Can Homomorphic Encryption Reduce the Security Risks in Telemetry Post Processing Environments?

Jeff Kalibjian
Perspecta

KEYWORDS

homomorphic encryption, data security, disk level encryption, file level encryption, database encryption, application encryption, data analysis, function obfuscation, data malleability

ABSTRACT

Homomorphic encryption [1, 2] is a branch of cryptography in which data transformation operations can be performed on already encrypted data—promising better protection of data as the data no longer needs to be decrypted in order for specific analysis operations to be performed. Thus, better security is achieved by absolutely minimizing the amount of time sensitive data is potentially exposed. After reviewing homomorphic encryption principles, system level architectures will be presented discussing where homomorphic encryption may best fit in the generally accepted data security taxonomy involving disk, file, and application encryption. Emphasis will be placed on application to telemetry post-processing environments.

INTRODUCTION

Data security challenges in telemetry post processing activities are noteworthy because telemetry data is very valuable to an organization. Telemetry data provides valuable insight into how systems may or may not be working properly and if these systems have national security impact the telemetry data is even more sensitive. Data security in the enterprise spans data at rest, data in motion, data sent through e-mail, data used by applications, and data backed up. Of particular import is data at rest security and its impact on data used by applications. This is because when data needs to be used by an application the data is at most risk of compromise since the application will need to decrypt the data elements it needs to process. The mechanism used to secure the data at rest (e.g. disk encryption, file encryption, database encryption, or application encryption) will ultimately determine the envelope of time decrypted data will be vulnerable. However, what if the application did not need to decrypt the data to process it? What if it could merely apply transformations to the encrypted data and have it be the same as if the data was decrypted, had the same transformations applied, and then re-encrypted? This would truly be revolutionary as there would then be no envelope of time data would need to be decrypted to undergo transformation; thereby, eliminating the possibility of compromise during this type of processing. However, before examining these technological possibilities it will be important to

review both the merits and perils of achieving data at rest security with disk encryption, file encryption, database encryption, and application encryption.

DATA AT REST ENCRYPTION PRACTICES

There are basically four encryption approaches used to protect enterprise server and database data at rest. Self-encrypting disks provide the least amount of data security to the organization. This is because when the disk powers on and the proper authentication/password information is presented, the entire content of the disk is available for decrypted use by the host operating system (OS) and any higher layer application; with no further protections on the data except what the operating system might offer. Thus, the only real benefit of encryption at this level is if the disk is physically stolen, or inadequately erased when it is decommissioned because the disk will be in its self-encrypted state when it is both powered down, or when powered up and an incorrect password to decrypt the disk has been presented.

With file based encryption, dedicated processes run just above the operating system to seamlessly decrypt the files sensitive data resides in when the data is going to be needed by an application, adding significant protections beyond self-encrypting disks. Typically, a dedicated software based agent performs this role and is deployed at just above the OS level to intercept all file read/write operations, evaluate established policy and determine if applications and/or users attempting to access data in files (flat files, database files, etc.) have a right to do so. The truly enormous advantage of encryption at the file level is no modifications need to be made to existing applications that use the data stored in files or databases. However, the downside is that for the period of time the files are decrypted (this may occur when an application or even database administrator needs to access the data), all the data in the files are theoretically vulnerable to compromise. Thus, if files being decrypted for an application happened to be database files, then for the period of time those files were decrypted, all the data in the database would be potentially vulnerable. On the positive side, another advantage of this approach is that key management policies and data access policies can be centrally orchestrated with dedicated (and redundant) management applications controlling all the file encryption/decryption agents.

Database encryption can almost be considered a subset of application encryption. Basically in database encryption very specific subsets of the data in the database can be encrypted utilizing the native encryption capabilities of the database. The downside of this approach is that if the organization has many different vendor databases; each database will have its own key management and policy applications, making it somewhat tedious to centrally manage all the policies. These solutions also have to decrypt and re-encrypt all encrypted data when a key rollover occurs. This is a time-consuming process which in many cases takes the database off line during the process. However, on the positive side, implementing database security may not necessarily require application changes, as the database will take responsibility for seamlessly decrypting the data before it is presented to the application---but there will be downtime if a production database which is not using decryption, is enabled to do so.

