Data Security: On Premise or in the Cloud

By Ulf Mattsson – ISSA member, New York Chapter

This article discusses data protection techniques suitable for deploying on-premise or in private or public clouds. When making the decision on which techniques, there are several things to consider including how the data is used and secured.

Abstract

With sensitive data residing everywhere and the breach epidemic growing, the need for advanced data security solutions has become even more critical. Compliance with regulations like GDPR, PCI DSS, CCPA, and HIPAA are driving the need for de-identification of sensitive data. This article discusses data protection techniques suitable for deploying on-premise or in private or public clouds. When making the decision on which techniques, there are several things to consider including how the data is used and secured.

When companies store data in a public cloud, they don't have direct control over their data. The principal security challenges of public cloud hosting are keeping data from prying administrators and keeping different hosted user communities separate. Storing and using data on premise poses similar concerns: keeping sensitive data out of the wrong hands, whether insiders or cyber thieves. Encryption protects data as it is stored, used, and transmitted on computer networks and is applicable at the database, file, or volume level, or at a granular cell level that applies to a specific column or field. Instead of building walls around servers or hard drives, a protective layer of encryption can protect specific sensitive data items or objects.

Encryption considerations

Encryption should be performed on premise if you want total control of your sensitive data and encryption keys. On a cloud platform, if you are sending cleartext data and still want to gain some control of your data, you can bring your own cloud encryption key to the public cloud platform. You may consider keeping control of your encryption keys and not expose cleartext data at the public cloud. A cloud encryption gateway could be an option. You may keep your keys, encryption, and tokenization separated from the public cloud. One option is to segment these security controls into a private cloud, hosted on premise or outsourced.

Two main types of data encryption exist: asymmetric, also known as public-key encryption, and symmetric. Symmetric-key ciphers are well suited to encrypt short fields and lower computing cost than asymmetric encryption. Symmetric-key ciphers can limit attacks from quantum computers for many years if you are using AES 256-bit keys. If you are using public-key encryption, you should consider moving away from RSA and ECC algorithms and at least significantly increase key lengths.

Encryption can adversely affect the system's performance and functionality. A benefit of tokenization vs. encryption is the business utility and agility that tokens offer. Customizable token schemes, such as length- and format-preserving tokens, can retain portions of the sensitive data to preserve much of its original operational value. A data warehouse may need to process 10 million encryption operations per second. This may not be an issue with PCI data tokens but a major issue with PII or PI data if using FPE encryption.

Data analysis software creates a cache, or a local copy, of frequently used data and may even be swapped out to permanent file storage on disk. This can present security problems from attackers searching for sensitive data. Tokenization or format-preserving encryption can provide data protection also in memory caches.

Figure 1 illustrates that businesses are moving to different cloud models over time [17] and will be using a combination of platforms for computation and data storage. Each platform will need different considerations for protecting sensitive

data and operational aspects. The global increase in security and privacy regulations will force many organizations to find the right balance for each data item and business use case. I think that GDPR and CCPA are setting the stage. The impact of using different cloud computing models needs careful data security considerations.

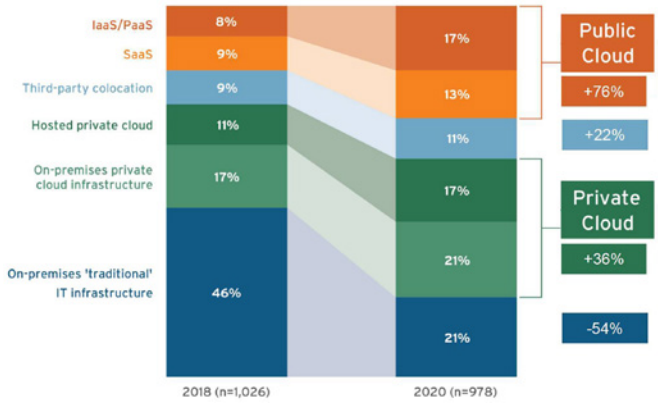

Figure 1 – Businesses are moving to different hybrid cloud models [17]

A company that illustrates common use cases

Election Systems Inc. ("the company") is our example of a growing company that is developing and selling election support software applications globally. In the core use case in the election process, the voters are provided with a tracking code that when voting is complete they will be able to enter into a website to confirm their vote was counted and not altered; the website will not display their actual votes. The company needed a way for their customers and partners to pay for products online over the Internet using a web browser or mobile device, but also needs to comply with several international regulations including GDPR and CCPA. It needs to reduce the risk of identifying individuals (PII and PI data) in operational systems and in development and test systems.

