Proposition de recherche doctorale

Architecture de fragmentation et défragmentation pour la protection de grandes masses de données dans de grands systèmes distribués

Mots clés :
● Directeur de thèse : Gerard Memmi
● Co-encadrant(s) :
● Unité de recherche : Laboratoire Traitement et Communication de l'Information
● Ecole doctorale : École Doctorale Informatique, Télécommunications, Électronique de Paris
● Domaine scientifique principal: Divers

Résumé du projet de recherche (Langue 1)
Résumé du projet de recherche (Langue 2)

1. Motivations With the diversification of sources of data production and the multiplication of possibilities to store them particularly in public data storage areas, it becomes quite clear that opportunities to intercept valuable data in a variety of locations where they can be observed (and altered or destroyed) are accordingly multiplied. Today, datacenters are continuously attacked, the risk is acute and data protection to achieve security and privacy is a must for any decent public or hybrid storage system. Protecting data can be achieved by strictly controlling its access or by transforming it (through encryption, anonymization, or obfuscation) and making it understandable solely by selected people who own a method or a key to reverse the transformation and getting back the initial information. This technique can become extremely costly especially with bigdata; hence the idea of adding another weapon to encryption: fragmentation. Some critical data could be fragmented and encrypted, some other data less critical would be only fragmented. Fragmentation is not a new idea. It is used in computer science in many different ways: by operating system to optimize disk space management, by database management or distributed systems to gain in performance particularly in latency, by routing algorithms in communication to increase reliability and support disaster recovery when combining replication and fragmentation together. It is also since the beginning of time used to keep a secret. We are interested by the usage of any fragmentation technique to protect data with applications in information privacy. First of all, we differentiate the notions of segment and fragment: a segment is a part separated from the rest of the information along natural lines of division; a fragment is a small part, usually broken off. In general, a segment will have an understandable meaning for a human or a machine; a fragment is a more general piece of information possibly with little or no understandable meaning when considered without a context. By fragmentation, we mean a covering (a partition would be a special case) of the information into pieces of information, each piece data naturally and format (by bi, image (bmp or jpeg), v. This can also influence the way fragmentation is performed. And data are organized in database; today we deal with noSQL database and this also must be addressed in our study. Defragmentation can also be seen as a challenge. This goes beyond the simple fact that of course, the place where defragmentation is performed is a place of high vulnerability since this where the entire information is recomposed. Defragmentation can be as tangled as resolving a jigsaw puzzle. Therefore, some design effort must be dedicated to this task. Data versioning must also be taken into account in our project since defragmenting without taking version into account could easily lead to inconsistency. Measuring performance particularly the overhead due to fragmentation, assessing risks, and quality of the information protection must be addressed for instance by comparison with classical encryption standard such as AES [AES 11]. It would be ultimately pivotal with regards to the adoption of these techniques to be able to avoid defragmenting information and perform some computation directly on fragmented data. How to perform such computation? We may want to look at multiparty computation as introduced in [Yaho 82] or searchable encryption as described in [Curtmola...11]. 2. References [AES 11] FIPS-197-Advanced Encryption Standard (AES), National Institute of Standards and Technology, Nov, 2001 [Bessani...11] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa “DEPSKY: Dependable and Secure Storage in a Cloud-of-Clouds.” EuroSys’11 Salzburg Austria April 2011. [Constantin 12] L. Constantin “Researchers Identify Stuxnet-like Cyberespionage Malware Called ‘Flame’ PCWorld, May 2012. [Curtmola...11] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky “Searchable symmetric encryption: Improved definitions and efficient constructions.” J. of Computer Security 19(5): 895-934 (2011) [Gen 09] C. Gentry “A Fully Homomorphic Encryption Scheme” Stanford U. PhD, September 2009 [Shah...11] J. Shah and V. Saxena Video Encryption: A Survey, International Journal of Computer Science Issues,Volume 8, Issue 2, 2011 [Shamir 79] A. Shamir “How to share a secret”. Communications of the ACM 22 (11): 612–613, 1979 [Wang ...14] W. Wang, Y. Hu, L. Chen, X. Huang, and B. Sunar “Exploring the Feasability of Fully Homomorphic Encryption” to appear in IEEE Trans on Computers [Yaho 82] A C Yao, Protocols for Secure Computations (extended abstract) Proceedings of the 21st Annual IEEE Symposium on the Foundations of Computer Science, pp 160-164, 1982.
Technical challenges identification A public environment is particularly vulnerable. Many users can access to many servers loading persistent software that later can observe in a stealth manner without being easily detected [Constantin 12]. We have to assume that computation or data is observed on at least one unknown machine: how to ensure that: never all fragments are stored or even transit by the same machine. If that was the case fragmentation would be almost useless: only the order in which fragment pass by in front of an observer could still confuse him. Fragmenting can be considered as a simple idea; however, it is not so simple to efficiently design and develop it in order to comply with both the motivations we just described and the avoidance of an important overhead. One key question lies in the way fragments are built in order to get the observation of one fragment inconsequential. At the same time, it would be nice to fragment with enough redundancy to address an availability requirement such as: the loss of one fragment is inconsequential. We are here hinting that an effective fragmentation should not be a partition. It must be said that data nature and format (text, image (bmp or jpeg), video) can also influence the way fragmentation is performed. Data are organized in database; today we deal with noSQL database and this also must be addressed in our study. Defragmentation can also be seen as a challenge. This goes beyond the simple fact that of course, the place where defragmentation is performed is a place of high vulnerability since this where the entire information is recomposed. Defragmentation can be as tangled as resolving a jigsaw puzzle. Therefore, some design effort must be dedicated to this task. Data versioning must also be taken into account in our project since defragmenting without taking version into account could easily lead to inconsistency. Measuring performance particularly the overhead due to fragmentation, assessing risks, and quality of the information protection must be addressed for instance by comparison with classical encryption standard such as AES [AES 11]. It would be ultimately pivotal with regards to the adoption of these techniques to be able to avoid defragmenting information and perform some computation directly on fragmented data. How to perform such computation? We may want to look at multiparty computation as introduced in [Yah0 82] or searchable encryption as described in [Curtmola…11].