AFRL-AFOSR-VA-TR-2016-0101

Ecologically Inspired Framework for Assured Data Cloud

Murat Kantarcioglu
UNIVERSITY OF TEXAS AT DALLAS

02/24/2016
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ RTA2
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
06-03-2016

2. REPORT TYPE
Final Performance

3. DATES COVERED *(From - To)*
01-04-2012 to 30-09-2015

4. TITLE AND SUBTITLE
Ecologically Inspired Framework for Assured Data Cloud

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-12-1-0082

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
Murat Kantarcioglu

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
UNIVERSITY OF TEXAS AT DALLAS
800 W CAMPBELL RD
RICHARDSON, TX 75080 US

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AF Office of Scientific Research
875 N. Randolph St. Room 3112
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR RTA2

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
A DISTRIBUTION UNLIMITED: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
As a part of this project, we explored risk based approaches for securely managing data in the cloud. Basically, inspired by how living organisms manage risks in nature while preserving and conserving energy, we developed novel risk based approaches that balances risk (i.e., potential sensitive data disclosure risk), computational cost (i.e., how long will it take to run the security enhanced tasks in the cloud?), monetary cost (i.e., how much more you would pay to the cloud service provider due to security enhanced cloud computing?). In different settings, we solve the variants of the multi-objective optimization framework where we find best query execution plan Q among all possible query plans that minimizes the total run time while it does not exceed the predefined monetary costs and risk measures.

15. SUBJECT TERMS
Securty, Private Clouds

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Murat Kantarcioglu
Unclassified	Unclassified	Unclassified	UU		**19b. TELEPHONE NUMBER** *(Include area code)* 972-883-6616

AFOSR Final Performance Report

Project Title:	Ecologically Inspired Framework for Assured Information Cloud
Award Number:	FA9550-12-1-0082
Start Date:	04/30/2012
Program Manager:	Tristan N Nguyen

Principal Investigators:

Murat Kantarcioglu
University of Texas at Dallas
www.utdallas.edu/~muratk
muratk@utdallas.edu

Co-Principal Investigators:

Bhavani Thuraisingham
University of Texas at Dallas
http://www.utdallas.edu/~bhavani.thuraisingham/
bxt043000@utdallas.edu

Alain Bensoussan
Universit of Texas at Dallas
http://www.utdallas.edu/~axb046100/
axb046100@utdallas.edu

1. Summary of Accomplishments:

As a part of this project, we explored risk based approaches for securely managing data in the cloud. Basically, inspired by how living organisms manage risks in nature while preserving and conserving energy, we developed novel risk based approaches that balances risk (i.e., potential sensitive data disclosure risk), computational cost (i.e., how long will it take to run the security enhanced tasks in the cloud?), monetary cost (i.e., how much more you would pay to the cloud service provider due to security enhanced cloud computing?). In different settings, we solve the variants of the following multi-objective optimization framework where we find best query execution plan Q among all possible query plans that minimizes the total run time while it does not exceed the predefined monetary costs and risk measures.

$$
\begin{aligned}
&\text{minimize}_{Q \in Q_p} && RunTime(Q) \\
&\text{subject to} && (1)\ MonetaryCost(Q) \le M \\
& && (2)\ Risk(Q) \le R
\end{aligned}
$$

To our knowledge, as also reported in Network World Magazine[1], this is the first framework that integrates rigorous risk management tools "…that meets the conflicting goals of performance, sensitive data disclosure risk and resource allocation costs getting weighed and balanced.". The framework proposed as a part of this proposal resulted in numerous publications in top security, data management and cloud computing venues and already received hundreds of citations according to Google Scholar. We summarize the general applicability of the above framework by briefly discussion how it is applied in two very different application settings. In addition to following examples and contributions, we developed the first encrypted key-value store that supports efficient search and access control capabilities for hybrid clouds.