The California Consumer Privacy Act

The California Consumer Privacy Act (CCPA) is a comprehensive consumer protection law set to take effect on January 1, 2020. The CCPA will apply to a wide range of businesses that handle Californians' personal information, obligating such businesses to comply with a host of new requirements governing their collection, use, and sharing of PI. The definition of PI contained in the CCPA is the broadest formulation of protected information in US law. It applies to all information that "identifies, relates to, describes, is capable of being associated with, or could reasonably be linked, directly or indirectly, with a particular consumer or household," including name, email address, biometric information, IP address, device identifiers, and browser-derived information (such as information stored in cookies, web beacons, and web pixels) [16].

De-identification techniques

De-identified data can be collected, processed, stored, or shared for a wide range of applications and purposes [5]. Each application requires the de-identified data to possess specific properties in order to accomplish its purpose. It is therefore necessary to preserve these properties after de-identification. For example:

- Testing of software applications requires data that pertains to or emulates certain characteristics of the anticipated real data in order to achieve behavior under the test as close as possible to the conditions that will apply during use of the application.
- Statistical reporting includes collecting data at the level of individual data principals and generating statistical reports for a population on certain characteristics or events.
- Publishing of data for research purposes [also known as privacy-preserving data publishing] often involves sharing sensitive data at the level of individual data principals.
- Performing data analytics on behalf of another party [also known as privacy-preserving data mining] requires the transfer of data at the level of individual data principals as well as of statistical data.
- Accessing and processing of sensitive, truthful, unencrypted data at the level of individual data principals by authorized internal parties in data centers.
- Linking data to its corresponding data principal in certain cases by specially appointed parties.

Data minimization (i.e., limiting the data to what is directly relevant and necessary to accomplish a specified purpose) at the earliest possible stage typically makes the task of data de-identification easier.

Pseudonymization

The term "pseudonymization" is referenced in the European Union's General Data Protection Regulation (GDPR) and refers to a category of de-identification techniques that involves replacing a data principal's identifier (or identifiers) with indirect identifiers specifically created for each data principal. As such, pseudonymization is a technique that enables linking of associated records from different datasets without revealing the identities of the data principals. The relation between different privacy-enhancing data de-identification terminology and classification of techniques is defined in "ISO/IEC 20889" [5].

A pseudonymization process generates supplementary information that can include identifiers removed from the original dataset, pseudonym assignment tables, or cryptographic keys. Such information can be used in the process of a controlled re-identification. To protect this information from re-identification attacks, appropriate "technical and other organizational measures need to be applied to such supplementary information in accordance with the organization's objectives and re-identification risk assessment."

Pseudonyms can be "cryptographically derived from the values of the attributes that they replace through encryption or hashing" [5]. Such a process is sometimes referred to as *key coding* the attributes in the dataset. It is important to note

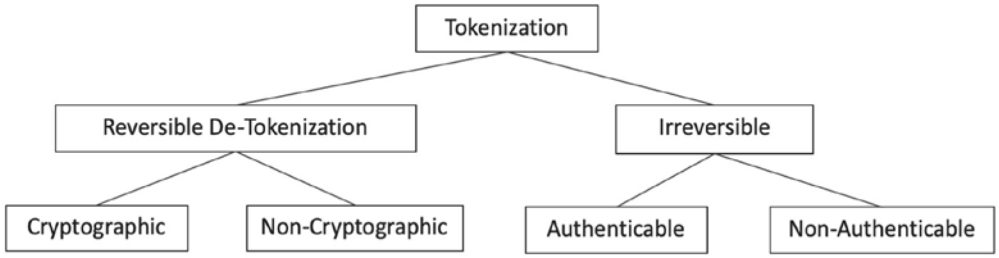

Figure 2 – Tokenization classification

that, given the appropriate key, encrypted attributes can be decrypted using the corresponding algorithm, while hashing is a one-way mathematical process.

