Autonomous Aggregate Data Analytics in Untrusted Cloud

Ganapathy Mani*, Denis Ulybyshev‡, Bharat Bhargava†
Department of Computer Science & CERIAS
Purdue University West Lafayette, USA
*manig@purdue.edu**, *dulybysh@purdue.edu*‡, *bbshail@purdue.edu*†

Jason Kobes‡‡, Puneet Goyal††
Northrop Grumman Corporation, McLean, USA‡‡
Dept. of Computer Sci. & Eng., IIT Ropar, India ††
Jason.Kobes@ngc.com‡‡, *puneet@iitrpr.ac.dot.in*††

***Abstract*—Intelligent Autonomous Systems (IAS) are highly reflexive and very cognizant about their limitations and capabilities, interactions with neighboring entities, as well as the interactions with its operational environment. IAS should be able to conduct data analytics and update policies based on those analytics. These tasks should be performed autonomously i.e. with limited or no human intervention. In this paper, we introduce advanced aggregate analytics over untrusted cloud and autonomous policy updates as a result of those analytics. We will be using Active Bundle (AB), a distributed self-protecting entity, wrapped with policy enforcement engine as our implementation service. We propose an algorithm that can enable individual ABs to grant or limit permissions to their AB peers and provide them with access to anonymized data to conduct analytics autonomously. When these processes take place, ABs do not need to rely on policy enforcement engine every time, which increases scalability. This workflow also creates an AB environment that is decentralized, privacy-preserving, and autonomous.**

***Keywords*-aggregate analytics; cognitive autonomy; cloud; autonomous systems; privacy;**

I. Introduction

Decentralized Artificial Intelligence (DzAI) in a cloud environment is nothing but methodology that concerns about the behavior of an autonomous entity in a multi-entity world [1]. IAS systems should act individually and react to their interactions with other entities in the environment.To minimize the complexities and communication, computation, and storage overheads, IBM proposed Automatic Computing Initiative with the goal of developing autonomous entities that can handle themselves [2]. Google proposed a distributed storage system known as Bigtable [3] to store structured data in a distributed manner. Aggregated analytics were conduced over that distributed database system by individual entities (clients).

Similarly, Active Bundle (AB) [4] is designed with a policy enforcement engine and it is able to manage itself and authenticate clients. But the policies are defined and set manually at the design phase. A distributed entity needs to perform data analytics and learning on its own with limited or not human intervention to become an autonomous entity. In this paper, we propose autonomous aggregate analytics over untrusted cloud with an aiding workflow of obtaining authentication from peer ABs, just once. That authentication certificate will be used to access other ABs data (anonymized through perturbation) to conduct meaningful analytics and make decisions based on the results. We will discuss a details of this authentication algorithm in a later section.

Our proposed workflow relies on Active Bundle (Figure 1) [4] [5] [6] for secure data exchanges between entities in autonomous system. Active Bundle is a self-protecting structure that incorporates encrypted data, access control policies and policy enforcement engine (a Virtual Machine (VM) loaded with policies that are set at the design time by the administrator). Data is stored in the form of key-value pairs with encrypted values. Each data item is encrypted with a separate AES symmetric key, which is generated on-the-fly based on execution flow. Data request to an Active Bundle follows the following steps: (a) authentication, (b) client attributes evaluation, including trust level and cryptographic capabilities of the browser [7] [8] [9], and (c) access control policies evaluation. Accessible values from key value pairs are decrypted with the derived decryption keys. Active Bundle is written in Java and implemented as a JAR-file. Details of communication between service and Active Bundle are covered in [4] [5] [6].

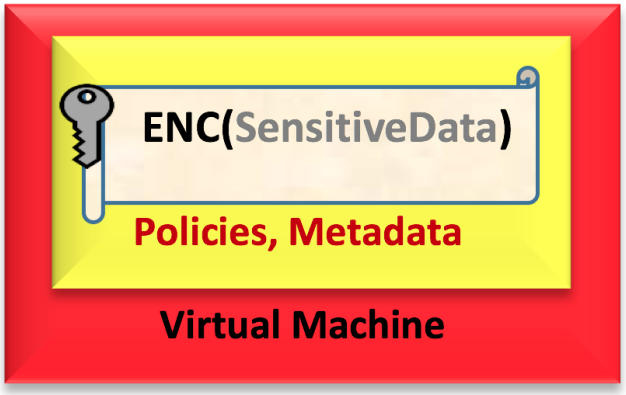

Figure 1. Architecture of Active Bundle

Every served data request is recorded in the log file, that can be stored either on central server or on a client. Provenance record contains the following fields:

- data recipient—the client: in our test case of ABs with medical data, its either data owner (usually a patient), doctor, or an insurance company representative.