2. Risk-based Query Processing in Hybrid Clouds [3]

An emerging trend in cloud computing is that of hybrid cloud. Unlike traditional outsourcing where organizations push their data and data processing to the cloud, in hybrid clouds in-house capabilities/ resources at the end-user site are seamlessly integrated with cloud services to create a powerful, yet cost-effective data processing solution. Hybrid cloud solutions offer similar benefits as traditional cloud solutions. Yet, they provide advantages in terms of disclosure control and minimizing cloud resources given that most organizations already have an infrastructure they can use. Exploiting such benefits, however, opens numerous questions, the foremost of which is how should one split the data and computation between the public and private sides of the infrastructure? Different choices have different implications from the perspectives of sensitive data disclosure, computational performance and resource allocation

[1] Christine Burns Rudalevige, "Hybrid clouds pose new security challenges", Network World http://www.networkworld.com/article/2163059/cloud-computing/hybrid-clouds-pose-new-security-challenges.html

costs. On one extreme, one may choose to outsource the entire data and workload to the public cloud (as is typical to outsourcing solutions). While simple to implement, such a solution, incurs the highest resource allocation cost in terms of cloud service (both storage and computing), and is most vulnerable to data leakage. In addition, the outsourcing strategy may not even be optimal in terms of performance since it wastes local resources which are now unused. An alternate strategy might be to replicate data at both, the private and public sides, and to split the workload between the two sides. While simple queries may be computed on the private side, the complex ones can be performed over the public infrastructure. The above strategy exploits local resources, and thereby reduces the cost of the required cloud services. However, the resource allocation cost and the amount of sensitive data that is exposed to the public cloud will be maximum in this case.[2] Another possibility could be to only replicate some part of the data to the public side so as to enable the distribution of the computation while limiting the disclosure risks and resource allocation costs to the desired thresholds. The possibilities described above are just three of the multitude of computation partitioning choices. The third option seems to be the best one in terms of various end-user requirements such as performance, costs, and sensitive data exposure. An observation to be made here is that as different variants of the computation partitioning problem are formulated, a myriad of design choices present themselves. These choices are based on various data and workload formats (dynamic queries or batch jobs), as well as different query execution techniques over hybrid clouds.

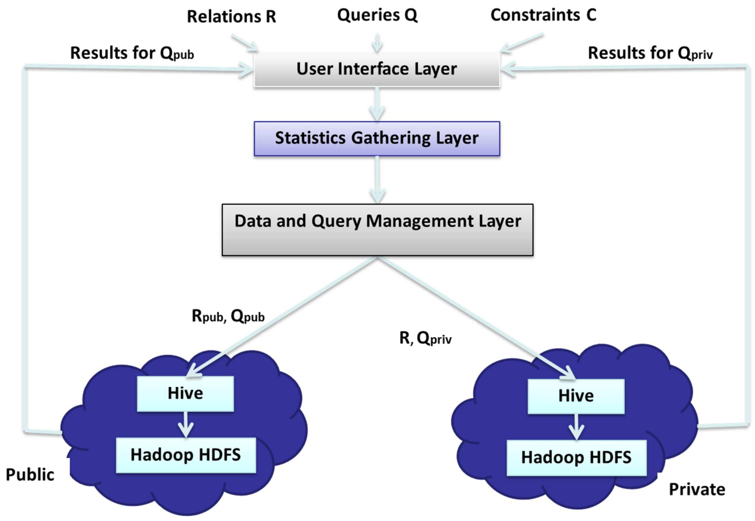

Figure 1: Proposed Architecture

In this specific work, we formalized our generic risk management framework for the computation (and the implied data) partitioning problem for hybrid clouds and developed a framework for splitting data processing tasks such that the desired goals of performance, disclosure risk and monetary expenses are achieved. In particular, given a workload of jobs (specifically SQL style HIVE) the underlying dataset (assumed to be relational) and the machine characteristics of private and public clouds, we proposed a dynamic programming approach to solve the computation partitioning problem.

[2] Here public cloud could be considered as untrusted larger cloud infrastructure, and private cloud could be considered, small but trusted infrastructure. Therefore, proposed solution could be used to reduce the trust in a given infrastructure.

2.1.Proposed Architecture

Overview of our proposed system architecture are given in Figure 1. The system mainly consists of two components: The Statistics Gathering layer performs the task of statistics collection over the dataset and query jobs, while the Data and Query Management layer decides on the data and workload partitioning for the given set of queries. Our focus in this work was on the Data and Query Management layer of the system, though, as will become clear, statistics gathering is essential to determine optimal query workload and data distribution.