Tokenization

Tokenization is a process by which a surrogate value, called a "token," replaces the primary account number (PAN) and, optionally, other data. The tokenization process may or may not include functionality to exchange a token for the original PAN ("de-tokenization"). The security of an individual token "relies predominantly on the infeasibility of determining the original PAN knowing only the surrogate value (i.e., token)" [6] Different classes of tokens may exist; these are created through distinct mechanisms and may support different use cases. In general, tokens are either created by a mathematical process (e.g., cryptographic function) or by a non-cryptographic process (e.g., data look-up through a database function).

Figure 2 illustrates classification of different tokenization approaches according to the PCI Security Standards Council. Tokenization can provide protection for data in storage or when processed and make the protected data useful in business processes for operations on PII and PCI data.

On-premise tokenization might result in a higher risks due to sensitive data remaining in the on-premise environment and the associated personnel and hardware costs. A private cloud-based tokenization on the other hand can provide significant reduction in PCI DSS scope and risk since sensitive data is removed from the on-premise environment and isolated from the public cloud.

Vault-based or vault-less

Vault-based tokenization (VBT) is using a vault for token storage and can be suitable for any centralized cloud deployment or centralized token generation platform. Synchronization of token generation across multiple distributed token vaults can be a challenge with this tokenization model. CPU impact and latency with VBT are typically like a database lookup query transaction to access the vault.

Vault-less tokenization (VLT) is based on an algorithm and potentially randomized memory data to generate tokens in a state-less model. VLT is suitable for on-premise deployment or distributed token generation where high performance and low latency could be required. VLT could typically be used in transaction switches and data warehouse systems. CPU impact with VLT is typically like AES encryption operations.

Election Systems Inc. needed a way for customers to pay for products online using a web browser. They implemented tokenization via payment application using a **vault-based tokenization** approach, hosted in a private cloud that reduced the PCI DSS scope and reduced their on-premise attack surface to enable integration with several payment processors. An on-premise vault-less tokenization would create a larger card holder data environment and increase the attack surface. They are now not locked into a single processor or third party via the centralized private cloud vaulting of payment tokens. With this solution, their settlement applications and call center applications are still in PCI DSS scope and access tokens and transaction details when needed.

The company is also using **vault-less tokenization** of PII data for GDPR compliance and PI data for CCPA compliance in their data warehouse to support high-volume performance, avoiding any remote access to a database vault. This on-premise VLT tokenization approach can be more than 10 times faster that format-preserving encryption or a vaulted tokenization solution.

Format-preserving encryption

Format-preserving encryption (FPE) is a method transforming data that is formatted as a sequence of symbols in such a way that the encrypted form of the data has the same format, including length, as the original data (see figure 3).

For example, a format-preserving-encrypted nine-digit Social Security number is a sequence of nine decimal digits. FPE facilitates the de-identification or pseudonymization of sensitive information, as well as the retrofitting of encryption technology to legacy applications where a conventional encryption mode is not feasible. NIST "SP 800-38G: Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption" was published in March 2016 in order to specify and approve the FF1 and FF3 methods for format-preserving encryption [8].