Finally**,** with application level encryption, very specific subsets of data can be encrypted. This granular control affords data in the enterprise to have the most protection as data is essentially encrypted to the very last moment until it is required by the application. While application level encryption affords the organization the best security; it also involves some complexity in that the applications will need to be specifically modified to leverage the granular encryption services. It is at the application level that both tokenization and Format Preserving Encryption (FPE) approaches can also be introduced. In tokenization a non-sensitive entity is substituted for a sensitive data item where the substitution method is very difficult to reverse if the tokenization system is not present. In FPE, the output of an encryption process is encrypted but in the same format as the input. For instance, if the input was in a credit card number format, the output would be in the same format. This allows the FPE encrypted items to be directly placed in the same database schema intended for use with the original non-encrypted data. Compounding the complexity is the need to migrate the existing data sets to the tokenized or format preserved encryption form. However, this is a one-time operation. Also note with application level encryption overall application performance can be improved as the data is protected once and then is used in its protected state throughout the network. There is no need for encrypt/decrypt cycles at every server in the network. Security policy is based on the data element no matter where it resides; it is not specific to a platform or an application.

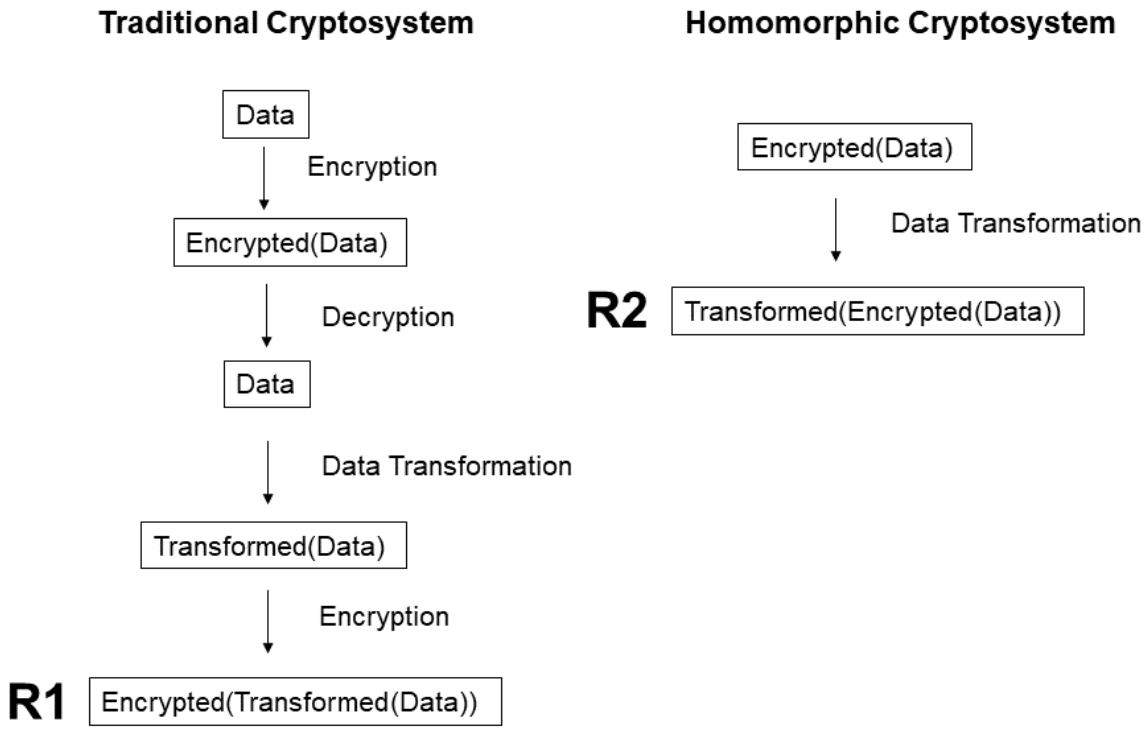

Figure 1. Contrasting a traditional cryptosystem with a homomorphic encryption system. Decryption of R1 and Decryption of R2 will yield the same result!

Although self-encrypting disk technology, file level encryption, database encryption and application level encryption are different methodologies for securing data; they do all have one thing in common; when sensitive data needs to be used, data must be decrypted leaving it vulnerable to compromise—the difference being the amount of time sensitive data will need to remain unencrypted and the volume of the data that remains unencrypted over the time period the data is being used. With disk encryption the amount of time is long and the volume is great as after the password is input to decrypt the entire disk is accessible to the host operating system. With application level encryption the amount of time is short and the volume small, as only the data element the application needs to process must be decrypted and only for the short time it needs to be directly used by the application. File level encryption is somewhere in between.