A user starts by submitting a set of relations, $R = \{R_1,R_2 \cdot \cdot \cdot ,R_m\}$, a query workload, $Q = \{q_1, q_2, . . . , q_n\}$, and a set of resource allocation and sensitive data disclosure constraints, C. The system initially performs the task of statistics collection over R and Q using the statistics gathering module. This module estimates the minimum set of required data items and the I/O sizes (alternatively running time) of base relations required to answer each query in Q. Additionally, the statistics SR are created as equi-width histograms and sent to the estimator modules. The computation partitioning module receives R, Q, C as well as the estimated I/O sizes and the minimum required set of data items for each query in Q, and then systematically solves the computation partitioning problem, CPP. In solving CPP, the monetary cost estimator is used by our algorithm to estimate the monetary costs of processing public cloud queries as well as storing intermediate public side data partitions, whereas the disclosure risk estimator is used to compute the amount of sensitivity that a solution candidate includes. On solving CPP, this layer produces two outputs: R_{pub} (the public cloud portion of R;) and furthermore, Q_{pub} the set of queries that will be executed over the public cloud. The private cloud stores the entire dataset R, whereas the public cloud only maintains the public-side data partition, R_{pub}. The non-sensitive and sensitive data in R_{pub} and R are stored using an appropriate representation technique on the public and private clouds respectively. Once the system has stored the data based on the solution to CPP, the system is now ready to support query processing.

2.2. Modification of Generic Framework for Hybrid Cloud Setting

Let sens(R’) be the estimated number of sensitive cells in dataset R’, baseTables(q) be the estimated minimum set of data items necessary to answer query q Є Q, runTx(q) be the estimated running time of query q Є Q at site x (either public or private), ORunT(Q’,Q’’) be the Overall execution time of queries in Q’, given that queries in Q’’ are executed on the public cloud, freq(q) be the frequency of running query q, MC be the defined monetary constraint, and DC be the defined sensitive data disclosure upper bound measured as number of sensitive items outsourced to the cloud, stor(R_{pub}) be the storage monetary cost of the public cloud partition, proc(q) be the processing monetary cost of a public side query q, than we can rewrite our generic formulization as follows: [3].

$$\begin{aligned}
&\text{minimize} && ORunT(Q, Q_{pub}) \\
&\text{subject to} && (1)\ store(R_{pub}) + \sum_{q \in Q_{pub}} freq(q)\ x\ proc(q) \leq MC \\
& && (2)\ sens(R_{pub}) \leq DC \\
& && (3)\ \forall q \in Q_{pub}\ baseTables(q) \subseteq R_{pub}
\end{aligned}$$

[3] Our framework could use any other sensitive data disclosure risk measure as well.

We showed that in our work, the above optimization problem could be solved using dynamic programming to find optimal workload partitioning that balances risk, computation monetary cost and run time.

2.3.Overview of the Experimental Results

Using existing TPC-H benchmark, and realistic cloud settings inspired by Amazon prices, we run experiments where public cloud is at least 3 times more powerful than the private cloud. In our experiments, the resource allocation cost was varied between 25- 50% of the total maximum value that was defined by the user. We defined four different overall sensitivity levels as, No-Sensitivity (the entire dataset is non-sensitive), 1%- Sensitivity, 5%-Sensitivity and 10%-Sensitivity (1%, 5% and 10% of the tuples of the lineitem table used in TPC-H benchmark are made sensitive). We defined seven different sensitive data exposure levels as 0% (none of the sensitive data is exposed), 10%, 25%, 40%, 50%, 75% and 100% (all of the defined sensitive data may be exposed). We then computed the overall performance of the query workload for different combinations of these three parameters, the results of which are presented in Figure 2. One of the first observations that can be made from Figure 2 is that when a user is willing to take additional risks by storing more sensitive data on the public side, they can gain a considerable speed-up in overall execution time.

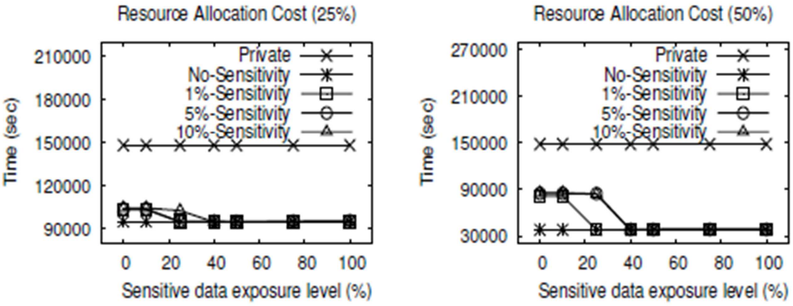

Figure 2: Overview of Results

Figure 2 also shows that when a user invests more capital towards resource allocation, a considerable gain in overall workload performance (even greater than 50%) can be achieved. This is expected since when more resources are allocated on the public side, we are better able to exploit the parallelism that is afforded by a hybrid cloud. Thus, the intuition that a hybrid cloud improves performance due to greater use of inherent parallelism is justified. Finally, from Figure 2, we also notice that we can achieve a considerable improvement in query performance (≈ 50%) for a relatively low risk (≈ 40%) and resource allocation cost (≈ 50%).