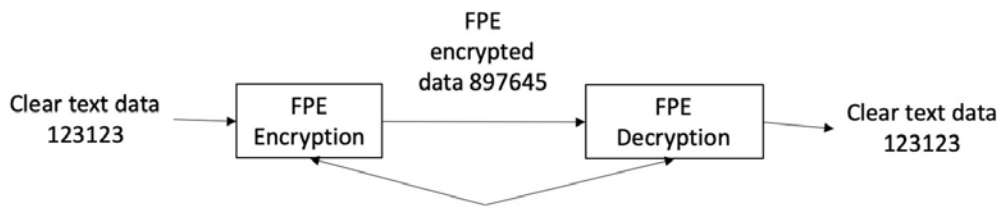

Figure 3 – Format-preserving encryption

Since the release of this publication, several sets of researchers have identified vulnerabilities when the number of possible inputs (i.e., the domain size) is sufficiently small. In response to the analysis of Durak and Vaudenay on FF3, NIST announced in April 2017 the intention to either revise the FF3 specification by reducing the size of its tweak parameter from 64 bits to 48 bits, as suggested by the researchers in their paper, or to withdraw FF3. In response to the analysis of Hoang, Tessaro, and Trieu, building on earlier work with Bellare, the recommendation was strengthened to a requirement: The minimum domain size for FF1 and FF3-1 in Draft SP 800-38G Revision 1 is one million. In this revision of SP 800-38G, the specifications of the two encryption methods, called FF1 and FF3-1, are updated in order to address potential vulnerabilities when the domain size is too small.

FPE provides protection for data in storage or when processed and makes the protected data useful in business processes for operations on PII, PI, and PCI data. FPE is suitable for deployment on-premise and in public or private clouds. CPU impact with FPE is typically 10 times more than AES encryption.

Our company decided to use format-preserving encryption to encrypt its sensitive client data before sending it to cloud-based Salesforce.com. This approach is highly transparent to most functions in Salesforce, and they considered to add homomorphic encryption for functions that need Salesforce to add up salary values without exposing cleartext values to Salesforce.

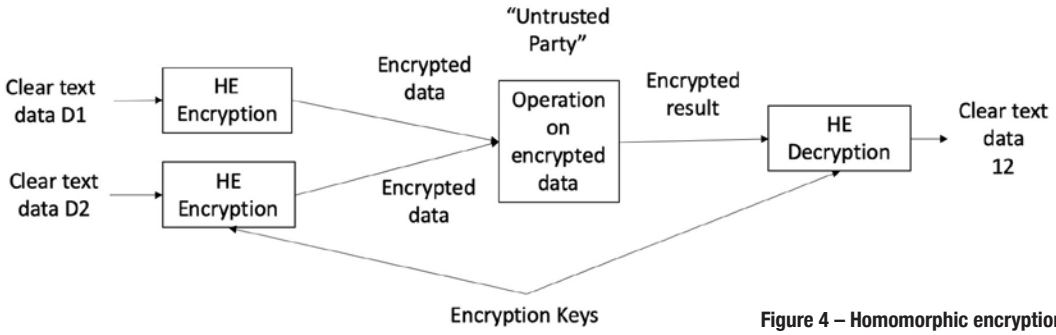

Figure 4 – Homomorphic encryption

Homomorphic encryption

The basic concept is to be able to encrypt some data and then still perform some useful calculations or process the data without decrypting (and thus exposing) it [18]. Figure 4 illustrates homomorphic encryption when used in secure multi-party computation (SMPC), the goal of which is to enable a group of independent data owners who do not trust each other or any common third party to jointly compute a function that depends on all their private inputs.

SMPC and secure outsourced computation are based on one party owning the data and a second party performs computation on the encrypted dataset without learning anything about the input data, intermediate values, or results. Homomorphic encryption schemes are currently attracting a lot of attention because they allow for the preservation of confidentiality of sensitive data in cloud computing. However, as pointed out in "Encryption Performance Improvements of the Paillier Cryptosystem," fully homomorphic encryption schemes, which enable both multiplication and addition operations on encrypted data, are currently still inefficient in practical settings. For this reason, "research efforts are also directed at different classes of homomorphic encryption, one of which is partially homomorphic encryption" [3].

These schemes allow performing one specific operation on encrypted data, for example addition or multiplication. It al-

lows performing sums on encrypted data, which is important in many use cases such as encrypted SQL databases [4], machine learning on encrypted data [1], and electronic voting [2][7].

