Management of digital records with RADAR by data dilution into Complex System

Anne Jeannin-Girardon
ICube laboratory, University of Strasbourg, UMR CNRS 7357, Strasbourg, France,
anne.jeannin@unistra.fr
Alexandre Bruyant
ICube laboratory, University of Strasbourg, UMR CNRS 7357, Strasbourg, France,
alexandre.bruyant@etu.unistra.fr
Nicolas Toussaint
Complex System Digital Campus UNESCO Unitwin, Manchester Metropolitan University, UK,
n.toussaint@mmu.ac.uk
Ismaila Diouf
Complex System Digital Campus UNESCO Unitwin, Cheikh Anta Diop University, Dakar – Senegal,
isma.diouf@gmail.com
Pierre Collet
ICube laboratory, University of Strasbourg, UMR CNRS 7357, Strasbourg, France, pierre.collet@unistra.fr
Pierre Parrend
ICube laboratory, University of Strasbourg, UMR CNRS 7357, ECAM Strasbourg-Europe, France,
pierre.parrend@unistra.fr

Abstract

Storing sensitive data in a centralized way can lead to significant loss, should the central node or networks links be defective as is the case in numerous countries, especially developing countries. Consequently, this paper proposes to deal with this problem by diluting sensitive data into an ecosystem of machines thanks to a platform called RADAR (Robust Anonymous DAta Records) that automatically fragments, encrypt, replicates and distributes data all over the network for anonymized, robust, encrypted storage and backup. RADAR gets nodes of a network to cooperate for storage and backups of sensitive data, even in case of non-reliable IT infrastructure.

Keywords: Epidemic replication, distributed storage, complex systems, data dilution, anonymity

1. Introduction

In some of the most important countries around the world such as India, China but also in Africa, Internet access is still precarious and therefore not reliable. This means that it is difficult for such nations to setup and use digital platforms that would help them to bridge the gap with highly connected countries which, on their side, are plagued with the opposite problem: having a very reliable network infrastructure makes them want to develop deterministic and perfect data management systems that can simply not be implemented in practice. The RADAR system presented in this paper proposes a more pragmatic approach, that can overcome the problems inherent to an unreliable network while at the same time, providing a reliable enough *Quality Of Service* (QOS) that may satisfy the demands of high-tech countries.

2. Related work

Distributed databases all use data replication to improve robustness.

Most approaches focus on improving access time, bandwidth consumption while guaranteeing that absolutely no piece of data can be lost. This is very costly and very difficult to achieve. Paper [1] proposes a hierarchical approach to ensure replicas are consistent over the network. The cluster of databases used to implement this approach stores newest data at the topmost level, while the lowest levels propagate replicas from one level to the next in an epidemic way. This allows them to define a range of QOS that is related to the consistency level that is required by the client application. Epidemic replication protocol is widely used to replicate data consistently in distributed databases. This approach is evaluated in [2]: in practice, it is very difficult to propose an analytic methodology to evaluate the variables associated with epidemic replication (such as bandwidth consumption, replicas consistency, ...) so the authors propose a simulator allowing one to optimize the parameters of their replication algorithms that scale with their network configuration. Across large scale networks, such as data grids, hybrid approaches are used, such as in [3]. In order to ensure efficient access to the distributed data of the grid, a scalable replica management system (using both hierarchical and flat topologies) and a dynamic data distribution algorithm are used jointly. The hierarchical topology contains nodes propagating replicas to direct ancestors/children while the flat topology propagates replicas using a Peer-to-Peer protocol. Such protocols are also widely used in distributed databases systems. Data in Peer-to-Peer systems are often assumed to be static and not subject to many updates. But, for many applications, data are frequently added, deleted or updated. In paper [4], the authors propose an epidemic replication method dedicated to Peer-to-Peer systems. One main characteristic of Peer-to-Peer systems is that no assumptions can be made on whether the machines of the network are online or not. Using rumor spreading algorithms [5] and push/pull methods so that peers can request updates or send updates to peers that sent a request update, the authors ensure some level of consistency for the data and achieve low latencies.

The work presented in this paper proposes to protect and increase the robustness and security of sensitive data by chopping it into several pieces, encrypting the pieces, replicating the pieces and sending the replicas randomly among a group of machines belonging to the RADAR network, resulting in a “dilution” of the original data in a Complex System (a number of autonomous machines interacting together to create a network architecture, whose emergent properties are robustness and security). The RADAR model relies on Complex Systems principles and does not need to define complicated hierarchical architectures or algorithms in order to ensure data consistency. No assumption are made about the level of reliability of the network on which RADAR is deployed, allowing unreliable infrastructures to use RADAR for distributed data storage. Moreover, the model ensures security and can thus be used for sensitive data storage such as medical records.