3. Managing Sensitive Encryption Key Exposure Risks in Public Clouds [7]

Despite its numerous advantages, cloud computing also introduces new challenges and concerns, primarily security and privacy risks. The concerns simply stem from outsourcing critical data (e.g., health records, social security numbers, or even cryptographic keys) and/or computing capabilities to a distant computing environment, where the resources are shared with other potentially untrusted customers.

)

In particular, to increase efficiency and to reduce costs, a CSP may place multiple virtual machines (VMs), belonging to different customers, to the same physical machine. In such an execution platform, VMs should be logically isolated from each other to protect the privacy of each client. The CSPs use virtual machine monitors (VMM) to realize logical isolation among VMs running on the same physical machine. However, the works that specially target public cloud infrastructures have shown that a clever adversary can perform cross- VM side-channel attacks (for brevity, cross-VM attack) to learn private information that resides in another VM, even under carefully enforced logical isolation. Initially, Ristenpart et al.[4] showed heuristics to improve an adversary's capabilities to place its VMs alongside the victim VMs, and learn crude information (e.g., aggregate cache usage). Even worse, Zhang et al.[5] managed to extract ElGamal decryption keys by cross-VM attacks.

These works have demonstrated that logical isolation and trustworthy cloud provider are not necessarily enough to guarantee the security of sensitive information. It would be too optimistic to assume that an adversary is only limited to the two aforementioned attacks. Unfortunately, there exists a wide variety of side-channel attacks, each with its own setup and methodology. Simply, the absence of such attacks on public cloud infrastructures does not necessarily mean that they are inapplicable.

To this end, we developed HERMES, a system that remedies the cryptographic key disclosure vulnerabilities of VMs in the public cloud by using well-established cryptographic tools such as Secret Sharing and Threshold Cryptography. Specifically, the key technique in our system is to partition a cryptographic key into several pieces, which are computed using threshold cryptosystems, and to store each share on a different VM. This makes it harder for an adversary to capture the complete cryptographic key itself, since it now has to extract shares from multiple VMs (note that there is no single key or a centralized key anymore in HERMES). To further improve the resilience, the same cryptographic key is re-shared periodically, so that a share is meaningful in only one time period/epoch. Consequently, we introduce two significant challenges against a successful attack: (i) Multiple VMs should be attacked, and (ii) each attack should succeed within a certain time period.

Using our generic model, we formalize the problem of finding good HERMES configurations (e.g., how many shares of each key, and how many shares are needed to reconstruct the secret), which minimizes the security risk for given monetary and performance constraints.

3.1. Risk-Aware Parameter Setting Mechanism for Protecting Sensitive Keys In The Clouds

In our formalization, we consider three main aspects: security, cost, and performance. Security aspect allows us to provide an upper bound on the possibility of a successful key extraction attack on HERMES for the given k (shares needed for correct decryption using the protected private key), l (total number of shares of the private key), and t (time to recreate and reshare the

[4] RISTENPART, T., TROMER, E., SHACHAM, H., AND SAVAGE, S. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In Proceedings of the 16th ACM conference on Computer and communications security (2009), ACM, pp. 199–212.

[5] ZHANG, Y., JUELS, A., REITER, M. K., AND RISTENPART, T. Cross-vm side channels and their use to extract private keys. In Proceedings of the 2012 ACM conference on Computer and communications security (2012), ACM, pp. 305–316.

secret key) values. Theoretically, increasing k and l, or decreasing t will make it harder for the adversary to achieve its goal by recovering keys. However, increasing l implies more defender VMs running on the cloud, which increases the total cost. Moreover, our experiments showed that the performance degrades as l and k increase together. Hence, the optimal values should be assigned to k, l, t for the given constraints (e.g., budget, performance limit).