Our company is developing and selling election support software applications globally. With the core use case in the election process, the voter is provided with a tracking code that, when voting is complete, he can enter into a website to confirm his vote was counted and not altered; the website will not display the actual votes [2]. In the ElectionGuard software development kit (SDK) from Microsoft this verification feature is enabled by homomorphic encryption [3], which allows mathematical procedures—like counting votes—to be done while keeping the data of people's actual votes fully encrypted [11]. This tracking code enabled by homomorphic encryption ensures that voters will be able to independently verify with certainty that their vote was counted and not altered. This will also enable voting officials, the media, or any third party to use a "verifier" application to similarly confirm that the encrypted vote was properly counted and not altered. This use of homomorphic encryption can enable end-to-end verifiable elections.

Masking

Masking typically involves removing all direct identifiers from the dataset or removing a portion of a direct identifier of a field. The term refers to a "de-identification technique that involves potentially stripping out some or all of the additional remaining identifying attributes for all records in the dataset." However, "removing a portion of a direct identifier so that it is no longer a direct identifier but still a unique identifier is considered to be a pseudonymization technique."

After masking has been performed, typically additional de-identification techniques are applied to the dataset [5].

Since masking is a one-way process, not reversible, it may be suitable in analytical systems and for application test environments, but typically not for operational transaction systems. This is suitable for any cloud or on-premise deployment model.

Our company is masking PII and PI data in the dev/test systems as a one-way de-identification that is highly transparent to applications and databases. Their concern is that their masking approach needs to be re-evaluated frequently to verify that the inference risk is not increasing due to the availability of additional public databases.

Hash functions

A hash function takes as input a key, which is associated with a datum or record and used to identify it to the data storage and retrieval application. The keys may be fixed length, like an integer, or variable length, like a name. In some cases, the key is the datum itself. The output is a hash code used to index a hash table holding the data or records, or pointers to them [13].

The function is a one-way function that does not preserve the data type or length of the input value. The most often used hash functions are SHA-1 and SHA-256, which produce 160- and 256-bit hashes, a binary long string, respectively (expressed as 40 and 64 characters). Since hashing is a one-way process, not reversible, it may be suitable in systems to verify the identity or integrity of data values and typically not useful for data protection in operational transaction systems.

A hash function is any function that can be used to map data of arbitrary size to fixed-size values [19]. The values returned by the function are called hash values, hash codes, digests, or simply hashes. The values are used to index a fixed-size table called a hash table. Use of a hash function to index a hash table is called hashing or scatter storage addressing. Hash functions and their associated hash tables are used in data storage and retrieval applications to access data in a small and nearly constant time per retrieval, and storage space is only fractionally greater than the total space required for the data or records themselves. Hashing is a computationally and storage space efficient form of data access that avoids the non-linear access time of ordered and unordered lists and structured trees, and the often exponential storage requirements of direct access of state spaces of large or variable-length keys.

Various techniques can be used to create pseudonyms. The choice of technique is based on factors such as the costs of creating the pseudonyms, the collision-resistance factor of a hash function (i.e., the probability of two inputs hashing to the same output), and the means by which the data principal can be re-identified for the purposes of a controlled re-identification [5].

A good hash function satisfies two basic properties:

1. It should be very fast to compute

2. It should minimize duplication of output values (collisions)

Hash functions rely on generating favorable probability distributions for their effectiveness, reducing access time to nearly constant. High table loading factors, pathological key sets, and poorly designed hash functions can result in access times approaching linear in the number of items in the table [13].

Cryptographic hash functions are specified in ISO/IEC 10118 [12]. NIST has approved hash algorithms in "Approved Hash Algorithms" [13] and in two federal standards: "FIPS 180-4, Secure Hash Standard" [14] and "FIPS 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions" [15]. FIPS 180-4 specifies seven hash algorithms: SHA-1 and the SHA-2 family of algorithms: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, and SHA-512/256.