3. Proposed RADAR model

Complex Systems

A *Complex System* can be defined as a set of autonomous entities in interaction, exhibiting multi-level emergent behaviour where (as Aristotle said some 2300 years ago) *the whole is more than the sum of the parts*. The typical example is turbulence. The local behaviour of particles results at a higher scale in the apparition of swirls and vortices that can be described using Navier-Stokes equations. However these equations do not appear at all at the particle level.

RADAR implements storage entities that have no global picture of the network, but that as a whole, behave like a reliable and secure data storage system: machines rely on each other to automatically backup their data.

Modus operandi

The whole RADAR system is composed of many (ideally more than 10) loosely interconnected machines. Because the network is deemed unreliable, machines never “expect” anything from other machines unless they need to recover from a complete data loss. Therefore, under normal operation mode, machines:

- backup their own data by sending it to *n* other machines they know about in the RADAR network (using a non-connected protocol such as UDP, because connected mode could request a too high QOS from the available internet network),
- accept encrypted pieces of data from other machines of the RADAR network for backup.

Whenever a machine needs to store a new local piece of data, it is (depending on the required level of security of the data) possibly chopped into several pieces that are timestamped, encrypted using the local cryptographic key of the machine, replicated and depending on the required level of reliability, sent to *n* other machines randomly chosen in the network (the more required reliability, the greater *n*) for backup, thus preserving the anonymity of the data owner since full information cannot be retrieved from a fragment. Fragmentation of sensitive data can either be based on semantics, or simply size. For instance, in our current application, medical records are fragmented into:

PPD: patient personal data (name, address, telephone number, e-mail address, ...),

MTD: medical textual data (sex, date of birth, weight, blood pressure, a.s.o. along with symptoms, diagnostics, medication, ...),

DMD: digital medical data (medical imaging, audiogram, electrocardiogram, ...),

PLD: private life data not related to medical condition (death of brother, ...) for a personal follow-up of the patient.

In order to further ensure data anonymity, for each patient, each fragment is encrypted and sent separately to *n* machines chosen at random among the known machines. This means that if data is stolen on a given machine, patient data is not compromised because for a particular patient, each machine will keep only one of the 4 (encrypted) fragments. For instance, if the MTD fragment is stolen, the attacker has no information about the personal data of the patient such as his/her name.

Security

- If a hacker gets hold of the cryptographic key of one machine and gets access to this machine, only the local data is compromised. All other data stored as a backup for other machines remain protected because it is encrypted with the cryptographic keys of the other machines.
- If a hacker gets hold of all the cryptographic keys of all the machines in the network, and physically gets access to one machine, the data of the machine is compromised (as above) and the hacker will be able to read all the fragments stored by the machine as a backup for other machines. However, because the different fragments have been randomly sent to different machines, it is highly unlikely that all 4 PPT, MTD, DMD and PLD fragments for one patient will be found in the backup files hold by one machine.

To sum up, with RADAR, the only way to have access to a particular patient's record that is not locally stored on one machine is to have full access (with passwords) to enough machines in the network to hope to gain access to all four fragments belonging to a particular patient, which in practice is not feasible if the network is made of hundreds of machines situated in different places (hospitals, medical centers spread across the country).

Hybrid encryption

RADAR uses the RSA [6] asymmetric encryption algorithm with 2048 bits keys: it is an asymmetric algorithm that uses a public key to encrypt the data, and a private key for decryption. Unfortunately, this algorithm is computationally intensive and cannot natively encrypt data that are longer than its key (2KB in our case). In order to circumvent these limitations, we use RSA in conjunction with AES [7], a much faster but symmetric algorithm (the same key must be used for encryption and decryption).

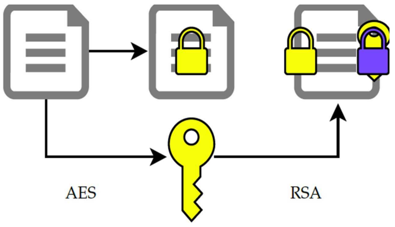

Fig. 1. Hybrid encryption used by RADAR

The data is encrypted via AES by a randomly generated key, that is itself encrypted with RSA (cf. Fig. 1).

This makes it possible to verify the origin and the integrity of the data. A hash code is generated using the data to be signed. The code is encrypted thanks to the private key of the sender. The receiver can verify the authenticity of the signature by deciphering the hash code and by comparing it with a new code generated on the data.

Each node receives an RFC 5280 signed certificate delivered by the administrator of the network, that guarantees that the node is indeed part of the network.