Measuring Security: To quantify the probability of a successful attack in an epoch, we assume that the adversary has to start from scratch in each epoch, which implies that it loses all its previously acquired information. This is a valid assumption, since shares for each epoch are independent from one another, and a captured share does not contribute any information to the next epoch. The inability of conducting acquired information to the following epochs makes it convincing to model the probability of a successful attack as an exponentially distributed random variable. Given the success rate parameter θ, the probability distribution for the attack is:

$$f(t) = \begin{cases} \frac{1}{\theta} e^{-t/\theta} & \text{if } t > 0 \\ 0 & \text{otherwise} \end{cases}$$

Since the exponential distribution is memoryless and the cryptographic key is re-shared in each epoch, we can simply assume that the input to f is the time difference from the last re-sharing moment. Then, given the length of the epoch τ, the probability of a successful attack is:

$$F(\tau,\theta) = \int_0^\tau f(t).dt = 1 - e^{-\tau/\theta}$$

Finally, assuming that the probability of capturing shares from a single VM is identical to and independent from all other VMs, the probability of capturing at least k shares from l defender VMs in an epoch is (which we use as a way to measure the security of the system):

$$Sec(l,k,\tau,\theta) = \sum_{i=k}^{l} \binom{l}{i} (1 - e^{-\tau/\theta})^i (e^{-\tau/\theta})^{l-i}$$

Measuring Cost: Modeling monetary cost in HERMES is rather simple compared to the other two aspects. Assuming that the cloud provider does not charge money for the inter-VM communications, the total monetary cost is Cost(l) = l.β, where β is the unit cost of running a single VM on the cloud provider. The cost of communication with the client is also neglected, since this is not an additional cost incurred by HERMES.

Measuring Performance: The method to formalize the expected performance depends heavily on the application that HERMES is running for, and the metrics that the defender considers. For instance, one may value throughput more than the latency while running HERMES. On the other hand, the effects of changing parameters (i.e., k, l) in the mail server case study are far different than changing the same parameters in the micro benchmarking experiments. For brevity, we show the performance of HERMES for the given k and l as Perf (l,k), and leave it to the defender to define the characteristics of the function.

Optimization Problem: Given the success rate parameter θ, the unit cost of a VM β, the budget limit L_{cost}, and the performance limit L_{perf}, the aim of the optimization problem is to minimize the probability of a successful attack in an epoch while keeping the total monetary cost below L_{cost} and the performance below L_{perf} . Formally, the optimization problem is:

$$\begin{aligned} \text{minimize:} \quad & Sec(l,k,\tau,\theta) \\ \text{subject to:} \quad & Cost(l) \le L_{cost}, Perf(l,k) \le L_{perf} \\ & l \ge k > 1, \tau > 0 \end{aligned}$$

3.2. Application of Secure Multi-objective Optimization Framework for Micro Benchmarking

Modeling performance is highly dependent on the case study and the aimed configuration, thus it is challenging to apply the optimization to every single case. Instead, we targeted to optimize HERMES for 100 concurrent clients in the micro benchmarking scenario, since all experiment results for the chosen configuration are given in our work [7]. For brevity, we make a further assignment of parameters by choosing re-sharing period as τ= 5 sec and success rate parameter as θ= 3600. τ = 5 sec is the smallest value that we have tested, and is a valid value that allows HERMES to complete several computations in each epoch. Furthermore, choosing small re-sharing period will tighten the overall security, since the adversary has to complete the attack in a very short period. On the other hand, choosing θ as 3600 is due to the existing cross-VM attacks, which necessitates hours to capture the cryptographic key. In an exponential distribution, expected waiting time to observe one success is θ. Since, we expect the attack to succeed in an hour, we assign θ = 3600, representing the number of seconds in an hour. In addition, we check θ = 600 secs to observe changes in optimal values. probabilities in an epoch for fixed expected latency limit

In this example, we picked latency as the target performance metric to consider, assuming that the defender aimed to serve 100 concurrent clients as fast as possible. The important step to model performance is to figure out Perf (l,k). To overcome this, we applied multiple linear regression on our experiment results, and came up with a formula that gives the expected latency value for the given l and k values. As it is challenging to test every possible formula, and increasing the number of variables may over-fit the training data, we chose a simple polynomial Perf (l,k) = c_0 +c_1.l +c_2.k+c_3.(l/k) to model the expected latency, where the coefficients are c_0 =118, c_1 =18, c_2 =31, and c_3 =7 learned from existing performance data. Finally, to observe the effects of different performance limits L_{perf}, we calculated optimal HERMES setups for L_{perf} $\in [50,200]$. Finally, assuming that the defender will use the cheapest VM instance on Amazon EC2, she will pay \$0.02 per hour, which is approximately \$175 per year. We vary the monetary budget between \$350 per year and \$2800 per year to check optimal values.