FIPS 202 specifies the new SHA-3 family of permutation-based functions based on KECCAK with four fixed-length hash algorithms: SHA3-224, SHA3-256, SHA3-384, and SHA3-512; and two closely related, "extendable-output" functions: SHAKE128 and SHAKE256.

Cloud-access security brokers

A cloud access security broker (CASB) is on-premise or cloud-based software that sits between cloud service users and cloud applications, monitoring all activity and enforcing security policies [20]. It offers a variety of services, including but not limited to monitoring user activity, warning administrators about potentially hazardous actions, enforcing security policy compliance, and automatically preventing malware [9]. It can be implemented as an encryption gateway, and sensitive data can be encrypted or tokenized before sending it to the cloud, avoiding exposure of encryption keys and security control processes on premise.

CASBs that deliver security must be in the path of data access, between the user and the cloud. Architecturally, this might be achieved with proxy agents on each end-point device or in agentless fashion without requiring any configuration on each device [10]. I like this separation of security controls, but I've experienced a few issues with application transparency when using a CASB to encrypt or tokenize data before sending it to Saleforce.com and other SaaS applications. The SaaS application may in some cases need clear text data to function properly.

Boiling it down

Thus, the question how to protect data on-premise or in the cloud really has an "it depends" answer. Organizations are increasingly adopting a hybrid model with a need to protect data on-premise, in a private cloud, and in the public cloud. Things to consider include whether the data is being protected from unauthorized access when stored, being transmitted, or processed.

Figure 5 illustrates a mapping of different data security and privacy techniques to deployments in different environments, including on-premise, public, and private clouds.

Tokenization and FPE can be highly transparent to most business processes and databases and be used in most operations on personally identifiable information and other types of short data strings. When using tokenization or cryptographic tools, the data is being protected from unauthorized access when stored, transmitted, processed, and for many business use cases.

When using the suppression techniques—differential privacy or the K-anonymity model—the data is being protected from unauthorized access when stored, but not when being transmitted, or processed. These approaches are based on one-way transformations and are typically used in analytical applications and test systems. The approaches can be used to provide privacy in any deployment model. The K-anonymity model is based on a cleartext database, and differential privacy should use a protected database. These models are useful for limit-

Privacy enhancing data de-identification terminology and classification of techniques			Data Warehouse	Centralized	Distributed	On-premises	Public Cloud	Private Cloud
De-identification techniques	Tokenization	Vault-based tokenization		y				y
		Vault-less tokenization	y	y	y	y	y	y
	Cryptographic tools	Format preserving encryption		y	y	y	y	y
		Homomorphic encryption			y		y	
	Suppression techniques	Masking	y	y	y	y	y	y
		Hashing	y	y	y	y	y	y
Formal privacy measurement models	Differential Privacy	Server model	y	y	y	y	y	y
		Local model	y	y	y	y	y	y
	K-anonymity model	L-diversity	y	y	y	y	y	y
		T-closeness	y	y	y	y	y	y

Figure 5 – Mapping data security and privacy techniques to deployment on-premises, public, and private clouds

ing exposure of individuals when collecting and analyzing personally identifiable information. These techniques are less suitable in operational transaction systems and more useful in analytical systems and development systems in different deployment models.

Figure 6 compares business utility and usefulness of protected data fields versus the security level of some major data protection techniques. The key point is that tokenization and FPE can provide a high level of utility and security since they are preserving the length and data type of the original input values.

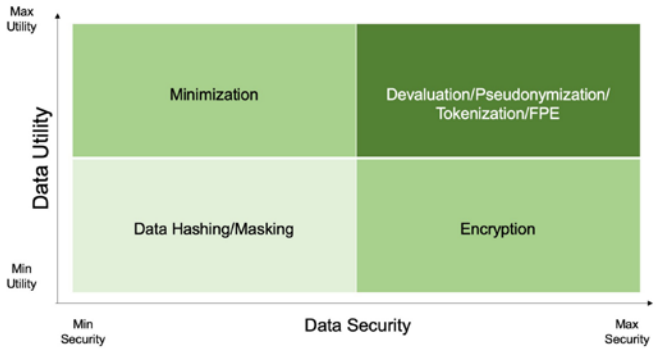

Figure 6 – Business utility versus security level of different data protection techniques

Conclusion

When making the decision on which de-identification techniques to deploy, there are several things to consider including how the data is used and secured.

