IT@Intel White Paper

Intel IT
IT Best Practices
Cloud Computing and Information Security
June 2012

Enhancing Cloud Security Using Data Anonymization

Although more research is necessary before it is ready for production use, data anonymization can ease some security concerns, allowing for simpler demilitarized zone and security provisioning and enabling more secure cloud computing.

Executive Overview

Intel IT is exploring data anonymization—the process of obscuring published data to prevent the identification of key information—in support of our vision of a hybrid cloud computing model and our need to protect the privacy of our employees and customers. We believe data anonymization is a viable technique for enhancing the security of cloud computing.

Although we realize that a 100-percent secure cloud infrastructure is impossible. we are exploring the possibility of anonymizing data to augment our cloud security infrastructure. Data anonymization makes data worthless to others, while still allowing Intel IT to process it in a useful way.

We conducted a proof of concept (PoC), in which we used data anonymization to protect event logging data stored in a public cloud. The PoC was successful in demonstrating that data anonymization can work and that obscured data is still useful for analysis. We were able to perform both performance analysis and security analysis on the anonymized data.

- In our performance analysis, we discovered performance issues; for example, one web site performed two redirects before the user accessed the actual content, thereby increasing the access time.
- In our security analysis, we were able to detect real security events, such as probes occurring on the web server.

Although more research is necessary before it is ready for production use, data anonymization can ease some security concerns, allowing for simpler demilitarized zone and security provisioning and enabling more secure cloud computing.

We plan to explore data anonymization further, including conducting a more extensive PoC, developing further use cases for data anonymization, educating potential enterprise cloud users about the potential benefits and pitfalls of data anonymization, and documenting existing open source data anonymization applications.

Jeff Sedayao
Enterprise Architect, Intel IT

Contents

IT@INTEL

The IT@Intel program connects IT professionals around the world with their peers inside our organization – sharing lessons learned, methods and strategies. Our goal is simple: Share Intel IT best practices that create business value and make IT a competitive advantage. Visit us today at www.intel.com/IT or contact your local Intel representative if you'd like to learn more.

BACKGROUND

Cloud computing can help reduce costs, increase business agility, and enable IT to focus on projects with a high return on investment. Intel is already experiencing the benefits of implementing an enterprise private cloud, and Intel IT is making progress toward the vision of a hybrid cloud usage model. As we pursue our vision of using hybrid clouds, we also realize the need to protect data according to Intel information security policies.

Security and privacy concerns are a significant obstacle that is preventing the extensive adoption of the public cloud, not only in Intel IT, but also across the industry. IT organizations are typically reluctant to place sensitive and valuable data in infrastructure that they do not control. This is especially true in Europe, which has strict laws concerning the use of and storage of personally identifiable information.

Furthermore, as the proliferation of devices making up the compute continuum continues, devices are increasingly implementing Global Positioning System capabilities, and client-aware, location-oriented social media applications are becoming more common. In these types of situations, location tracking becomes an issue—location information can be very useful for providing customized and localized services, but the storage and mining of location data is associated with privacy and regulatory issues. Multi-tenancy, where multiple tenants share cloud infrastructure poses an additional concern about the deliberate or accidental exposure of data.

We are exploring ways to make data safer for the cloud by preparing for potential security breaches. One method is to store cloud data in a way that makes it worthless to anyone except the owner of the data.

All forms of data protection involve a trade-off between security and ease of implementation, with more secure systems requiring the greatest effort to implement.

For example, homomorphic encryption, which is a strong encryption method that supports computations without decrypting the input, is a possible solution. But while theoretically possible, homomorphic encryption is too computationally intensive for practical use.

In general, making progress in privacy issues comes at the expense of having to do more work in processing the published data. We needed to find a practical approach to cloud data storage that both protects privacy and is not too difficult to implement.

DATA ANONYMIZATION CONCEPTS AND TECHNIQUES

Anonymization is a technique that enterprises can use to increase the security of data in the public cloud while still allowing the data to be analyzed and used. Data anonymization is the process of changing data that will be used or published in a way that prevents the identification of key information.