	$\theta = 600$		$\theta = 3600$	
L_{cost}/yr	Conf.	Sec()	Conf.	Sec()
\$1820	$(2,2)$	$6.8 \cdot 10^{-5}$	$(2,2)$	$1.9 \cdot 10^{-6}$
\$3640	$(4,3)$	$2.2 \cdot 10^{-6}$	$(4,3)$	$3.7 \cdot 10^{-8}$
\$7280	$(8,5)$	$2.1 \cdot 10^{-9}$	$(8,5)$	$2.8 \cdot 10^{-13}$
\$14560	$(16,10)$	$1.1 \cdot 10^{-17}$	$(16,10)$	$2.1 \cdot 10^{-25}$

Table 1: Attack success probabilities for different system parameters

Table 1 shows the results of the optimization procedure for varying monetary budget, and fixed L_{perf} =150. The results include the optimal HERMES setup and the probability of a successful attack in one epoch, for both θ = 3600 and 600. We observe that as we increase the monetary budget, HERMES is allowed to run with more VMs, resulting in lower probabilities of success for the adversary. For instance, when the budget is \$7280 per year and θ =3600, HERMES can be configured to run in (8,5) setup (i.e., divide the secret key into 8 shares where any 5 share can jointly decrypt a message), while the adversary has only $2.8x10^{-13}$ chance to capture the partitioned cryptographic key. Please see [7] for more experimental results.

In summary, we present HERMES, a novel system to protect cryptographic keys in cloud VMs. The key idea is to periodically partition a cryptographic key using additive or Shamir secret sharing. With two different case studies, we show that the overhead can be as low as 1%. With such small overhead in an average request, cryptographic keys become more leakage-resilient against any adversary. Furthermore, we model the problem of finding optimal parameters for the given monetary and performance constraints, which minimizes the security risk. Using our formal model, the defender can calculate the probability of a successful attack, and take precautions (e.g., increase the number of VMs, decrease epoch length).

Publications Accepted/In-print Directly Funded By This Project

1. SingRu Celine Hoe, Murat Kantarcioglu, Alain Bensoussan: Studying dynamic equilibrium of cloud computing adoption with application of Mean Field Games. Allerton Conference 2012:220-224
2. Robert Nix, Murat Kantarcioglu: Contractual Agreement Design for Enforcing Honesty in Cloud Outsourcing. GameSec 2012:296-308
3. Kerim Yasin Oktay, Vaibhav Khadilkar, Bijit Hore, Murat Kantarcioglu, Sharad Mehrotra, Bhavani M. Thuraisingham: Risk-Aware Workload Distribution in Hybrid Clouds. IEEE CLOUD 2012:229-236
4. Vaibhav Khadilkar, Kerim Yasin Oktay, Murat Kantarcioglu, Sharad Mehrotra: Secure Data Processing over Hybrid Clouds. IEEE Data Eng. Bull. (DEBU) 35(4):46-54 (2012)
5. Erman Pattuk, Murat Kantarcioglu, Vaibhav Khadilkar, Huseyin Ulusoy, Sharad Mehrotra: BigSecret: A Secure Data Management Framework for Key-Value Stores. IEEE CLOUD 2013:147-154
6. Kerim Yasin Oktay, Vaibhav Khadilkar, Murat Kantarcioglu, Sharad Mehrotra: Risk Aware Approach to Data Confidentiality in Cloud Computing. ICISS 2013:27-42
7. Erman Pattuk, Murat Kantarcioglu, Zhiqiang Lin, Huseyin Ulusoy: Preventing Cryptographic Key Leakage in Cloud Virtual Machines. USENIX Security 2014:703-718
8. Mehmet Kuzu, Mohammad Saiful Islam, Murat Kantarcioglu: Efficient privacy-aware search over encrypted databases. CODASPY 2014:249-256
9. Mohammad Saiful Islam, Mehmet Kuzu, Murat Kantarcioglu: Inference attack against encrypted range queries on outsourced databases. CODASPY 2014:235-246
10. Huseyin Ulusoy, Murat Kantarcioglu, Erman Pattuk, Kevin W. Hamlen: Vigiles: Fine-Grained Access Control for MapReduce Systems. BigData Congress 2014:40-47
11. Kerim Yasin Oktay, Sharad Mehrotra, Vaibhav Khadilkar, Murat Kantarcioglu: SEMROD: Secure and Efficient MapReduce Over HybriD Clouds. SIGMOD 2015:153-166
12. Mehmet Kuzu, Mohammad Saiful Islam, Murat Kantarcioglu: Distributed Search over Encrypted Big Data. CODASPY 2015:271-278
13. Mohammad Saiful Islam, Mehmet Kuzu, Murat Kantarcioglu: A Dynamic Approach to Detect Anomalous Queries on Relational Databases. CODASPY 2015:245-252
14. Erman Pattuk, Murat Kantarcioglu, Huseyin Ulusoy: BigGate: Access Control Framework for Outsourced Key-Value Stores. CODASPY 2015:171-182