- data origin—entry performed by a specific entity (doctor, patient, or insurer).
- type of data has been requested—changes depends on the request. Access control policies enforce the encapsulation of data.
- time of the request
- request result: approved or denied (enforced by policy enforcement engine).

The rest of the paper is organized as follows: section II describes related work that has been done using AB as well as aggregated analytics. Section III describes the autonomous aggregate analytics model with authentication mechanism. It also discusses the perturbation of data to maintain anonymity and preserve privacy. Section IV presents our evaluation of the AB's one time authentication vs. each time authentication in order to conduct aggregate analytics. Finally, we discuss our future work and conclude in section V.

II. Related Work

There are numerous autonomous models available to make the entities self-manage themselves through governing policies. In [10], the authors introduce a policy based management system Niche that enables and supports self-management. In this framework, the administrators can set up several policy enforcers called Policy-Manager-Group to avoid centralized policy decision making. In [11], the authors propose an aggregated analytics over distributed entities in the cloud. The authors utilize service-oriented decision support system (DSS in cloud) and make the model scalable for handling aggregate analytics over Big Data systems. Similarly, AB is also a service oriented architecture in the could. AB was designed as a privacy preserving mechanism during information dissemination in untrusted environments [12]. AB architecture gradually evolved to support Product Cycle Management (PCM) as a third party where the information is distributed and disseminated securely [13]. In this model, AB will act as an autonomous entity authenticating peer ABs to perform meaningful analytics and reduce overhead while increasing its scalability.

III. Autonomous Aggregate Analytics

The model constitutes a mechanism that is twofold: (1) authenticate requesting ABs by the same type of trustworthy AB(s) and (b) perform meaningful aggregated analytics at the same time preserving the privacy of data owners (sensitive data of owners is saved in ABs).

A. Authentication

One of the overheads of AB is the computation time complexity that comes with its policy enforcement engine. Every time, when the AB is accessed it will have to go through its policy enforcement engine and check all policies. It introduces redundancy and usability as well as scalability

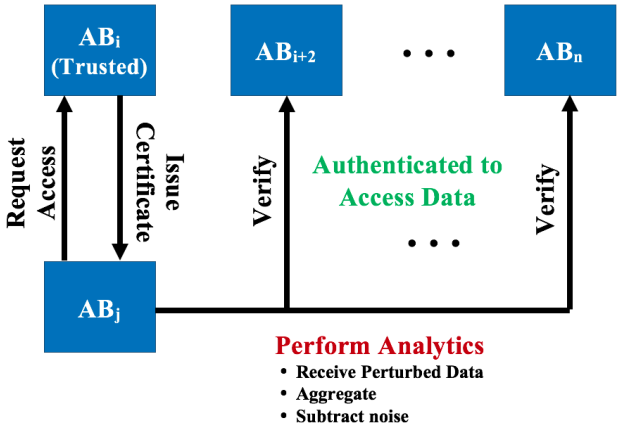

Figure 2. AB's efficient authentication protocol

are reduced. Figure 2 shows a solution to overcome the policy engine overhead.

A trusted AB of the same type (AB_1 and AB_2 have the same policies) can be given the task of issuing a verification certificate to requesting AB. Trust parameter can be user defined such as AB_1 was created longer than AB_2 and AB_1 was never compromised. Once AB_1 verifies AB_2's credentials then it issues an authentication certificate. If they are both of the same type, yet the trust level of the authenticator is not greater than the requester, then the certificate will contain restrictions such as "only provided encrypted data". In this case, the requester should know the private key to decrypt the data and perform analytics. If both scenarios are not met, the request is simply denied. Algorithm 1 describes the generic structure of the process.

