Privacy-Enhanced Transactions for Virtual Organisations

Erica Y. Yang[1] Jie Xu[2] Keith H. Bennett[1]

[1]Department of Computer Science
University of Durham
South Road, Durham DH1 3LE
United Kingdom

[2]School of Computing
University of Leeds
Leeds DH2 9LT
United Kingdom

{erica.yang, keith.bennett}@durham.ac.uk {jxu}@comp.leeds.ac.uk

Abstract

Virtual Organisations (VOs) that overcome the limitations of physical organisations are anticipated to play a vital role in facilitating large scale resource sharing and dynamic collaboration. The privacy of a VO's members is one of the major concerns, which may hinder the advance of this promising technology. This paper focuses on a privacy issue that arises when users send security-sensitive data (e.g., the content contains trade secrets or secret formula) to remote servers for data processing. We examine this issue, explore some partial solutions and present an overview of our current work in this area. Finally, we discuss the pros and cons of our approach and outline the future work.

1. Introduction

Security in conventional distributed systems is mainly concerned with communication privacy over insecure network channels. Cryptographic techniques are heavily exploited in protecting the exact content of messages among *mutually trusted* parties against attackers who may tap on the channels or actively disrupt the communication. An implicit assumption of this model is that *all* the participating parties will not leak the secret information to the outside world. Consider a typical example in cryptography. Alice and Bob need to talk privately over a public communication network. Various cryptosystems (e.g., RSA) can be employed to meet the security requirements. However, technically, nothing can prevent Alice from leaking the content of their conversation to an untrusted third party Trent, without letting Bob know. (Bob may know some time later. However, the legal and social obligations are beyond the scope of our work.)

Figure 1 illustrates a typical security architecture for conventional distributed system. It is very common that such systems are built, operated, and maintained by a single organisation in a centralised manner. The security of the system is ensured by employing a set of physically well-guarded and isolated servers that are interconnected by secure communication networks. Although the server may actually be composed of many untrusted components at an application level, such as the SINTRA [CP02] and the COCA [ZSR02] approaches, at an abstract level, the core of the architecture is a *single* trusted authority, i.e., the server.