Our example company, Election Systems Inc., used homomorphic encryption to keep the data of people's actual votes fully encrypted. It implemented a vault-based tokenization approach, hosted in a private cloud that reduced the PCI DSS scope and reduced the on-premise attack surface to enable integration with several payment processors. And they are using vault-less tokenization in their data warehouse to support high-volume performance.

Format-preserving encryption was used to encrypt their sensitive client data before sending to cloud-based Salesforce.com. They are using a data analytics application and need to comply with GDPR and CCPA and reduce the risk of identifying individuals by applying differential privacy. And they are masking PII and PI data in the dev/test Systems as a one-way de-identification that is highly transparent to applications and databases. Their concern is that their masking approach needs to be re-evaluated frequently to verify that the inference risk is not increasing due to the availability of additional public databases. Implementing the periodic cleanser filter process in DP and KA could help with this issue.

References

1. R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine Learning Classification over Encrypted Data," NDSS Symposium, 2015.
2. I. Damgard, M. Jurik, and J. B. Nielsen, "A generalization of Paillier's Public-Key System with Applications to Electronic Voting," International Journal of Information Security 9, no. 6, pp. 371-385, 2010.
3. Christine Jost, et al. "Encryption Performance Improvements of the Paillier Cryptosystem," Ericsson Research – https://eprint.iacr.org/2015/864.
4. P. Grofig , M. Härterich, I. Hang, F. Kerschbaum, "Experiences and Observations on the Industrial Implementation of a System to Search over Outsourced Encrypted Data," ResearchGate – https://www.researchgate.net/publication/288451423_Experiences_and_observations_on_the_industrial_implementation_of_a_system_to_search_over_outsourced_encrypted_data.
5. ISO. "Privacy Enhancing Data De-Identification Terminology and Classification of Techniques," – https://webstore.iec.ch/preview/info_isoiec20889%7Bed1.0%7Den.pdf.
6. PCI Council. "Tokenization Product Security Guidelines, Version: 1.0, April 2015," PCI Security Standards Council – https://www.pcisecuritystandards.org/documents/Tokenization_Product_Security_Guidelines.pdf?agreement=true&time=1570880509645.
7. Pascal Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," Advances in Cryptology - EUROCRYPT'99, vol. 1592 of Lecture Notes in Computer Science, pp. 223-238, 1999.
8. NIST, "SP 800-38G: Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption," NIST – https://csrc.nist.gov/publications/detail/sp/800-38g/rev-1/draft.
9. "Cloud Access Security Broker," Wikipedia – https://en.wikipedia.org/wiki/Cloud_access_security_broker.
10. Market Guide for Cloud Access Security Brokers, October 2016, https://www.gartner.com/en/documents/3488119.
11. Tom Burt, "ElectionGuard Available Today to Enable Secure, Verifiable Voting," Microsoft (Sep 24, 2019) – https://blogs.microsoft.com/on-the-issues/2019/09/24/electionguard-available-today-to-enable-secure-verifiable-voting/.
12. ISO. "ISO/IEC Standard 10118-3 IT Security Techniques — Hash- functions — Part 3: Dedicated Hash-Functions" – https://webstore.iec.ch/preview/info_isoiec10118-3%7Bed4.0%7Den.pdf.
13. NIST CSRC. "Hash Functions," – https://csrc.nist.gov/projects/hash-functions.
14. NIST CSRC. "FIPS 180-4, Secure Hash Standard" – https://csrc.nist.gov/publications/detail/fips/180/4/final.
15. NIST CSRC. FIPS 202, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions – https://csrc.nist.gov/publications/detail/fips/202/final.
16. Helen Goff Foster and Austin Smith, "The California Consumer Privacy Act: What Financial Services Providers Need to Know," Davis Wright Tremain LLP – https://www.dwt.

com/blogs/privacy--security-law-blog/2019/06/the-california-consumer-privacy-act-what-financial.