Data: AB_i and AB_j as inputs
Result: Certificate issued/denied/issued with restrictions
if *Type(AB_i) is same as Type(AB_j)* **then**
  if *Trust(AB_i) is greater than Trust(AB_j)* **then**
    Generate authentication certificate;
    Issue the certificate to AB_j;
  else
    Generate Certificate with restrictions (only access encrypted data);
  end
else
  Deny the request;
  Report to administrator;
end

Algorithm 1: AB Authentication Protocol

This protocol facilitates AB to scale up for large number of datasets and useful analytics can be performed without the overhead of policy enforcement engine as well as its communication, computation, and storage overhead.

B. Privacy-Preserving Aggregate Data Analytics

Each authenticated AB can perform individual aggregated analytics such as Count, Average, etc. on qualified attributes such as age or number of medications. To show this with an example, consider AB_1 wants to get average age stored in all the ABs of the system and assume that AB_1 is issued a certificate. AB_1 performs the following perturbation to make the data anonymous:

$$Total = (Age_1 + R) + Age_2 + ... + Age_n$$

Here R is a random number added to the age. This perturbed data will be sent to AB_2 where it adds the real value of the age to the total. This continues as a circular linked list data structure. When the total comes back to AB_1, the following operation takes place:

$$Average = \frac{(Total - R)}{n}$$

Based on this perturbation, it can be guaranteed that the real age of each AB cannot be known. Thus preserving the privacy of individual ABs. The aggregations also happen autonomously where any AB can initiate aggregate analytics at anytime and make decisions based on the results.

C. Autonomous Active Block

Autonomous Active Block (2AB) is a modification of AB where encrypted data module is enclosed by adaptive policy module, which will be influenced by data analytics and decision engine.

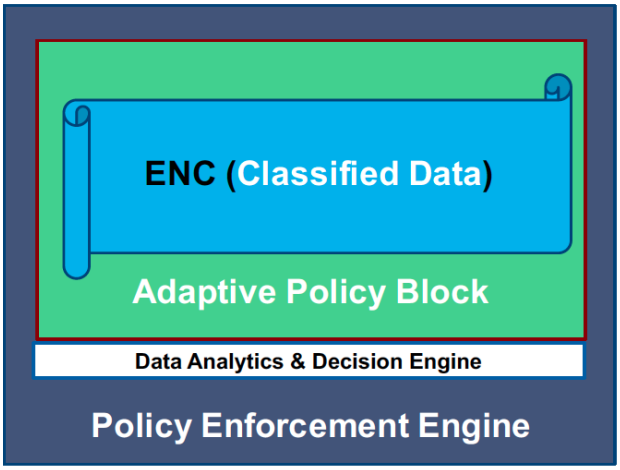

Figure 3. Architecture of Autonomous Active Block

Figure 3 shows the architecture of 2AB. It has the following components:

Classified data:

- One-way public key encrypted data

Adaptive Policy Block (APB)

- Describes Autonomous Active Block and its access control policies
- Policies manage Autonomous Active Block interaction with services and hosts

Data Analytics and Decision Engine:

- Conducts aggregated analytics and influences policies

Policy Enforcement Engine:

- Enforces policies specified in APB

2AB will be used to study and make changes to the policies autonomously.

IV. Evaluation

In this experiment, we measure latency of data request sent to Active Bundle, which is hosted by a local Server, located in the same network with the client. Server that hosts Active Bundle has the following characteristics:

1) Hardware: MacBook Pro
   Intel Core i7 CPU @ 2.2 GHz
   16GB DRAM
2) OS: macOS Sierra 10.12.6.

Client requests numerical integer field 'ID' from the Active Bundle that incorporates four access control policies. Hosting server runs the Active Bundle and listens to the opened port 5555. We measure Round-trip Time (RTT) for data request processing at the server side and do not consider network delays between client and server in this experiment. ApacheBench, ver.2.3 utility is used for RTT measurements. We run 50 requests in a row and compute average RTT. Detailed prototype tutorial is available [14]. Demo video [15] illustrates the Active Bundle concept.

