 (2009), http://eprint.iacr.org; To appear at EUROCRYPT 2010

[10] Garay, J.A., Kolesnikov, V., McLellan, R.: MAC precomputation with applications to secure memory. In: Samarati, P., Yung, M., Martinelli, F., Ardagna, C.A. (eds.) ISC 2009. LNCS, vol. 5735, pp. 427–442. Springer, Heidelberg (2009)

[11] Gennaro, R., Gentry, C., Parno, B.: Non-interactive verifiable computing: Outsourcing computation to untrusted workers. Cryptology ePrint Archive, Report 2009/547 (2009), http://eprint.iacr.org

[12] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: ACM Symposium on Theory of Computing (STOC 2009), pp. 169–178. ACM, New York (2009)

[13] Google App Engine, https://appengine.google.com

[14] Goldwasser, S., Kalai, Y.T., Rothblum, G.N.: One-time programs. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 39–56. Springer, Heidelberg (2008) Token-Based Cloud Computing 429

[15] IBM. IBM Cryptocards, http://www-03.ibm.com/security/cryptocards/

[16] J¨arvinen, K., Kolesnikov, V., Sadeghi, A.-R., Schneider, T.: Embedded SFE: Offloadingserver and network using hardware tokens. In: Financial Cryptography