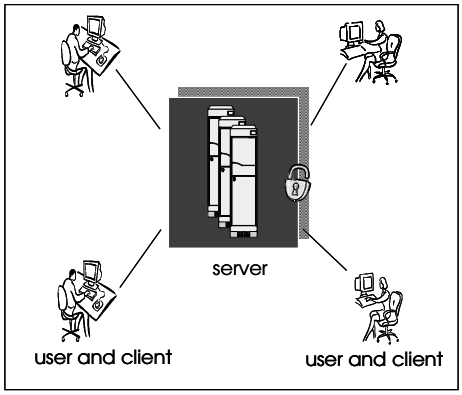

Figure 1 A Security Architecture for Conventional Distributed Systems

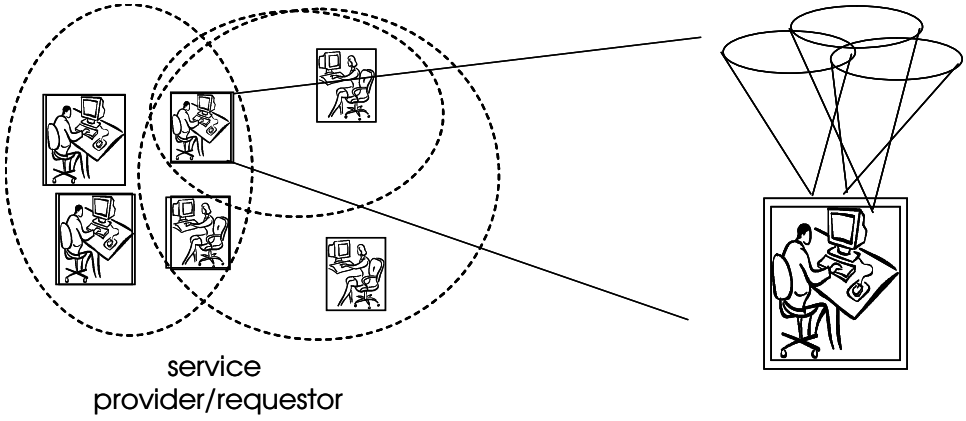

Figure 2 A Scenario of Overlapping Virtual Organisations

In a conventional distributed system, this may not be a real concern because the system typically assumes a static membership within well-defined boundaries of an organisation. The organisation's size is manageable and its members' behaviours are under strict scrutiny. Members usually have long term trust relationships (e.g., an undergraduate or a permanent member of staff) with the system. Even if any parties violate the rule, it can be easily detected due to the manageable scale of the system. It is reasonable to assume that the trust relationship among participants could be managed and maintained in such distributed systems. Consequently, the above model has been used for mainly defending against *outsiders*. Systems employing this model are vulnerable to *insider* attacks and little can be done to prevent, detect, remove, and tolerate such intentional faults.

The traditional model is inadequate for the fast growing need of a secure Grid. Grid computing is concerned with large scale and dynamic information sharing and coordinated use of resources over distributed Virtual Organisations (VOs). Compared to that of traditional distributed systems, the boundaries among VOs are blurred and *transient*, which are effectively reflecting the dynamic nature of the Grids. Figure 2 shows a scenario of VOs that service requestors and providers simultaneously engage in a number of overlapping VOs. In this scenario, it is hard to establish a long term and static trust relationship among the participants since the service provision status in the Grids is constantly changing. Essentially, that is because we have much less control over systems and services in VOs. Therefore, it is hard to achieve the degree of trust similar to that we could achieve in conventional distributed systems.

Typically, in a Grid environment, the physical boundaries among organisations are replaced with "virtual" boundaries (the dashed lines on the left in Figure 2), which may change along with the progress of collaborations. The interactions among participants are complex and dynamic due to the concurrent presence of multiple VOs. The members from one (physical) organisation may participate in more than one VO and need to access/share resources across multiple real-world organisational boundaries. A VO may also involve multiple sources of authorities, including, for example a user's VO membership and privileges derived from the VO's own authority. Hence, no *fixed* trust relationship can be easily established, managed and assumed beforehand. Continuing to use the traditional model may lead us into a vulnerable situation because there are no clear distinctions among outsiders and insiders in the new context.

As noted above, the establishment of a trust relationship among participants is a presumption of conventional distributed systems. Grid computing needs a more flexible and protected way for establishing and managing trust among participants during *and* after computations finish. The trustworthiness of service providers can not be naturally extended to the future. Nothing can guarantee

they wouldn't (intentionally or unintentionally) leak confidential information to the others.

Alternatively, we can take a *proactive* approach towards the trust problem by not relying on the trustworthiness of service providers. Without assuming a full trust relationship, participants send "encoded" transactions to execute on remote nodes and hence their privacy can be protected even against these nodes. Of course, this kind of encoding should be done in such a way that only the service requestors know how to decode.

2. Computing with Encrypted Data

This section describes the model for computing with encrypted data and gives some example functions that have been proposed for computing with encrypted data. The techniques we are proposing in the next two sections are based on the same principle of computing with encrypted data.

Computing with encrypted data [F85, AFK89] is concerned with performing calculation on a piece of information without knowing the exact content of it while ensuring results to be correct. Assume that we have two parties: Alice and Bob. Alice has an input i_a and want to calculate $f(i_a)$. She, however, does not have the resources to compute the answer. Bob has the resource and is willing to offer the help. The challenge is that i_a may contain highly security-sensitive data, such as secret formula or trade secret which Alice does not want to expose to Bob. In this case, conventional security mechanisms can help little because they mainly aim at transportation security. To conceal the content of a message, a sender commonly uses encryption to scrabble the data in such a way that only the receiver can decrypt it. The decrypted data is then processed by the receiver. Although widely adopted, this approach, however, makes an implicit assumption on the trustworthiness of the receiver. This assumption may no longer hold for some circumstances of large scale distributed computing, particularly the Grid computing. But can we process encrypted data?