Using data anonymization, key pieces of confidential data are obscured in a way that maintains data privacy. The data can still be processed to gain useful information. Anonymized data can be stored in a cloud and processed without concern that other individuals may capture the data. Later, the results can be collected and mapped to the original data in a secure area.

Achieving Privacy Using Anonymization

Figure 1 shows a simple theoretical example of using data anonymization to protect sensitive data. In the example, the goal is to calculate total revenue for several companies, without exposing the names of those companies. To accomplish this goal, company names are changed. For example, the name of Company A in the original data is changed to "Bob" in the cloud-based data. Also, fictitious records are added to the cloud-based data to further anonymize the data. A translation table stored in a secure enclave, which is typically a secured area on the enterprise network, maps the translation of company names and identifies which data is fictitious.

Using the anonymized data, total revenue can be calculated in the cloud without exposing sensitive information. The result can then be corrected internally by subtracting fictitious company amounts, using the translation table.

This type of anonymization can block some forms of data mining attacks, because by adding fictitious data, it is impossible to determine the number of companies, or the companies with the most revenue or the least revenue.

Anonymization can have severe consequences if done incorrectly. For example, a popular online movie supplier released a database of users and movie selections as part of a contest. To protect the identity of customers, the supplier replaced the customer names with random numbers and removed personal details. Security researchers showed that by correlating the data with publically available Internet Movie Database information, they could reveal the identities of many of the individual users.[1] A key learning from this situation is that data anonymization involves more than just deleting fields in a database.

Several formal models of security can help improve data anonymization, including k-anonymity and l-diversity. The next two sections provide further explanation and examples of these security models.

k-ANONYMITY

k-Anonymity is a formal model of privacy created by L. Sweeney.[2] The goal is to make each record indistinguishable from a defined number (k) of other records if attempts are made to identify the data.

A set of data is k-anonymized if, for any data record with a given set of attributes, there are at least k-1 other records that match those attributes. For example, consider a data set that contains two attributes: gender and birthday. The data set is k-anonymized if, for any record, k-1 other records have the same gender and birthday. In general, the higher the value of k, the more privacy is achieved.

k-Anonymity assigns properties to data attributes and requires that they be handled in specific ways, as shown in Table 1.

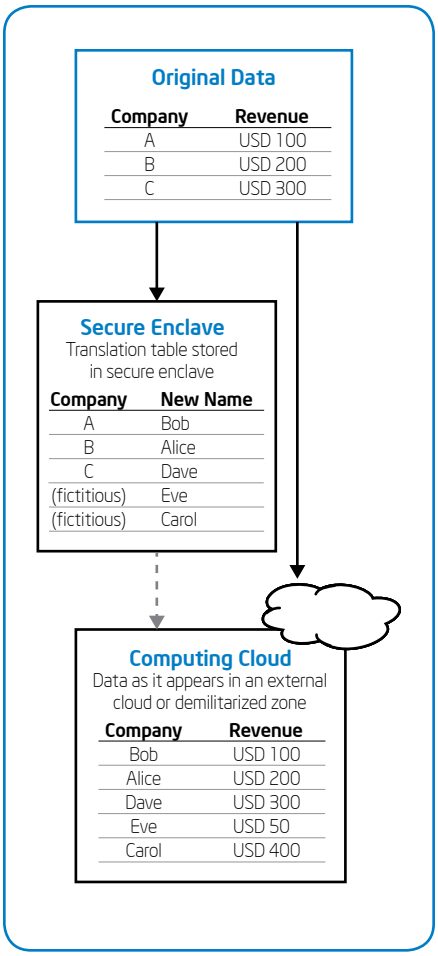

Figure 1. Data anonymization helps enable safer computing in the cloud.