Data loss and recovery

RADAR comes in when a local database is completely lost and not recoverable. What happens next is that the person in charge with the database that has disappeared contacts the administrator of the network, asking for a database rebuild. When the new machine is ready, it is added to the network again, but in recovery mode. The rebuild starts with one machine of the network sending to the other machines it knows (epidemic message transmission) the message that all files that have been received in the past from the failed machine must be sent back to it. The machine in recovery mode then starts to receive from the whole network all the data it previously sent. It uses the timestamps of the data to reconstruct its database in a consistent way. During the recovery, the machine keeps an account on the number of replicas it has received for all its data. After recovery, the machine sends again the missing number of replicas of a given data to random machines, so that it sums back to n.

Data loss probability

It is very difficult to estimate the probability that when a machine has lost all its data, all n copies of a given patient record are also lost at the same time. Computing this probability is so complex that most similar data replication systems use network simulation tools such as Chive [2] and Network Simulator (NS) [8] to run empirical tests to evaluate the robustness of their approach. Using the same methods shows that for most realistic RADAR configuration and failure probabilities, it is very difficult to effectively lose data.

Expected disk consumption

Depending on n, the number of replications that is chosen, for the reliability of a particular network, expected disk consumption is in average n+1 times the disk consumption of a single machine. Indeed, if one machine sends n copies of its data to other machines in the network, this means that it will receive data from (in average) n other machines in the network. Scaled to the medical information system of Senegal, (for which RADAR was initially designed) the system could use around 1000 medical centers (with at least one machine per center) to deal with a population of 15M inhabitants. With a uniform distribution of patients per center (which is of course not the case), each machine would store the medical records of 15k local patients. With n=3, this means that each machine would need to store the equivalent of 45k medical records, which is a reasonable amount by 2018 disk storage space. This is to be compared with the cost needed to create, maintain and operate a data center capable of dealing with all 15 million medical records of the country in a reliable and secure way over an unreliable Internet network.

4. Conclusion

Where most other systems focus on perfect consistency and reliability resulting in over-complex time consuming algorithms which, eventually, are not 100% bullet-proof, RADAR accepts that data may be lost, but focusses on reducing such occurrence to very low probability. It offers a system that is simple, while robust enough to work over an unreliable network and ensures data is stored anonymously when dealing with sensitive data such as medical records. RADAR is currently tested on a huge patient cohort of more than 300 000 patients over 60 nodes in India. Following work will concentrate on reliability thanks to large scale testing using network simulation tools.

Acknowledgements

The authors thank the N-Light endowment fund that funds this project, as well as the JIVA Ayurveda Indian company, with which RADAR is currently tested.

References

1. I. Arrieta-Salinas, J. E. Armend'riz-Iñigo and J. Navarro, "Classic Replication Techniques on the Cloud," *2012 Seventh International Conference on Availability, Reliability and Security*, Prague, 2012, pp. 268-273.
2. A. Jiménez-Yáñez, J. Navarro, F. D. Muñoz-Escoí, I. Arrieta-Salinas and J. E. Armendáriz-Iñigo, "Chive: A simulation tool for epidemic data replication protocols benchmarking," *2014 9th International Conference on Software Engineering and Applications (ICSOFT-EA)*, Vienna, Austria, 2014, pp. 428-436.
3. H. Lamehamedi, B. Szymanski, Z. Shentu and E. Deelman, "Data replication strategies in grid environments," *Fifth International Conference on Algorithms and Architectures for Parallel Processing, 2002. Proceedings.*, Beijing, China, 2002, pp. 378-383.
4. A. Datta, M. Hauswirth and K. Aberer, "Updates in highly unreliable, replicated peer-to-peer systems," *23rd International Conference on Distributed Computing Systems, 2003. Proceedings.*, 2003, pp. 76-85.
5. Laijun Zhao, Hongxin Cui, Xiaoyan Qiu, Xiaoli Wang, Jiajia Wang, SIR rumor spreading model in the new media age, In Physica A: Statistical Mechanics and its Applications, Volume 392, Issue 4, 2013, Pages 995-1003
6. Xin Zhou and Xiaofei Tang, "Research and implementation of RSA algorithm for encryption and decryption," *Proceedings of 2011 6th International Forum on Strategic Technology*, Harbin, Heilongjiang, 2011, pp. 1118-1121.
7. D. M. Alghazzawi, S. H. Hasan and M. S. Trigui, "Advanced Encryption Standard - Cryptanalysis research," *2014 International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, 2014, pp. 660-667.
8. C. Leng, M. Lehn, R. Rehner, and A. Buchmann, “Designing a testbed for large-scale distributed systems,” in Proceedings of the ACM SIGCOMM 2011 conference on SIGCOMM, New York, NY, USA, 2011, pp. 400-401.