Data Mediation with Enterprise Level Security[1]

Kevin E. Foltz and William R. Simpson
Institute for Defense Analyses
4850 Mark Center Dr.
Alexandria, Virginia 22311

[1] The publication of this paper does not indicate endorsement by the Department of Defense or IDA, nor should the contents be construed as reflecting the official position of these organizations.

ABSTRACT

Enterprise Level Security (ELS) is an architecture for enabling information sharing with strong security guarantees. It is built upon basic tenets and concepts that shape its component technologies and implementation. One challenge in sharing information is that the source and recipient of the information may use different units, protocols, data formats, or tools to process information. As a result, a transformation of the data is needed before the recipient can use the information. These conversions introduce potential security weaknesses into an ELS system, so an approach for enterprise-wide mediation is required. Methods in common use today, such as a man-in-the-middle (MITM) translation and an online mediation service, do not preserve the basic ELS tenets and concepts. This paper examines these existing approaches and compares them with two new approaches designed to preserve ELS security. It looks at the complete picture of security, performance, and ease of implementation, offering a framework for choosing the best mediation approach based on the data sharing context.

Keywords: Enterprise, Security, IT Security, System Design, Mediation, Confidentiality, Integrity.

1. INTRODUCTION

Adversaries continue to penetrate our network defenses and in many cases already exist within our network perimeter. They have infiltrated the online environment, jeopardizing the confidentiality, integrity, and availability of enterprise information and systems. The fortress model – hard on the outside, soft on the inside – assumes that the boundary can prevent all types of penetration [7], but this assumption has been proven wrong by a multitude of reported network-related incidents. A wiser assumption for data and information security practitioners is that the adversary exists within the network. The Enterprise Level Security approach starts with this assumption and offers a large set of security properties that work even in the face of embedded adversaries. Key design decisions include a distributed architecture and end-to-end security for all communication.

This paper describes a way to provide mediation services within an ELS framework. Mediation services present a unique challenge and a tempting target for embedded malicious entities because mediation takes place where data is changed but the normal end-to-end integrity verification methods are not feasible. A malicious entity that compromises a mediation service could selectively feed malicious content to an unsuspecting entity. Detection would be difficult because most entities only understand either the input format or the output format of data and cannot validate the translation. There is no perfect mediation approach, and this paper discusses various approaches and their tradeoffs. The following sections describe ELS, mediation challenges, potential mediation solutions, and how to choose a solution.

2. ENTERPRISE LEVEL SECURITY

The ELS design addresses five security principles:

- Know the Players – enforce bi-lateral end-to-end authentication;
- Maintain Confidentiality – use end-to-end unbroken encryption between data requester and provider;
- Separate Identity from Access and Privilege – use separate authentication and authorization credentials;
- Maintain Integrity – know that you received exactly what was sent;
- Require Explicit Accountability – monitor and log transactions.

A. Know the Players

In ELS, the identity certificate is an X.509 Public Key Infrastructure (PKI) certificate [1, 2]. This identity is required for all active entities, both person and non-person, as shown in Figure 1. PKI credentials are verified and validated. Ownership is verified by a holder-of-key check. Supplemental authentication factors may be required from certain entities, such as biometric data.

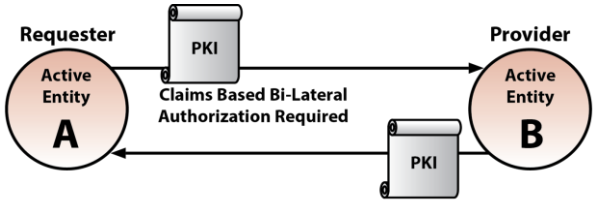

Figure 1: Bi-lateral Authentication

B. Maintain Confidentiality

Figure 2 shows how ELS establishes end-to-end Transport Layer Security (TLS) encryption through the numerous intermediaries that may route, scan, or process data between requester and application [3]. The intermediaries may view and manipulate the encrypted content, but they are not able to view or modify the raw unencrypted content without triggering an error at the endpoints.