[1] Schneier, Bruce, "Why 'Anonymous' Data Sometimes Isn't." *Wired* magazine (2007). www.wired.com/politics/security/commentary/securitymatters/2007/12/securitymatters_1213

[2] Sweeney, L., "k-anonymity: A Model for Protecting Privacy." *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems* 10 (2002): 557–570. www.epic.org/privacy/reidentification/Sweeney_Article.pdf

Table 1. k-Anonymity Attributes

Attribute Type	Property	Example	Action Required
Key	Can identify an individual directly	Name, social security number	Remove or obscure
Quasi-identifier	Can be linked with external information to identify an individual	Zip code, birthday, gender	Suppress or generalize
Sensitive	Data that an individual is sensitive about revealing	Income, type of illness	Needs to be de-linked from the individual

Table 2. Sample k-Anonymized Patient Records

Zip	Age	Disease
130▪	2▪	Heart disease
130▪	2▪	Heart disease
130▪	2▪	Heart disease
130▪	2▪	Viral infection
130▪	3▪	Cancer
130▪	3▪	Cancer

▪ denotes a suppressed value.

Table 3. Sample Patient Records Created with l-Diversity, Where l=2

Zip	Age	Disease
130▪	2▪	Heart disease
130▪	2▪	Heart disease
130▪	2▪	Heart disease
130▪	2▪	Cancer
130▪	2▪	Cancer
130▪	2▪	Viral infection
130▪	2▪	Viral infection
130▪	3▪	Viral infection
130▪	3▪	Viral infection
130▪	3▪	Cancer
130▪	3▪	Cancer

▪ denotes a suppressed value.

Studies have shown that 87 percent of people in the United States can be identified by the combination of zip code, birthday, and gender.[3]

Sensitive attributes, such as the type of illness or disease someone may have, should be de-linked from the individual. To guarantee k-anonymity, there should be k identical sequences of quasi-attributes in the data.

Table 2 contains sample records from a theoretical hospital: Zip and Age are quasi-identifiers and Disease is a sensitive attribute. Both the zip code and the patient age have been suppressed, with patient age listing only what decade of life the patients are in. The table has at least two copies of each quasi-identifier, making this table k-anonymous, where k=2.

k-Anonymity guarantees that an individual cannot be identified from a given number of people in a set of size k. If there are not k identical sequences of quasi-identifiers, fictitious records could be added to the data, with the understanding that the effects of those fictitious records on processing must be removed at some point.

L-DIVERSITY

k-Anonymity is vulnerable to a number of attacks. For example, using the data in Table 2, at least two attacks are theoretically possible.

- **Homogeneity attack.** If an attacker knows that Bob has an entry in the data and that Bob is in his 30s, the attacker will know that Bob has cancer.
- **Background knowledge attack.** If an attacker knows that 21-year-old Yuko has an entry in the data and that as a young woman she is unlikely to have heart disease, the attacker then knows Yuko has a viral infection.

Use of l-diversity, another privacy model, can counter both of these attacks.[4] l-Diversity improves anonymization beyond what k-anonymity provides. The difference between the two is that while k-anonymity requires each combination of quasi-identifiers to have k entries, l-diversity requires that there are l different sensitive values for each combination of quasi-identifiers. Table 3 adds l-diversity to the data from Table 2. This data has k-anonymity, where k=4, as well as l-diversity, where l=2. Homogeneity and background knowledge attacks described above are both impossible with the anonymized data in Table 3.

While l-diversity adds more privacy than k-anonymity alone, the natural occurrence of sensitive attributes may not provide enough variety to achieve l-diversity. Fictitious data can be inserted into the data to increase occurrences, but these must be compensated for when doing analysis. Also, probabilistic inferences are still possible. For example, from the data in Table 3, it is possible to determine that Yuko has either cancer or a viral infection, with a 50-percent chance of either one.

Anonymization Techniques

Both k-anonymity and l-diversity require that data be obscured. Several techniques can be used to obscure data, as shown in Table 4. In the table, sample data is modified to show how each anonymization technique works.