15. Huseyin Ulusoy, Pietro Colombo, Elena Ferrari, Murat Kantarcioglu, Erman Pattuk: GuardMR: Fine-grained Security Policy Enforcement for MapReduce Systems. ASIACCS 2015:285-296

Honors/Awards

Alain Bensoussan, SIAM – W.T. & Idalia Reid Prize, 2014

Alain Bensoussan, Fellow American Mathematical Society, 2013

Murat Kantarcioglu, IEEE Senior Member, 2013

Murat Kantarcioglu, ACM Senior Member, 2013

Murat Kantarcioglu, Homer Warner Award (Best Paper), American Medical Informatics Association (AMIA) Annual Symposium, 2014

Bhavani Thuraisingham, the SDPS 2012 Transformative Achievement Gold Medal for interdisciplinary research on integrating computer sciences with social sciences

Bhavani Thuraisingham, IBM Faculty Award in Cyber Security, 2013

Bhavani Thuraisinghan, named one of the 15 Top Cyber Security Professors by Forensics Colleges, December 2013.

Bhavani Thuraisingham, Recipient of the 2014 Inaugural Award for Outstanding Research Leadership in Information Management, Integration, Reuse and Security presented at the IEEE IRI (Information Reuse and Integration) Conference in San Francisco, CA in August 2014 by the Society for Information Reuse and Integration.

1.

1. Report Type

Final Report

Primary Contact E-mail
Contact email if there is a problem with the report.

muratk@utdallas.edu

Primary Contact Phone Number
Contact phone number if there is a problem with the report

972-883-6616

Organization / Institution name

University of Texas at Dallas

Grant/Contract Title
The full title of the funded effort.

Ecologically Inspired Framework for Assured Data Cloud

Grant/Contract Number
AFOSR assigned control number. It must begin with "FA9550" or "F49620" or "FA2386".

FA9550-12-1-0082

Principal Investigator Name
The full name of the principal investigator on the grant or contract.

Murat Kantarcioglu

Program Manager
The AFOSR Program Manager currently assigned to the award

TRISTAN N NGUYEN

Reporting Period Start Date

04/30/2012

Reporting Period End Date

09/30/2015

Abstract

As a part of this project, we explored risk based approaches for securely managing data in the cloud. Basically, inspired by how living organisms manage risks in nature while preserving and conserving energy, we developed novel risk based approaches that balances risk (i.e., potential sensitive data disclosure risk), computational cost (i.e., how long will it take to run the security enhanced tasks in the cloud?), monetary cost (i.e., how much more you would pay to the cloud service provider due to security enhanced cloud computing?). In different settings, we solve the variants of the multi-objective optimization framework where we find best query execution plan Q among all possible query plans that minimizes the total run time while it does not exceed the predefined monetary costs and risk measures.

To our knowledge, as also reported in Network World Magazine , this is the first framework that integrates rigorous risk management tools "…that meets the conflicting goals of performance, sensitive data disclosure risk and resource allocation costs getting weighed and balanced.". The framework proposed as a part of this proposal resulted in numerous publications in top security, data management and cloud computing venues and already received hundreds of citations according to Google Scholar. We summarize the general applicability of the above framework by briefly discussion how it is applied in two very different application settings. In addition to following examples and contributions, we developed the first encrypted

key-value store that supports efficient search and access control capabilities for hybrid clouds.

Distribution Statement

This is block 12 on the SF298 form.

Distribution A - Approved for Public Release

Explanation for Distribution Statement

If this is not approved for public release, please provide a short explanation. E.g., contains proprietary information.