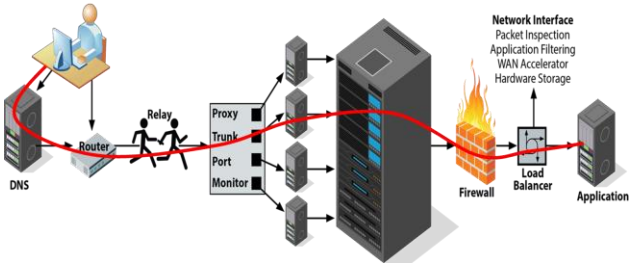

Figure 2: End-to-End Encryption

C. Separate Access and Privilege from Identity

ELS can accommodate changes in location, assignment, and other attributes by separating the use of associated attributes from the identity. Whenever changes to attributes occur, access claims are recomputed based on new associated attributes, allowing immediate access to required mission information. As shown in Figure 3, access credentials use the Security Assertion Markup Language (SAML). SAML authorization tokens used with ELS differ from the more commonly used single-sign-on (SSO) authentication tokens [4]. Authentication is performed through TLS using PKI credentials. This separation prevents a compromised SAML token from providing immediate access. The credential for access and privilege is bound to the requester by ensuring a match of the distinguished name used in both authentication and authorization credentials.

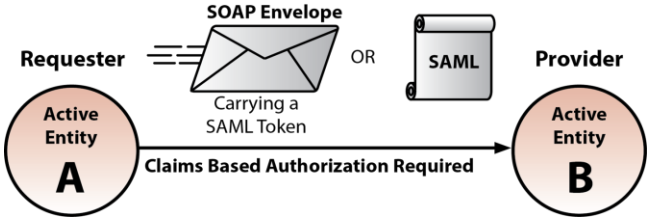

Figure 3: Claims-Based Authorization

D. Maintain Integrity

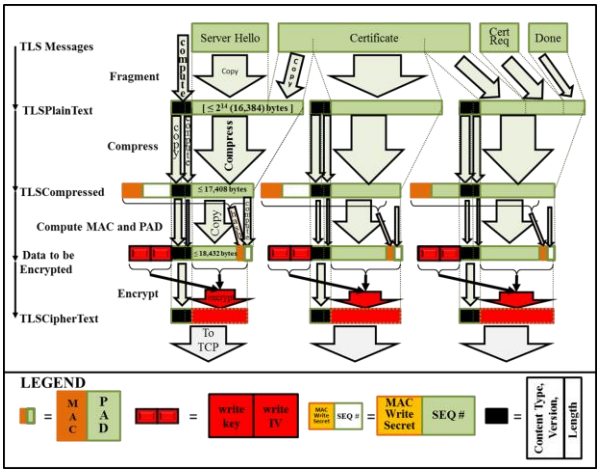

Figure 4: Integrity Measures

Integrity is implemented by end-to-end TLS message authentication codes (MACs), as shown in Figure 4. Chained integrity, in which trust is passed on transitively from one entity to another, is not used because it is not as strong as employing end-to-end integrity. At the application layer, packages (SAML tokens, etc.) are signed by the sender, and signatures are verified and validated by the receiver.

E. Require Explicit Accountability

As shown in Figure 5, ELS monitors specified activities for accountability and forensics. The monitor files are formatted in a standard way and stored locally. For enterprise files, a monitor sweep agent reads, translates, cleans, and submits records to an enterprise database for recording log records periodically or on demand. Local files are cleaned periodically to reduce overall storage and to provide a centralized repository for help desk, forensics, and other activities [5, 6].

By abiding with the tenets and principles discussed above, ELS allows users access without accounts by computing targeted enterprise claims. ELS has been shown to be a viable, scalable alternative to current access control schemas. A complete description of ELS basics is provided in [8].

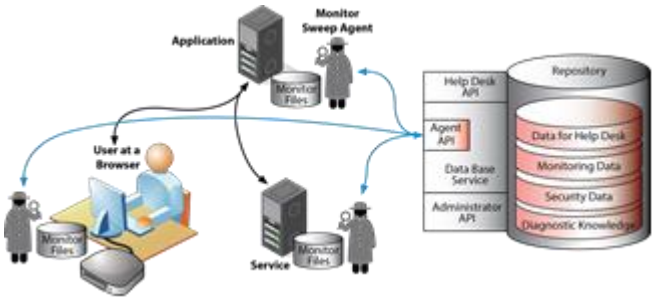

Figure 5: Accountability through Centralized Monitoring

3. THE SECURE MEDIATION PROBLEM

Data mediation is the process of transforming data from one format to another while preserving the original meaning. This is a common problem in large enterprises in which different groups use different methods to represent data. When the data is shared between groups, it is not useful in its native form and must be converted to a new format. Examples include the following conversions:

- Miles to kilometers,
- Address to latitude and longitude,
- Word processing document to PDF file,
- SQL database to XML database.