[3] Samarati, P., and L Sweeney, "Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression." Technical report SRI-CSL-98-04. SRI computer science laboratory. Palo Alto, CA (1998). www.epic.org/privacy/reidentification/Samarati_Sweeney_paper.pdf

[4] Machanavajjhala, Ashwin, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam, l-Diversity: Privacy Beyond k-Anonymity. *ACM Trans. Knowl. Discov. Data* 1, 1, Article 3 (2007). http://www.cs.cornell.edu/~vmuthu/research/ldiversity-TKDD.pdf

Table 4. Anonymization Techniques to Obscure Data

Technique	Description and Application	Example						
		IP Address	First Name	Last Name	Employee Number	Years of Service	Phone Number	Monthly Salary
		SAMPLE DATA						
		143.183.23.3	Bob	Smith	325211	13	408-555-2935	USD 5000
		143.183.23.10	Alice	Jones	452893	3	408-555-2931	USD 4000
Hiding	▪ A value is replaced with a constant value (typically 0); sometimes called a "black marker." ▪ Useful for suppressing sensitive attributes that may not be needed for processing. For example, to publish data for a phone directory, salary information is not needed.	143.183.23.3	Bob	Smith	325211	13	408-555-2935	0
		143.183.23.10	Alice	Jones	452893	3	408-555-2931	0
		The salary field is hidden.						
Hashing	▪ Maps each value to a new (not necessarily unique) value. ▪ Useful for mapping a large, variable amount of data into a number of a certain length.	143.183.23.3	6834523439811			13	408-555-2935	USD 5000
		143.183.23.10	2349510342932			3	408-555-2931	USD 4000
		The first name, last name, and employee number are hashed into a single fixed-length number.						
Permutation	▪ Maps each original value to a unique new value. ▪ Allows the translation of the new value back to the original value, given a translation table that is stored in a secure area.	143.183.23.3	Rob	Clemente	325211	13	408-555-2935	USD 5000
		143.183.23.10	Eva	Gonzales	452893	3	408-555-2931	USD 4000
		The first name and last names are mapped to new values.						
Shift	▪ Adds a fixed offset to the numerical values. ▪ Useful for concealing data while allowing computations in areas such as the cloud.	143.183.23.3	Bob	Smith	325211	13	408-555-2935	USD 15000
		143.183.23.10	Alice	Jones	452893	3	408-555-2931	USD 14000
		USD 10,000 is added to the salary values.						
Enumeration	▪ Maps each original value to a new value to preserve ordering. ▪ Allows analysis of data that requires ordering, such as ranking people by salary.	143.183.23.3	Bob	Smith	325211	13	408-555-2935	USD 25000
		143.183.23.10	Alice	Jones	452893	3	408-555-2931	USD 20000
		Salary data is changed, but the relative order of the salary is preserved.						
Truncation	▪ A field is shortened, losing data at the end. ▪ Useful for hiding data while still preserving information for the data.	143.183.23.3	Bob	Smith	325211	13	408	USD 5000
		143.183.23.10	Alice	Jones	452893	3	408	USD 4000
		The phone number is truncated after the area code. This hides the phone number but preserves the information about where the employee is located.						
Prefix-preserving	▪ Retains n-bit prefix on IP addresses.	143.183.79.169	Bob	Smith	325211	13	408-555-2935	USD 5000
		143.183.3.25	Alice	Jones	452893	3	408-555-2931	USD 4000
		The IP address is scrambled, but the 16-bit prefix is preserved.						

The Difference between Data Anonymization and Encryption

Although data anonymization and encryption are related topics and are both useful techniques for securing cloud-based data from privacy and security breaches, they are not the same thing.

- Data anonymization is the process of transforming data so that it can be processed in a useful way, while preventing that data from being linked to individual identities of people, objects, or organizations.
- Encryption involves transforming data to render it unreadable to those who don't have the key to decrypt it.

Encryption can be a useful tool for doing anonymization, particularly when hiding identifying information in a set of data. However, encryption, while useful, is neither necessary nor sufficient for doing anonymization. Data can be successfully anonymized without encryption, and encrypted data is not necessarily anonymized.

EXAMPLE

Table 5 shows an example of a file where the goal is to calculate the average salary by location and years of service, while obscuring the identity of the individual.

This data is obscured using the following techniques:

- Phone numbers are truncated, leaving only the area code.
- Names and employee numbers are hidden.
- Salary and years of service are offset, using shift.
- The prefix of the IP addresses is preserved.

This approach protects the privacy of individuals' salaries, so the calculation of salary averages could be done in a cloud without having to worry about data and results being exposed. The results could be adjusted in a secure enclave using knowledge of the shift values associated with salary and years of service. As long as the shift values remain confidential, the true average salary is protected. Phone numbers have been truncated to prevent using those as quasi-identifiers to reveal the identity of the individuals associated with data entries.