Here, we give a presentation on the principle of computing with encrypted data. Readers are refereed to [AFK89] for the formal details. In the Alice and Bob example, we need a mechanism to do the following.

Alice needs a mechanism to transform $i_a \in Dom$ into $a \in Dom$ so that from a, little or no information can be obtained by Bob, where Dom denotes the domain i_a and a are drawn. (The "little" here, is rather imprecise; it is defined by the specific algorithms used. No information means that each possible element from Dom is equal likely to be i_a. Whereas little information means that the amount of information obtained by the receiver is restricted by the computational resources s/he possesses.) Let us represent the transformation, done by Alice, as follows.

$$a = E(i_a, k),$$

where k is a piece of random information that is used to scrabble i_a and E is the encryption function. Alice sends a to Bob.

When Bob receives a, he applies the function f, probably with some input i_b from Bob and obtains b as the result:

$$b = f(a, i_b).$$

Upon receiving b, Alice performs the reverse transformation by applying the decryption function D with k and i_a as follows and recovers the result $f(j)$:

$$f(i_a) = D(b, k, i_a).$$

Note that the above presentation is a general case. As a special case, Bob can have no input to the function f, that is, $b = f(a)$. The rest of the model remains the same.

There are a number of functions that have been proposed for processing encrypted data. Commonly, they are based on well-known cryptographic problems [AFK89], such as the Discrete Logarithm Problem (DLP), the Quadratic Residuosity Problem (QRP) and the Primitive Root Problem (PRP). Others are based on one-time key method (e.g., [ALN87]) or exploit the homomorphism property of polynomials (e.g., [RAD82]).

In the following, we give some brief descriptions on the DLP function for computing with encrypted data.

An Example for The DLP Function [F85]

In this example, Alice wants to obtain the discrete logarithm of a given input i but she

lacks of the resources to do so. But Bob can do the calculation it for her.

Given a prime p, a generator g for Z^*_p, a random number $k \in Z$ and an integer $i \in [1, p - 1]$, Alice transforms i into another integer $a = i \cdot g^k \pmod{p}$ and sends a, p, g to Bob.

Bob then does the following:

$$b = \log_g a \pmod{p},$$

and send b back to Alice.

Finally, Alice performs the following function to recover the result.

$$i = b - k \pmod{p - 1}.$$

3. PIR and TIR Technologies

This section introduces the PIR and TIR technologies and describes how their relationship with other security mechanisms.

Informally, an encoded transaction is a form of computation which is scrabbled in such a controlled way that the intention of the computation is unintelligent to the computing nodes. Of course, the sender should have a "reverse" mechanism in place to recover the result generated by the computation. There are fundamentally two different approaches towards a solution to the problem: information theoretic and computational. The former approach provides so-called unconditional security protection in that no information about the intention can be derived while the latter achieves security based on the computational difficulties of certain number theoretic problems.

An interesting approach on encoded transactions is the so-called Private Information Retrieval (PIR) technology. This research was first initiated by Chor et. al. in their seminal paper [CGKS95]. PIR is concerned with protecting the intention of users (i.e., what is requesting) against honest-but-curious remote servers.

The index i of an intended item is encrypted by splitting it into a set of sub-queries and generating random and independent queries q_1, q_2, …, q_k, where k is the number of replicas, based on these sub-queries. The queries can be viewed as an encrypted index. They are then sent to the remote database servers to process. By computing with the database contents and the queries, the PIR server side computations produce a set of answers. Based on the answers, the user reconstructs the result, i.e., the intended data item. The scheme can tolerate up to t "curious" servers which collude together to find out i.

This approach eliminates the need for establishing a complete trust by anoymising the transactions. It should be emphasised that conventional authentication mechanisms are still needed to ensure authenticity of participants. Similarly, authorisation mechanisms should also be in place to enforce privilege checking and access control etc. The privacy enhanced technique should be viewed as an extra level of security beyond authentication, authorisation and accountability. The purpose of the anonymization is to conceal the real interest of senders to protect them from the attacks of the receivers. Hence, this approach reduces the need for the sender to completely trust the other side.