SF298 Form

Please attach your SF298 form. A blank SF298 can be found here. Please do not password protect or secure the PDF The maximum file size for an SF298 is 50MB.

afosr-form-kantarcioglu.pdf

Upload the Report Document. File must be a PDF. Please do not password protect or secure the PDF . The maximum file size for the Report Document is 50MB.

afosr-final-report-kantarcioglu.pdf

Upload a Report Document, if any. The maximum file size for the Report Document is 50MB.

Archival Publications (published) during reporting period:

1. SingRu Celine Hoe, Murat Kantarcioglu, Alain Bensoussan: Studying dynamic equilibrium of cloud computing adoption with application of Mean Field Games. Allerton Conference 2012:220-224

2. Robert Nix, Murat Kantarcioglu: Contractual Agreement Design for Enforcing Honesty in Cloud Outsourcing. GameSec 2012:296-308

3. Kerim Yasin Oktay, Vaibhav Khadilkar, Bijit Hore, Murat Kantarcioglu, Sharad Mehrotra, Bhavani M. Thuraisingham: Risk-Aware Workload Distribution in Hybrid Clouds. IEEE CLOUD 2012:229-236

4. Vaibhav Khadilkar, Kerim Yasin Oktay, Murat Kantarcioglu, Sharad Mehrotra: Secure Data Processing over Hybrid Clouds. IEEE Data Eng. Bull. (DEBU) 35(4):46-54 (2012)

5. Erman Pattuk, Murat Kantarcioglu, Vaibhav Khadilkar, Huseyin Ulusoy, Sharad Mehrotra: BigSecret: A Secure Data Management Framework for Key-Value Stores. IEEE CLOUD 2013:147-154

6. Kerim Yasin Oktay, Vaibhav Khadilkar, Murat Kantarcioglu, Sharad Mehrotra: Risk Aware Approach to Data Confidentiality in Cloud Computing. ICISS 2013:27-42

7. Erman Pattuk, Murat Kantarcioglu, Zhiqiang Lin, Huseyin Ulusoy: Preventing Cryptographic Key Leakage in Cloud Virtual Machines. USENIX Security 2014:703-718

8. Mehmet Kuzu, Mohammad Saiful Islam, Murat Kantarcioglu: Efficient privacy-aware search over encrypted databases. CODASPY 2014:249-256

9. Mohammad Saiful Islam, Mehmet Kuzu, Murat Kantarcioglu: Inference attack against encrypted range queries on outsourced databases. CODASPY 2014:235-246

10. Huseyin Ulusoy, Murat Kantarcioglu, Erman Pattuk, Kevin W. Hamlen: Vigiles: Fine-Grained Access Control for MapReduce Systems. BigData Congress 2014:40-47

11. Kerim Yasin Oktay, Sharad Mehrotra, Vaibhav Khadilkar, Murat Kantarcioglu: SEMROD: Secure and Efficient MapReduce Over HybriD Clouds. SIGMOD 2015:153-166

12. Mehmet Kuzu, Mohammad Saiful Islam, Murat Kantarcioglu: Distributed Search over Encrypted Big Data. CODASPY 2015:271-278

13. Mohammad Saiful Islam, Mehmet Kuzu, Murat Kantarcioglu: A Dynamic Approach to Detect Anomalous Queries on Relational Databases. CODASPY 2015:245-252

14. Erman Pattuk, Murat Kantarcioglu, Huseyin Ulusoy: BigGate: Access Control Framework for Outsourced Key-Value Stores. CODASPY 2015:171-182

15. Huseyin Ulusoy, Pietro Colombo, Elena Ferrari, Murat Kantarcioglu, Erman Pattuk: GuardMR: Fine-grained Security Policy Enforcement for MapReduce Systems. ASIACCS 2015:285-296

Changes in research objectives (if any):

None.

Change in AFOSR Program Manager, if any:

Dr. NGUYEN assigned as program manager during the last year of the project.

Extensions granted or milestones slipped, if any:

No cost extension granted for 6 months.

AFOSR LRIR Number

LRIR Title

Reporting Period

Laboratory Task Manager

Program Officer

Research Objectives

Technical Summary

Funding Summary by Cost Category (by FY, $K)

	Starting FY	FY+1	FY+2
Salary			
Equipment/Facilities			
Supplies			
Total			

Report Document

Report Document - Text Analysis

Report Document - Text Analysis

Appendix Documents

2. Thank You

E-mail user

Feb 22, 2016 22:10:55 Success: Email Sent to: muratk@utdallas.edu