PROOF OF CONCEPT

We conducted a proof of concept (PoC) showing that data anonymization is a viable technique to use for secure cloud computing. The PoC used data anonymization to protect and process event log data stored in a public cloud.

Collaboration with external companies and the need for greater awareness of what is occurring in our computing environment is leading us to do more and more data logging. In the future, much of this log data will be generated outside of Intel's network perimeters or in network demilitarized zones (DMZs). We need the ability to store and later analyze these logs to make them useful. We wanted to determine if we could use a software-as-a-service (SaaS) log management supplier in the public cloud but still maintain data security. Using a SaaS log management supplier, instead of simply storing log data on enterprise servers, provides a fast, searchable, and easy-to-use log archive that can reduce the number of log entries needed for other processing and that is in a Hadoop*-ready format for Big Data mining.[5]

PoC Implementation

The PoC, which we conducted over several weeks, had the following overall structure:

- Evaluate anonymization tools for storing anonymized data.
- Store anonymized logs in a public cloud-based SaaS log management application.
- Analyze logs using Hadoop on the stored logs to gather security and performance data.
- Document results and report them.

We used an existing external application to generate security and performance data that we could send to the SaaS log management supplier for analysis. The application that we chose runs on an academic cloud and monitors the performance of a number of web sites. Performance data, such as the time it takes to look up the web server's domain name, the time it takes to set up a TCP connection, and the page download rate, are made available on a web site that

[5] Hadoop* is a project that provides an open source implementation of frameworks for reliable, scalable, distributed computing, and data storage.

Table 5. An Example of Calculating the Average Salary While Obscuring Identity

IP Address	First Name	Last Name	Employee Number	Years of Service	Phone Number	Monthly Salary
143.183.79.169	x	y	0	11	408	USD 15000
143.183.3.25	x	y	0	1	408	USD 14000

runs on each virtual machine (VM). Figure 2 illustrates the relationship between the SaaS log management supplier, the VMs, and the secure enclave that stores the data anonymization translation table.

Anonymization takes place on the VMs sending the data, and data is de-anonymized within a secure enclave. For the PoC, 47 VMs sent anonymized log information to the SaaS log management supplier at the rate of 2,800 events (individual log entries) per hour.

We decided to anonymize IP addresses in the web server access logs and URLs in the web server access log files.

- We used the IP::Anonymous perl library to mask IP addresses.
- We used the MD5 hash function to anonymize URLs. Although most of the security industry views the MD5 hash function as too weak to be effective, we chose it as an easily implementable example. An actual implementation would use another hash function.
- The anonymization software that we used to conceal IP addresses uses Advanced Encryption Standard (AES) encryption to scramble IP addresses.

SECURITY ANALYSIS

We wanted to determine whether the data distribution web servers were being probed. We planned to probe some of the ports on the web servers and then see if we could detect that activity in the anonymized logs. One way to detect a security anomaly is to look for a spike in log activity. Because this method of detection can be subverted through a slow attack, we also conducted a probe in which we performed only one port probe on one of the data distribution web servers.

PERFORMANCE ANALYSIS

We also wanted to use the SaaS log management supplier to study the performance of the web sites we were analyzing. For example, we wanted to calculate summary statistics such as average TCP connection set up times and standard deviation of those times.

Results

We found that we could glean useful security and performance information from the anonymized data that we stored and analyzed in the cloud. Our simulated attacks could be detected in the anonymized logs, and we found performance issues in anonymized performance logs.

- During the security analysis testing, we didn't detect any active probing of the monitoring VMs. However, we searched older logs and found that there had been probes on the web server. This confirmed our theory that the approach we took in looking for security business intelligence events could detect real events.
- Although the SaaS log management supplier we used during the PoC didn't support number-crunching analytics, such as calculating averages, we were able to pinpoint other performance issues. For example, we discovered that one web site performed two redirects before the user accessed the actual content, thereby increasing the access time.