In general, these conversions may be arbitrarily complex or domain-specific. This paper addresses the enterprise-wide challenge of doing these conversions in a way that is consistent with ELS. It does not address the development or implementation of the conversion algorithms. It assumes a conversion method exists and addresses the challenge of distributing its use securely across the enterprise.

As part of ELS, certain important properties must be preserved. ELS does not allow any intermediaries to intercept or modify communication between two communicating entities. In the case of mediation, this means that mediation computation is not allowed to take

place between the sender and receiver on unencrypted data, either on the wire or by an explicit man-in-the-middle (MITM). For example, using an online translation site to browse foreign language websites does not fit the ELS model, since the translation site is acting as a MITM between the sender and receiver.

ELS requires end-to-end integrity of data. The receiver must know that the received data is what the sender actually sent. Again, the MITM translation does not work because the connection is only with the MITM, who can attest to the integrity of the MITM-to-receiver connection, but not the sender-to-receiver connection. ELS requires end-to-end integrity, not piecemeal integrity.

ELS attempts to minimize the number of external entities that must be trusted. The more trust that is required for a solution to work, the fewer options for deployment there are, and those options have more built-in vulnerabilities, since each required trust relationhip is a potential point of failure. In any communication, the receiver must trust the sender, since the receiver is requesting data from the sender. Similarly, the sender must trust the receiver. ELS provides an end-to-end bilateral authenticated TLS connection from trusted sender to trusted receiver, which includes confidentiality and integrity through its encryption and message authentication code (MAC).

When mediation is required between two entities, a third mediation entity is involved in addition to the two communicating entities. This mediation entity is trusted to accurately transform the sender's information to the receiver's information. It may be a local tool at the sender or receiver, or it may be a third party that performs the mediation. Although it appears that there is no mediation entity when the sender or receiver does the mediation, in effect it takes on the role of the mediation entity, so the other entity must trust it to perform mediation accurately. When third-party software is used to do mediation at an endpoint, the third-party software, and hence the third party providing the software, must be trusted because the endpoint simply executes this code without understanding what it does. In each case, there is a change in the data and the entity determining this change must be trusted to do it correctly because neither endpoint can independently verify that the input and output of the conversion correspond to each other.

In some cases, it may be possible to partially verify a conversion, such as converting a PNG image to a JPG image, because it is possible to look at the result and compare it to the original. But this is only a superficial check. A receiver still needs to trust that the conversion process has not inserted additional information or malicious code. Most endpoints that require mediation services do not have an intimate knowledge of the formats or their potential exploits and vulnerabilities. They simply want to process data in a particular format.

4. APPROACHES

Different approaches to mediation are presented below and compared against the requirements for ELS.

A. MITM Mediation

The first potential solution is a MITM mediation service, shown in Figure 6. The requester connects to the mediation service and requests data from a provider. The mediation service then retrieves the desired data, transforms it as necessary for the requester, and provides it to the requester.

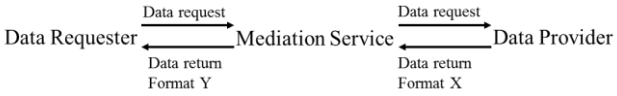

Figure 6: Man-in-the-Middle Mediation

This mediation approach does not provide a connection between requester and provider with confidentiality because the mediation service can view all traffic between the requester and provider. It also does not provide a connection with integrity. Although each of the connections with the mediation service has integrity, this is not the same as integrity from requester to provider because changes to the request and response by the mediation service cannot be identified by the requester or provider. Because the connections lack confidentiality and integrity, the data the requester receives lacks confidentiality and integrity guarantees. Both the requester and provider must trust the mediation service for this solution to work.

The data that must be transmitted in this solution is the data between source and mediation service in its original format and the data between mediation service and requester in its new format, so this method requires two data transmissions. The mediation computation is performed at the mediation service.

B. Mediation Service

The second potential solution is a mediation service that the requester calls to do mediation, shown in Figure 7. The components of this solution are similar to the MITM solution, but instead of acting as a MITM for the connection between the requester and provider, the mediation service is explicitly called by the requester to mediate the received data.