A POTENTIALLY SUPERIOR APPROACH: HOMOMORPHIC ENCRYPTION

While application level encryption affords an organization the absolute minimum in sensitive data exposure time, homomorphic encryption totally eliminates it. This is because homomorphic encryption systems have a property that transformational operations to data can be directly

Table 1. Challenges with Homomorphic Encryption

Challenge Item	Comment
New cryptographic algorithms	Of AES, Triple DES and RSA only RSA is partially homomorphic and only with respect to multiplication. New homomorphic crypto systems will need to be vetted by the user community
Computational requirements	Homomorphic systems created to date require big compute power for even simple transformational operations to complete
Malleability	Encrypted data can be altered without decryption leading to the question: how to identify the originally encrypted data? Or how to track what transformations have already been done to the encrypted data?
Functions to operate on the encrypted data not private	While homomorphic encryption protects the privacy of data operated on; there is no provision for protecting the function(s) that will be applied to the encrypted data; potentially giving adversaries clues with respect to the content of the encrypted data.
NIST FIPS 140-2 evaluation not likely possible	A FIPS 140-2 certification requires use of NIST certified cryptographic algorithms. Homomorphic encryption systems create new cryptographic systems that will require a from scratch evaluation by vetting entities.

applied to encrypted data with the result being as if the data was decrypted, the transformations applied and then the data re-encrypted again, see Figure 1. It should be pointed out that homomorphic encryption is probably best applied at the application level; so individual data elements that are encrypted can be directly transformed. The concept of leveraging homomorphic encryption on a complete data file somewhat defeats the intent of homomorphic encryption, as arbitrary operations to the entire file would be difficult to standardize.

HOMOMORHIC ENCRYPTION

The mathematics underlying homomorphic encryption are beyond the scope of this paper. Instead, the characteristics of these new homomorphic systems will be discussed in order to understand the implications of their use. Beyond its unique attribute with respect to performing mathematical operations directly on encrypted data without the need to decrypt, homomorphic encryption is not a total panacea! It has several characteristics which will make it challenging to successfully deploy in the marketplace. The concept of homomorphic encryption is not new. Before Craig Gentry disclosed he had created a fully homomorphic encryption system in 2009 (this was his PhD thesis subject at Stanford University), many people were engaged in such research. In fact, it was known that other crypto systems previously created were partially homomorphic. For example, the famous RSA encryption algorithm developed by Adi Shamir, Ron Rivest and Len Adleman [3] is homomorphic with respect to multiplication; however, not addition; notably both arithmetic functions are needed in order for any mathematical operation to be implemented. Unfortunately, the most popular symmetric ciphers in use today like Triple DES [4] and AES [5] are not homomorphic. And herein lies the first troubling characteristic with respect to homomorphic encryption---by definition, a new crypto system is being created that can deliver the unique functionality. One can't just somehow "bolt" homomorphism onto the existing old (dependable) crypto algorithms like RSA and AES. Of course the primary benefits of AES and RSA and Triple DES is they have been used for years and years and have been totally vetted by the scientific data privacy community. Any new homomorphic system coming out, will be just that: new and totally not vetted.

There are other issues of concern with homomorphic encryption. First the theoretic systems proposed would require a great deal of computational power to practically implement. Further, one must also be aware that while the data remains encrypted, there are no privacy guarantees for the algorithm(s) or transformation(s) that will be used on the encrypted data. In one sense, this could potentially leak information about the data being processed. In other words, if one knows the data transformations that will be of interest for data, one might be able to conclude or anticipate the type of data that will be operated on. Finally, homomorphic systems are malleable; i.e. they alter the encrypted data with no real mechanism for remembering what was done to the data or what the original data was. These somewhat problematic characteristics of homomorphic encryption are summarized in Table 1.

SECURITY CERTIFICATION IMPLICATIONS

In the security industry proving one's security product itself is secure is very important. The two most important certifications in this regard are the National Institute of Standards and

Technology (NIST) FIPS 140-2 standard [6] and the International Standards Organization (ISO) Common Criteria (CC) [7]. Many industry governance standards as well as governments themselves require their organizations/agencies only purchase Commercial Off the Shelf (COTS) security products that have achieved one or more of these certifications. FIPS 140-2 is a hardware specification that is used to evaluate hardware that protects cryptographic keys and performs sensitive cryptographic operations on data at rest and in motion. This hardware is often called a Hardware Security Module (HSM). There are four levels of merit in the specification, with each increasing level implementing more robust security safeguarding measures.