Key Learnings

Our PoC demonstrated that anonymization for cloud processing can work, although more research is necessary before it is ready for production use. The following points summarize our key learnings from our PoC:

- Anonymized logs can be used for both performance analysis and security analysis.
- Anonymization can ease some security concerns, allowing for simpler DMZ and security provisioning and making it possible to do a number of operations in public clouds.
- When using encryption to anonymize the data, it is important to manage the encryption keys that control the real-to-anonymized value mapping.

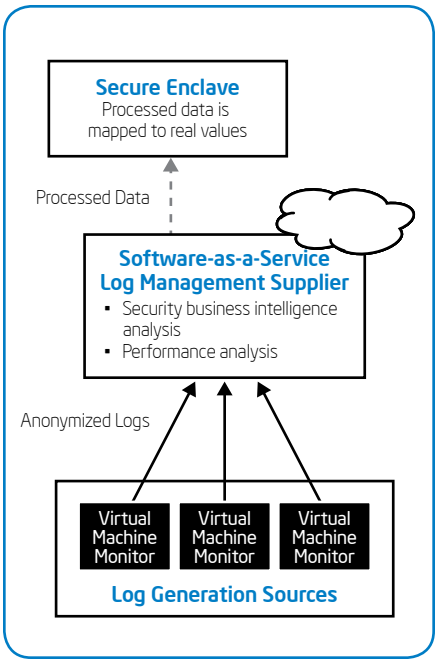

Figure 2. Our proof of concept analyzed anonymized log data in the cloud.

- Anonymization for the cloud could be a new and major use case for Intel® Advanced Encryption Standard – New Instructions (Intel® AES-NI), as this technology speeds up AES encryption used in some anonymization techniques, such as hashing.

NEXT STEPS

Intel's IT security architecture and cloud engineering groups deemed our PoC useful enough to warrant further work concerning external logging and data anonymization. We plan to refine anonymization use cases and conduct another PoC with Intel data. As we collaborate with more companies and pursue our hybrid cloud vision, our work with anonymization has the potential to be a useful technique for enabling enterprise usage of public clouds.

Products already exist today that take advantage of anonymization techniques. These products make it possible for companies to use publicly hosted SaaS offerings without revealing sensitive information. However, it is clear that more work needs to be done with anonymization.

- We need to educate potential enterprise cloud users about anonymization and its potential benefits and pitfalls. While data anonymization isn't foolproof, it is appropriate for certain use cases, and there are ways to determine if vulnerabilities exist in the anonymized data.
- Available open source tools seem capable but are not well documented, and some have been abandoned by their creators. We intend to document some of these open source tools.
- Our PoC used AES encryption to do anonymization. Intel AES-NI instructions could be used to speed up the anonymization process. Therefore, secure use of public clouds could be a potential use case for Intel AES-NI. We will adapt the open source anonymization tools to use Intel AES-NI and investigate if that adds value to the anonymization process.

CONCLUSION

Intel has achieved great savings and success with our enterprise private cloud, and we are moving toward a hybrid cloud usage model and employing techniques that protect the privacy of Intel's employees and customers. But for that model to work, we need to protect data according to Intel information security policies. We conducted a PoC that showed that data anonymization—the process of obscuring published data to prevent the identification of key information—is a viable technique for enhancing the security of cloud computing.

Collaboration with external companies, and a desire for greater awareness of what is happening in our computing environment, is leading Intel IT to do more and more data logging. In the future, much of this log data will be generated outside of Intel's network perimeters, or in DMZs. Instead of insisting that cloud infrastructures be totally secure, we can prepare for potential security breaches in the cloud by anonymizing the data, making it worthless to others while still allowing Intel IT to process it in a useful way.

Although data anonymization is not foolproof, it is one important tool in our continuing pursuit of secure cloud computing. We intend to further research data anonymization, including developing further use cases, educating potential enterprise cloud users about the potential benefits and pitfalls of data anonymization, documenting existing open source data anonymization applications, and conducting another, more extensive, PoC.

For more information on Intel IT best practices, visit www.intel.com/it.

ACRONYMS

Intel® AES-NI	Intel® Advanced Encryption Standard – New Instructions
DMZ	demilitarized zone
PoC	proof of concept
SaaS	software as a service
VM	virtual machine