Our recent work [YXB03] has developed an attack-Tolerant private Information Retrieval (TIR) technique by considering PIR in a more realistic attack model - malicious server model, in which servers could *also* actively disrupt the service, for example, by delivering purposefully manipulated answers. Our technique achieves information theoretic privacy protection with the use of *replication*. However, one major question remains open: are we able to remove the specific requirement for replication in PIR and TIR?

4. An Overview of Latest Results

This section describes our latest results on computational PIR and TIR (cPIR and cTIR) informally. (We are currently in the process of writing a formal presentation of the complete results.)

Since replicated servers are necessary for achieving information theoretic PIR [CGKS95], relaxing the privacy requirement is the only alternative to get around replication [KO97]. Like other computational PIR (cPIR) techniques (e.g., [KO97]), we propose here a new scheme by exploiting a number theoretic assumption for a *single* server PIR and TIR solution. Specifically, the Discrete Logarithm Problem (DLP) is extended and used to protect the secrecy of computation. Loosely speaking, DLP is a computational problem for computing discrete logarithms over finite fields. It is well known that given a sufficiently large field, the

DLP problem is hard to break. This problem has been employed in various cryptographic protocols, such as the Diffie-Hellman's key exchange protocol (which is the foundation of SSL) and the ElGamal public key cryptosystem. Here we first briefly describe the DLP problem and then outline how we apply the problem to construct our cPIR and cTIR schemes.

The DLP Problem

A DLP problem [MOV97] is the following: given a finite cyclic group G of order n and its generator g, for any element $a \in G$, find the integer x, $0 \le x \le n-1$, such that $g^x = a$.

We are interested in a specific DLP problem which is calculated in subgroups of Z^*_p, where p is a prime number and Z^*_p is a multiplicative group. Let p, q be two primes and $q|(p-1)$. Let G be the unique cyclic group of Z^*_p of order q, let g be a generator of G. The discrete logarithm problem in G is the following: given p, q, g, and $a \in G$, calculate a unique integer x, $0 \le x \le q-1$, such that $g^x = a \pmod p$. That is, the uniqueness of x is collectively determined by p, g, and a.

This DLP problem is popular in the cryptographic community and has been used widely as the base for other cryptographic techniques.

Our cPIR and cTIR

Getting rid of the replication requirement means that all queries may be sent to one single server to process. Without further protection, the server will find out the index. (That is one of the important reasons that replication is essential for the original PIR and our TIR schemes.)

In the new schemes, to protect these queries, the user sends $g^{q_1}, g^{q_2}, ..., g^{q_k}$ to the server and the server computation is based on these transformed queries. The computation on both sides is modulo p. Note the user keeps g and q locally. Without knowing them, the server can gain little information about the queries, subject to the computational resources the server has. That is the reason why the new schemes are described as computational.

The real challenge for the new schemes is the calculation and verification of the results. Note that the scheme aims to retrieve information from the server. Of course, the user doesn't know what the result (i.e., the intended information) is. Since the queries are now encrypted, the result is embedded in the returned answers, which are, however, also encrypted. To recover the results, our solution is based on two observations. First of all, given any $x \in G$, the user can perform polynomial time computation $g^x \pmod p$. According to the scheme, a result b satisfies the following condition: $b = g^x \pmod p$, where x is intended information.

Secondly, we assume that the range of the intended information is known. By putting them together, we have a way to recover and verify the result. That is, for a given result b, we can easily find out the intended information by computing each possible value of x and comparing it with b. If the result is reconstructed from one or more corrupted answers, the result will become invalid.

The significances of our new schemes include: 1) it requires only a single server, 2) it is a new technique for cPIR and cTIR (computational TIR) based on a different but relatively well-known computational assumption, 3) it can still work in a replication setting (this is useful in the presence of redundant resources and for the purpose of fault/intrusion tolerance). That is, the new cTIR scheme *with* replication can also achieve fault tolerance in the presence of malicious servers.

5. Future Work

To really understand the strengths and intrinsic limitations of our new techniques, we believe that extensive experimentations, both at an intranet and Internet scale, are important. Also, as one of the deliverables of the project, we plan to construct a software toolkit to make it easy for programmers to integrate our technologies with other applications.