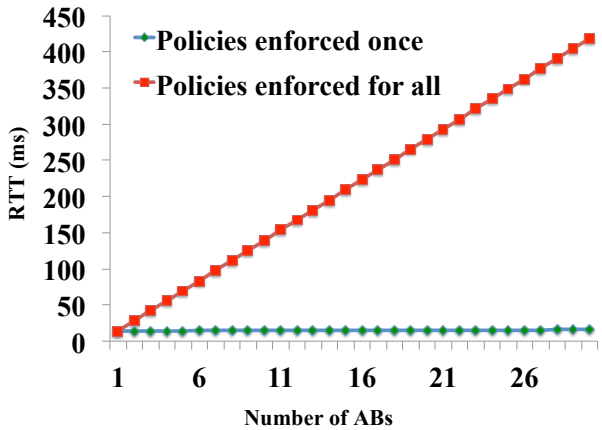

Figure 4. AB with one time policy enforcement + AB with each time policy enforcement

The figure 4 shows the overhead acquired by ABs when they each time enforce the policies through policy enforcement engine. We used python script to compute the time

for reading and calculating average of 30 ABs' ages in unencrypted file. We added those computation times to the one time average of 50 requests (the plotted average RTTs are 50 $\times$ 30 ABs = 1500 iterations). The graph shows that the increase in overhead is exponential when it comes to enforcing policies over and over again for large number of iterations.

V. Conclusion

In this paper, we proposed a novel and efficient mechanism for authentication in a distributed cloud through AB service architecture. Our model uses a peer-to-peer authentication system that significantly reduces policy enforcement overhead. The aggregate analytics model allows any AB to initiate aggregate queries at anytime across the AB service oriented architecture and perform some very useful data analytics. Since the data is perturbed, the privacy of each AB is well-preserved. By imposing constraints on untrustworthy ABs, the model provides robust security to the distributed cloud environment. We provided the overhead of normal AB without our authentication model as experimental results.

In the future, we intend to quantify trust using machine learning models. We will also implement autonomous policy updates based on the data analytics that are performed by each AB.

Acknowledgment

This research work is supported by NGC Research Consortium.

References

[1] Y. Miiller, "Decentralized artificial intelligence," *Decentralised AI*, pp. 3–13, 1990.

[2] P. Horn, "Autonomic computing: Ibm\'s perspective on the state of information technology," 2001.

[3] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A distributed storage system for structured data," *ACM Transactions on Computer Systems (TOCS)*, vol. 26, no. 2, p. 4, 2008.

[4] R. Ranchal, "Cross-domain data dissemination and policy enforcement," 2015.

[5] L. B. Othmane, *Active bundles for protecting confidentiality of sensitive data throughout their lifecycle*. Western Michigan University, 2010.

[6] L. Lilien and B. Bhargava, "A scheme for privacy-preserving data dissemination," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 36, no. 3, pp. 503–506, 2006.

[7] D. Ulybyshev, B. Bhargava, M. Villarreal-Vasquez, A. O. Alsalem, D. Steiner, L. Li, J. Kobes, H. Halpin, and R. Ranchal, "Privacy-preserving data dissemination in untrusted cloud," in *Cloud Computing (CLOUD), 2017 IEEE 10th International Conference on*. IEEE, 2017, pp. 770–773.

[8] "W3c web cryptography api," 2018. [Online]. Available: https://www.w3.org/TR/WebCryptoAPI/

[9] "Web authentication: an api for accessing scoped credentials," 2018. [Online]. Available: http://www.w3.org/TR/webauthn/

[10] L. Bao, A. Al-Shishtawy, and V. Vlassov, "Policy based self-management in distributed environments," in *Self-Adaptive and Self-Organizing Systems Workshop (SASOW), 2010 Fourth IEEE International Conference on*. IEEE, 2010, pp. 256–260.

[11] H. Demirkan and D. Delen, "Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud," *Decision Support Systems*, vol. 55, no. 1, pp. 412–421, 2013.

[12] P. Angin, B. Bhargava, R. Ranchal, N. Singh, M. Linderman, L. B. Othmane, and L. Lilien, "An entity-centric approach for privacy and identity management in cloud computing," in *Reliable Distributed Systems, 2010 29th IEEE Symposium on*. IEEE, 2010, pp. 177–183.

[13] R. Ranchal and B. Bhargava, "Protecting plm data throughout their lifecycle," in *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*. Springer, 2013, pp. 633–642.

[14] D. Ulybyshev[1], B. Bhargava[1], L. Li, J. Kobes, D. Steiner, H. Halpin, B. An[1], M. Villarreal[1], R. Ranchal, and T. Vincent[1], "Secure dissemination of ehr in untrusted cloud."

[15] D. Ulybyshev, "Secure dissemination of ehr demo video," 2018. [Online]. Available: https://www.dropbox.com/s/4wg3vuv52j4s16v/NGCRC-2017-Bhargava-Demo1.wmv?dl=0