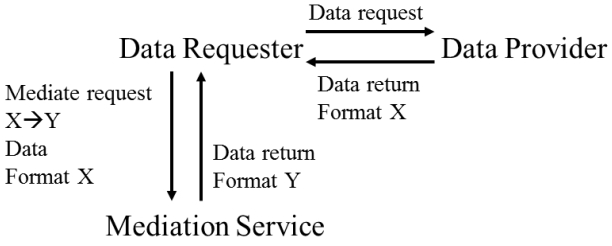

Figure 7: Mediation Service

In this case, there are two connections, both of which meet ELS confidentiality and integrity requirements. However, the mediation service still must be trusted to handle the data, so the confidentiality and integrity of the received

data cannot be guaranteed. This solution is slightly better than the MITM solution because the requester does receive the original data with integrity from the provider. Although mediation is required before using this data, in many cases a sanity check can be performed on this data or a small sample of this data prior to mediation. If independent mediation services are available, it may be possible to provide stronger guarantees on integrity by comparing the output of all such mediation services. However, this is available only at the cost of further reduced confidentiality. Only the requester needs to trust the mediation service for this model to work. The provider implicitly must trust the mediation service because it is possible that any requester will call the mediation service. However, this is rolled into the existing trust of the data requester to properly handle data.

The data to be transmitted in this solution is the data from provider to requester in its original form, this same data from requester to mediation service, and the data from mediation service to requester in the desired form. This approach requires three data transmissions, an increase of one data transmission over the MITM model. Computation is again performed entirely at the mediation service.

C. Mediation Tool Service

A third potential solution is a mediation service, shown in Figure 8, that does not convert data but instead provides a tool to do the conversion, in the form of code that runs on the requester's machine. The requester requests and receives the original data from the provider. It then requests a mediation tool from the mediation tool server to convert it to the desired format. After receiving the tool, the requester performs the conversion using the tool.

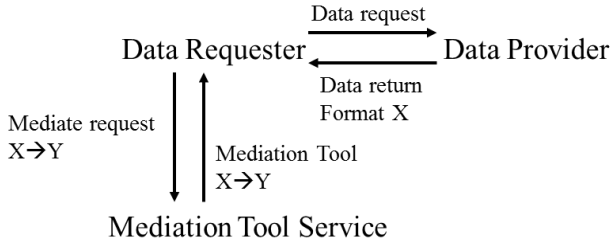

Figure 8: Mediation Tool Service

Again, there are two connections, both of which meet ELS confidentiality and integrity requirements. In addition, because no data is transmitted to the mediation service, the confidentiality of the data is preserved. The integrity still suffers from the problem of traceability through the conversion process. However, because the algorithm is run locally, this provides higher assurance than relying on the mediation service to both provide and run the conversion. Now the requester must only trust that the conversion tool is correct. Again, comparison of results with other tools can provide some assurance that the data is converted properly. Also, comparison of a hash of the tool can provide assurance that the tool received matches with a known-good tool as certified by the enterprise. These still do not provide full data integrity, but this is marginally better than what is possible with the mediation service approach. The trust that is required for this solution to work is the requester's trust of the mediation tool service.

The data transmissions required involve only the single transmission of the original data from provider to requester. However, a new transmission is now required of the mediation tool from the mediation tool service to the requester. Depending on the size of the tool compared to the data set, this may be more or less than the two data transmissions of the MITM solution or the three data transmissions of the mediation service solution. Computation has now shifted from the mediation service to the requester.

A possible performance optimization for computation, especially for mobile or other computationally limited devices, would be to use third-party resources to do the computation, such as cloud servers. However, this introduces an additional trust relationship and additional data transmissions. To preserve security and minimize trust relationships, it is best to keep the computation on the requester's device, depite possible performance issues.

D. Homomorphic Encryption MITM

A fourth solution uses homomorphic encryption, in which data is encrypted such that the encrypted data can be manipulated to perform meaningful computations on the data when decrypted. In this approach, the sender uses homomorphic encryption on the transmitted data and the mediation service mediates the encrypted data. The mediation service can either be a MITM or called directly by the receiver. In either case, it does not know the original decrypted data – it just needs to perform the homomorphically translated mediation function on the homomorphically encrypted data. This allows more flexibility in how the mediation service is implemented and preserves ELS properties.

For homomorphic encryption we examine both the MITM and mediation service architectures. The MITM version is shown in Figure 9.