6. Applications and Conclusions

We consider two types of applications to be the key applications of this technology. They are public repositories and centralised authorities in a VO environment. We have presented and demonstrated our technique (information-theoretic TIR) in last year's all hands meeting [YXB03] on a distributed repository case. With respect to the second type of application, we plan to integrate the technique with the Shibboleth framework. Our purpose is to demonstrate the use of less trusted centralised management authorities in a Grid computing environment.

The need for a high degree of privacy protection and security assurance very much depends on applications and context that the technologies are used. For the Grids, it is the same. Obviously, not every Grid applications require this level of protection. However, many privacy-sensitive Grid applications may potentially benefit from this technology because of the following: i) there is no need for secure communication because the entire computation is encrypted; ii) there is no need to worry about leaking of security sensitive information aftermath.

Finally, we remark on the feasibility of our cPIR and cTIR technologies. Although we have listed two applications above and have done some implementations/demonstrations, the general applicability and feasibility of these technologies to a wider range of applications are still under investigation. The real question is to what extend we can exploit these technologies for real world applications. This is probably one of the most challenging open questions for this research.

Acknowledgement

The authors are supported by the EPSRC/DTI e-Demand project.

References

[ALN87] Niv Ahituv, Yeheskel Lapid, and Seev Neumann, “Processing Encrypted Data,”, Vol. 30 (9), CACM, Sept. 1987.

[AFK89] M. Abadi, J. Feigenbaum, J. Kilian, “On Hiding Information from an Oracle,” Journal of Computer and System Science, 39(1), 21-50, 1989.

[CP02] C. Cachin and J. A. Poritz, “Secure Intrusion-Tolerant Replication on the Internet”, Proc. Intl. Conference on Dependable Systems and Networks (DSN-2002), Washington DC, USA, 2002.

[CGKS95] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, “Private Information Retrieval”, Proc. 36th Annual Symposium on Foundations of Computer Science (FOCS’95), Milwaukee, Wisconsin, USA, 23-25 Oct. 1995, pp. 41-51. Journal version: J. of the ACM, vol. 45, no. 6, 1998, pp. 965-981.

[F85] J. Feigenbaum, “Encrypting Problem Instances Or …, Can you Take Advantage of Someone Without Having to Trust Him?”, Proc. CRYPTO’85, 477-488, Springer, 1986.

[KO97] E. Kushilevitz and R. Ostrovsky, “Replication is Not Needed: Single Database, Computationally-Private Information Retrieval”, Proc. 38th Ann. IEEE Symposium on Foundations of Computer Science (FOCS’97), 1997, pp. 364-373.

[MOV97] A. Menezes, P. van Oorschot and S. Vanstone, “Handbook of Applied Cryptography,” CRC Press, 1997.

[RAD82] R. L. Rivest, L. Adleman, M. L. Dertouzos, “On Data Banks and Privacy Homomorphisms,” in Foundation of Secure Computation, R. A. DeMillo, D. P. Dobkin, A. K. Jones, and R. J. Lipton, Eds, Academic Press, New York 1982, pp. 169-179.

[YXB02b] E. Y. Yang, J. Xu and K. H. Bennett, “A Fault-Tolerant Approach to Secure Information Retrieval ”, in *Proc. 21st IEEE International Symposium on Reliable Distributed Systems (SRDS2002)*, Suita, Osaka, Japan, Oct. 2002.

[YXB03] E. Y. Yang, J. Xu, K. H. Bennett, "Sharing with Limited Trust: An Attack-Tolerant Service in Durham e-Demand Project", in Proc. UK eScience Second All-Hands Meeting, Simon J. Cox Eds, Nottingham Conference Center, U.K., Sept. 2nd-4th, 2003, ISBN 1-904425-11-9.

[ZFR02] Lidong Zhou, Fred Schneider, Robbert van Renesse, “COCA: A Secure Distributed On-line Certification Authority,” Vol. 20 (4), ACM TOCS, 2002.