17. Fernando Montenegro and Jeremy Korn, "Cloudvisory Seeks to Help Navigation through WellDefined Journey to Cloud Security," 451 Research Report, Cloudvisory – https://www.cloudvisory.com/whitepaper-451-research-report.html.
18. Robert Slade, "Homomorphic Encryption," ISSA Journal, November 2019 – https://www.bluetoad.com/publication/?i=632044&pn=Page+7.
19. "Hash Functions" – https://en.wikipedia.org/wiki/Hash_function.
20. "Cloud Access Security Broker" – https://en.wikipedia.org/wiki/Cloud_access_security_broker.

About the Author

Ulf Mattsson has been an ISSA member and participated in X9 standards for more than fifteen years. He has more than 70 issued US patents, contributed to the development of PCI DSS, and has been developing products and services for robotics, ERP, CRM, data encryption and tokenization, data discovery, cloud application security brokers, and web application firewalls. He may be reached at ulf@ulfmattsson.com.

Women in Cybersecurity: An Interview with Betty Burke

Continued from page 8

we both came away with different perspectives. It was a great learning experience we both took with us as we moved onto other jobs. We have continued to mentor each other throughout our careers. Now, as we are in different situations, we know when we need to channel the other person's strengths, which means she needed to learn to sit quietly and listen, and I needed to speak up and give my input.

Obstacles and challenges

Q: Did you experience any challenges along the way?

A: An early challenge, personally, was not having the confidence or ability to participate in meetings. Maybe those who are introverted, like myself, can relate to this. There were two things that were helpful for me. First, I made the decision early in my career that I needed to learn how to talk in meetings. I made a decision to have a least one comment or question at every meeting I was in. I was very nervous. Second, came several years later, when I joined Toastmasters. I learned how to lead by practicing at every meeting and taking on officer roles that taught me how to create goals and how to obtain those goals. Third, I took what I learned at Toastmasters and applied it when I led the MN ISSA Chapter. My ISSA-related work has been critical in helping me to take on leadership roles in my professional career.

Q: What is the most difficult obstacle you've had to overcome in your professional life?

A: My most difficult obstacle was having a boss that was a bully, and it was difficult as I was working through an episode of prolonged depression. I felt like I was not skilled in information security and that I didn't know anything. I was able to overcome that by being involved in my local ISSA chapter, where I had many connections and avenues for fellowship and support. Having that level of support from the community was so important in getting me through that period of my life.

ISSA Impact

Q: How has ISSA impacted your cybersecurity career?

A: Getting involved in ISSA has been wonderful to learn about new and emerging technologies. By attending meetings in person, it helps to discuss with others what their security pain points are or ask for recommendations on how to solve a problem. Also, getting involved at the leadership level is a fantastic way to meet others in the field at all levels of their career. Networking with others has been important when looking for new opportunities. It helps to keep a pulse on the security community and where new opportunities exist. As a community, we help each other if and when we are looking for a new job. For example, when someone was looking to get into a local company, I was able to steer them to talk to an individual who could put in a referral, which guaranteed at least a phone interview. This individual did get the job.

Q: As an ISSA chapter president, what chapter accomplishments have you been instrumental in seeing through to completion?

A: Over the past six years as chapter president, I am very proud to have been leading our board in six areas: 1) increasing membership significantly, 2) keeping a full board, 3) increasing sponsorship and providing value to our sponsors, 4) providing valuable educational content to our audience, 5) contributing to various local organizations such as Aspirations in Computing, and 6) giving scholarships to local colleges.

Q: Did you have a particular leadership focus that led to your chapter's success?

A: Yes, we had two focus areas: Our board accountability and our chapter's operational processes. For example, every year, we would have to rethink the elections process, and we now have that documented so that we can have elections run consistently every year. We focused on providing quality content