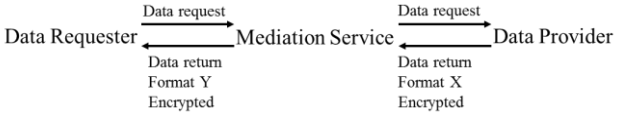

Figure 9: Homomorphic Encryption

With a MITM mediation service, there are two connections. These have the same properties at the connection level as the standard MITM mediation setup. However, the mediation service can no longer view unencrypted content because it operates only on encrypted data. For this reason, confidentiality of the data is preserved between requester and provider. Integrity, however, is not preserved because the mediation service modifies the content. This invalidates integrity protection on the original data, and any new integrity measures are only from the MITM, not the original source of the data. Like the normal MITM mediation approach, both requester

and provider still must trust the mediation service to properly transform the data.

With a separate mediation service using homomorphic encryption, again confidentiality is preserved, and again integrity is similar to that of the normal mediation service approach, in which the requester must trust the mediation service to properly translate.

One new security issue with homomorphic encryption is that some data may leak through the homomorphic encryption schemes, since they must preserve certain properties like sums, products, or ordering of values. An evaluation must be done to determine whether the security properties of the homomorphic encryption are sufficient for protecting the data to be encrypted.

The data transmissions involved are the same as those for normal MITM mediation and mediation service approaches. For the MITM, the data must be transmitted from provider to mediation service and from mediation service to requester. For the mediation service, the data must be transmitted from provider to requester and from requester to mediation service, and then the mediated data must be sent back from the mediation service to the requester.

The computation requirements get more complicated. The mediation computations are done at the mediation service. However, these are now homomorphic encrypted computations, which are more expensive than normal computation. This imposes an additional burden on the mediation service to perform the mediation computation. The encryption of requests and decryption of received results also imposes a potentially large burden on the requester. A possible alternative is the use of partial homomorphic encryption (PHE), which allows limited operations to be performed on data but is potentially much faster. For computations that can be implemented through PHE, this is a potential solution to the performance problems of full homomorphic encryption. PHE holds promise for many standard database operations, but it is not possible on more complicated transformations like arbitrary mathematical or logical expressions [9].

E. Comparison of Solutions

Table 1 shows a comparison of the different approaches. Confidentiality and integrity generally increase going down the list of methods. In particular, the last three options provide end-to-end data confidentiality while the first two options do not. No solution provides end-to-end data integrity, mainly because the conversion algorithm is treated as a black box, which does not allow traceability of integrity through the conversion process. The mediation tool service allows the possibility of limited integrity checks by examining the actual mediation tool code, but in general, it is difficult to analyze code in this way.

The two MITM-based solutions require both requester and provider to explicitly trust the mediation service, while the mediation service and mediation tool service approaches require only the requester to trust the mediation service. This makes the non-MITM solutions easier to adopt, since only the entity requesting the data must trust the mediation service.

Table 1 Comparison of Mediation Methods

Mediation Method	Connection Confidentiality	Connection Integrity	Data E2E Confidentiality	Data E2E Integrity	Needed Trust Relationships*	Data transfers	Algorithm Transfers	Computation at Mediation Svc	Computation at Endpoint
MITM Mediation	N	N	N	N	2	2	0	1	0
Mediation Service	Y	Y	N	N	1	3	0	1	0
Mediation Tool Service	Y	Y	Y	N	1	1	1	0	1
Homomorphic Encryption (MITM)	Y	Y	Y	N	2	2	0	>1	>0
Homomorphic Encryption (service)	Y	Y	Y	N	1	3	0	>1	>0
*A trust relationship means that the sender or receiver needs to trust someone else with the transmitted data in order for this method to work.									

For performance, the different approaches use from one to three data transmissions. The mediation tool service approach has the distinction of requiring the least data transmissions, an additional algorithm transfer, and computation on the requester endpoint instead of the mediation service.

5. CHOOSING A SOLUTION

Based on the analysis above, the mediation tool service and homomorphic encryption approaches are superior for security because they preserve end-to-end data confidentiality, while the MITM and mediation service approaches do not. These approaches may require additional compute or network resources, but the exact resource implications depend on the data and transformation to be performed.

For large data sets with simple transformation algorithms, the mediation tool service approach is well suited because the algorithm transfer will be fast and the number of data transfers is minimized. The relatively simple transforms can also be easily handled by the endpoint itself. For smaller data sets or more complicated transformations, the first two approaches offer potentially improved performance becasue the mediation service can perform the computationally intense transformations, and the extra data transfers incur only a small increase in network utilization due to the small data set size. However, this must be balanced with the security risks of lost confidentiality and lost integrity.

For ELS systems, only the mediation tool service and homomorphic encryption approaches are viable for the general end-to-end mediation problem. However, under certain circumstances the data may be sufficiently generic

that its release is not a problem, or the mediation service may be a trusted party in the transaction. In these cases, the MITM mediation and mediation service approaches conform to ELS because the mediation service is part of the transaction and not just an external party in a two-entity communication.

Ease of implementation of these approaches is an important consideration when building a system. The MITM and mediation service approaches are simple to set up and use. The MITM simply acts as a source for data sets, where the requester indicates the data and format desired. This is the online translation approach. The requester indicates the data to be translated and the source and destination languages, and the translate tool retrieves the data and presents the translation to the requester. The mediation service is similar – the user uploads the data and requests a transformation. Many online base 64 converters use this approach, as do many other file format conversion sites. The requester uploads the data to the site and receives the transformation as a response.

In the MITM case, the sites often contain public data because the MITM must be able to access it to do the transformation, so there is no security concern. For the mediation service, the data sent to the service may be sensitive, so caution is needed in using such sites. In an enterprise, policy may enable a mediation service to access all data in the enterprise, which would expand the scope of a MITM approach but also require access management at the MITM mediation service because it could be used as a backdoor to access restricted data if not properly secured.

For the mediation tool service approach, the mediation service must choose a representation of the algorithm in code. A simple Javascript implementation might be appropriate for simpler transformations, while an executable might be better for more complicated file conversions. For security, the mediation tool service should sign all executables so that their integrity and source can be verified. Then these trusted executables can be installed and used in the future instead of downloading them again. Changes and updates can be indicated by a changed hash as provided by the mediation tool service. It is important to choose a tool that is compatible with different types of requesters. For example, an executable that runs on a desktop may not work on a mobile device. However, compatibility may require the use of inefficient languages like Javascript in a browser, so a tradeoff between performance and portability is an important consideration, and multiple tools could be offered to address different requesters' needs.

Homomorphic encryption implementation is currently very slow, so this is not a viable implementation option for most transformations. However, PHE might be viable for simpler transformations, and as technology in both homomorphic and partial homomorphic encryption develops, these may become more mainstream and optimized for performance. The homomorphic encryption option also requires distributing encryption and decryption keys and metadata to requesters and the mediation service in order to perform operations and recover the encrypted data. This is an additional security function that the implementation must address.

6. SUMMARY

Sharing data among different entities in an enterprise often requires mediation. However, these translations are not always available to those who need them, so this raises the issue of how to implement mediation for the enterprise in a secure way. Simple implementation approaches in common use today do not preserve security properties of the ELS architecture such as end-to-end confidentiality. Approaches that preserve ELS properties offer improved security, but they have different implications for performance and ease of implementation.

REFERENCES

[1] DoDI 8520.2, Public Key Infrastructure (PKI) and Public Key (PK) Enabling, 24 May 2011.

[2] Public Key Cryptography Standard, PKCS #1 v2.2: RSA Cryptography Standard, RSA Laboratories, October 27, 2012.

[3] RFC 5246 The Transport Layer Security (TLS) Protocol Version 1.2, 2008-08.

[4] N. Ragouzis et al., Security Assertion Markup Language (SAML) V2.0 Technical Overview, OASIS Committee Draft, March 2008.

[5] William R. Simpson and Coimbatore Chandersekaran, CCCT2010, Volume II, pp. 84–89, "An Agent Based Monitoring System for Web Services," Orlando, FL, Apr 2011.

[6] William R. Simpson and Coimbatore Chandersekaran, 1st International Conference on Design, User Experience, and Usability, part of the 14th International Conference on Human-Computer Interaction (HCII 2011), "A Multi-Tiered Approach to Enterprise Support Services," 10 pp. Orlando, FL, July 2011. Also published in: A. Marcus (Ed.): Design, User Experience, and Usability, Pt I, HCII 2011, LNCS 6769, pp. 388–397, © Springer-Verlag Berlin Heidelberg 2011.

[7] Frank Konieczny, Eric Trias and Nevin Taylor, "SEADE: Countering the Futility of Network Security," Air and Space Power Journal, Sep–Oct 2015, Vol 29, No.5, p. 4.

[8] Simpson, William R., CRC Press, "Enterprise Level Security – Securing Information Systems in an Uncertain World", by Auerbach Publications, ISBN 9781498764452, May 2016, 397 pp.

[9] Virgil Gligor, "Homomorphic Computations in Secure System Design," Final Report Carnegie Mellon University, Pittsburgh, PA 15213, July 10, 2014.