Common Criteria (CC) is an International Standards Organization and Standardization/International Electrotechnical Commission (ISO/IEC) standard and is recognized internationally by 25 countries. Common Criteria is really a methodology for evaluating security claims about a product. It's not looking at the entire product, only its security features. The CC includes specifying a Security Target (ST) document that defines the security functionality and assurance methodologies used to verify that those functionalities are actually achieved. The ST document also defines the Target of Evaluation (TOE)—which is the product or subset of product for which the methodology will be performed. A Protection Profile (PP) is a ST that is written in a general way for a particular product type—for example, firewall products or intrusion prevention products. Common Criteria has seven Evaluation Assurance Levels (EAL). The desired EAL level being pursued dictates the rigor of how those assurances are verified.

A natural question is: would it be possible for a homomorphic based encryption product to achieve a FIPS 140-2 or Common Criteria certification? With respect to FIPS 140-2 probably not. This is because FIPS 140-2 Level 1 assurance centers around use of NIST evaluated (i.e. certified) cryptographic algorithms (e.g. Triple DES, AES). Recall to engineer a homomorphic encryption system an entirely new cryptosystem must be created. Unless NIST were to certify that new cryptosystem; a FIPS 140-2 evaluation would not be possible despite the fact that FIPS 140-2 Levels 2, 3, 4 levels relate to how the HSM is secured (e.g. tamper evidence, data zeroization, etc.), not really the cryptographic algorithm used. Unfortunately, a Level 2, 3, or 4 evaluation requires Level 1 compliance. With respect to Common Criteria the story is more optimistic as there is no real reason why a homomorphic encryption system could not achieve a desired EAL level. The challenge would be defining a new ST and TOE for that system. If done in a general way, a PP could be defined that other homomorphic encryption system manufacturers could leverage if they wished to purse a CC evaluation.

TELEMETRY POST PROCESSING IMPLICATIONS

At first thought one might conclude that the data security in telemetry post-processing environments would be substantially enhanced by deploying homomorphic encryption. This would generally be true modulo one important factor: the robustness of the new crypto system

implemented to be homomorphic! Before general use in either public sector or private sector environments, there would need to be significant vetting of the crypto system to ensure it could not be easily compromised. Given that such a system was implemented correctly, organizations deploying such systems would significantly narrow their data security exposures---as data would no longer need to be decrypted in order to be transformed. However, another factor would need to be managed before practical use could be adopted: data malleability

As previously mentioned, by definition, homomorphic system data are malleable. This characteristic is not an insignificant issue in telemetry post processing environments. What transforms are applied to post processed data is a critical issue. Thus, post processing environments will need to establish an accountability mechanism to track the transforms applied to encrypted data. Once such a tracking capability was put into operation, a telemetry post-processing environment could maximally leverage a homomorphic encryption capability given adequate compute power to support the homomorphic computations.

SUMMARY

Homomorphic encryption is an exciting area of research in cryptography. There is no doubt that the underlying paradigm---being able to mathematically operate on encrypted data without the need to decrypt the data could transform data security in organizations. Thus, it is very germane to environments where telemetry post-processing and analysis are carried out. However, much more research needs to be done in order to overcome some of the troubling characteristics associated with homomorphic encryption including, the introduction of new cryptosystems that must be vetted for their security, computational horsepower required to practically implement, protecting the disclosure of the functions operating on the encrypted data and the data malleability side effect of homomorphic encryption. None of these obstacles are insurmountable and thus it is within reason to believe at some point in the nearer future homomorphic cryptosystems will be a viable option for telemetry post processing environments.

References

[1] Hayes, B, ***Alice and Bob in Cipherspace***, American Scientist, https://www.americanscientist.org/article/alice-and-bob-in-cipherspace

[2] Armknecht, F., Boyd, C., Carr, C., Gjosteen, K., Jaschke, A., Reuter, C., Strand, M., ***A Guide to Fully Homomorphic Encryption***, 2015, https://eprint.iacr.org/2015/1192.pdf

[3] Rivest, R., Shamir, A., Adleman, L., ***A Method for Obtaining Digital Signatures and Public Key Cryptosystems***, Communications of the ACM—Special 25th Anniversary Issue, Volume 26, Issue 1, Pg. 96-99, January 1983, Association for Computer Machinery

[4] Karn, P., Metzger, P, Simpson, W, RFC 1851, ***The ESP Triple DES Transform***, Internet Engineering Task Force, September 1995, http://www.ieft.org

[5] National Institute of Standards and Technology (NIST), ***FIPS 197 Advanced Encryption Standard (AES)***, NIST, November 2001, https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf

[6] National Institute of Standard and Technology (NIST), May 2001, ***Security Requirements for Cryptographic Modules***, National Institute of Standard and Technology, http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf

[7] International Organization for Standardization/International Electrotechnical Commission (ISO/IEC), April 2017, ***Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5***, International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf