Secure Enterprise Mobile Ad-hoc Networks

William R Simpson, *Member IAENG* and Kevin E. Foltz

Abstract — **Threat intrusions to enterprise computing systems have led to a formulation of guarded enterprise systems. The approach was to build an impenetrable fortress to prevent hostile entities from entering the enterprise domain. However, this defense and its many reinforcements have repeatedly been found inadequate. The current complexity level has made the fortress approach to security, which is implemented throughout the defense, banking, and other high-trust industries unworkable. An alternative security approach, called Enterprise Level Security (ELS), is the result of a concentrated multi-year program of pilots and research. The primary identity credential for ELS is the Public Key Infrastructure (PKI) certificate, issued to the individual who is provided with a Personal Identity Verification (PIV) card with a hardware chip for storing the private key. All sessions are preceded by a PKI mutual authentication (secondary authentication may be employed when necessary) within Transport Layer Security (TLS) 1.2, and a secure communication pipeline is established. This process was deemed to provide a high enough identity assurance to proceed. However, mobile ad-hoc networking allows entities to dynamically connect and reconfigure connections to make use of available networking resources in a changing environment. These networks range from tiny sensors setting up communications based on a random or unknown configuration to aircraft communicating with each other, the ground, and satellites. Scenarios have differing requirements in terms of setup, reconfiguration, power, speed, and range. This paper presents an adaptation of the ELS principles to the mobile ad-hoc scenario.**

Index Terms — **Enterprise Level Security, Field Connectivity, Mobile Ad-hoc, Mobil Nexus, Networking, Service Requirements, Sub Networks**

I. INTRODUCTION

Mobile ad-hoc implementations are a derivative of normal network approaches. Additionally, they are required to meet the basic security architecture. Each of these will be reviewed before discussing mobile ad-hoc services.

A. Network Overview

The network consists of many different technologies that are split into different layers. One conceptual model for this layering is the Open Systems Interconnect (OSI) [1] seven-layer model shown in Figure 1.

The network layer must be considered to allow upper level layers (including transport session and application layers to conform to the security model) in this document. This layer provides addresses that are unique within a network, allowing communication through internet protocol (IP) routers to any other node that is connected to the same network. The use of bridges and network address translation (NAT) allows different networks with overlapping IP addresses to communicate with each other. However, this often relies on the use of transmission control protocol (TCP) port numbers to distinguish endpoints when traversing network boundaries. The IP layer can use IPv4 or IPv6. Each includes a version of IP security (IPSec) that allows authenticated and encrypted communication between devices.

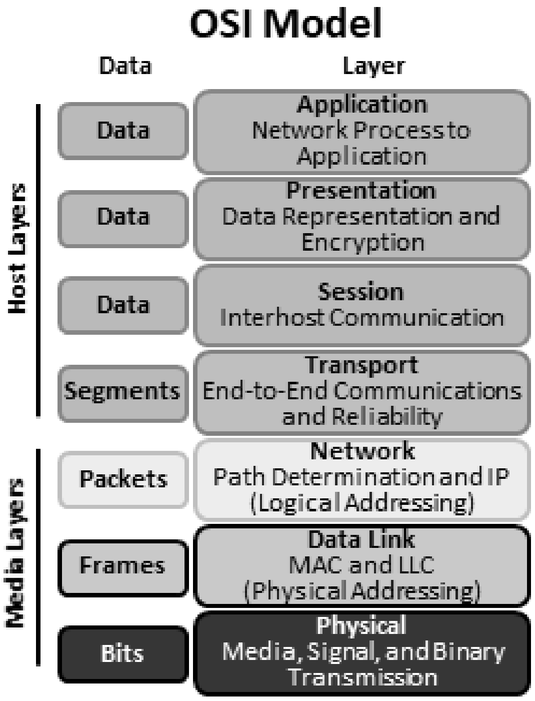
OSI Model
Data
Layer
Host Layers
Data
Application
Network Process to Application
Data
Presentation
Data Representation and Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Communications and Reliability
Media Layers
Packets
Network
Path Determination and IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Physical Addressing)
Bits
Physical
Media, Signal, and Binary Transmission

Fig. 1. The OSI layer model is the most commonly used communication model.

Below the network layer is the data link layer, that connects one device to another. This layer has two sub-layers: the logical link control (LLC) and media access control (MAC). The LLC is the higher sublayer, focusing on multiplexing, while the MAC layer handles addressing and channel access control.

The MAC address is unique to different hardware instances on a subnetwork, allowing unambiguous point-to-point local communication. This can be wired (Ethernet) or wireless, (Wi-Fi). It can be point-to-point using a wire from one machine to another, or broadcast using Ethernet or Wi-Fi. Wi-Fi provides security through various protocols, such as Wired Equivalent Privacy (WEP) or Wi-Fi Protected Access (WPA) [2].

Manuscript received 1 March 2019; revised 22 March 2019. This work was supported in part by the U.S. Secretary of the Air Force and The Institute for Defense Analyses (IDA). The publication of this paper does not indicate endorsement by any organization in the Department of Defense or IDA, nor should the contents be construed as reflecting the official position of these organizations

Kevin E. Foltz is with the Institute for Defense Analyses. (email: kfoltz@ida.org).
William R. Simpson is with the Institute for Defense Analyses, 4850 Mark Center Drive, Alexandria, Virginia 22311 USA and is the corresponding author phone: 703-845-6637, FAX: 703-845-6848 (e-mail: rsimpson@ida.org).

Ethernet and Wi-Fi include not just data link layer protocols, but also specifications for the underlying physical properties of the waveforms and the structure of signals. These can provide some security through frequency-hopping patterns, beam-forming, or other physical layer techniques.

In some cases, such as Link 16 [3], multiple layers are integrated into a single protocol. This facilitates communication between layers. It reduces modularity and portability, but it can allow functions like basing higher-layer coding rates and transmission windows on physical layer signal-to-noise ratios. This could distinguish network congestion from jamming and initiate appropriate responses.

B. Mobile Ad-Hoc Networking

Mobile ad-hoc networking includes a broad range of possible implementations. These implementations range from unstructured networks like specific mobile ad-hoc networks (MANET)s [4], where there is no existing infrastructure and nodes must dynamically configure themselves into a functioning network, to situations in which a mobile node connects to existing infrastructure. This document focuses on situations in which nodes come in and out of communication range of fixed infrastructure and situations in which nodes dynamically connect and disconnect to each other and different networks.

These situations allow many of the higher-layer functional and security protocols to function properly. The following sections describe different aspects of the networking infrastructure that together support the concept of ad-hoc connections and mobility. Figure 2 illustrates those network types.

ELS is a capability designed to counter adversarial threats by protecting applications and data with a dynamic claims-based access control (CBAC) solution. ELS helps provide a high assurance environment in which information can be generated, exchanged, processed, and used. The ELS design is based on a set of high-level tenets that are the overarching guidance for every decision made, from protocol selection to product configuration and use [5].

This article is based in part on a paper published by WCECS 2018 [6].

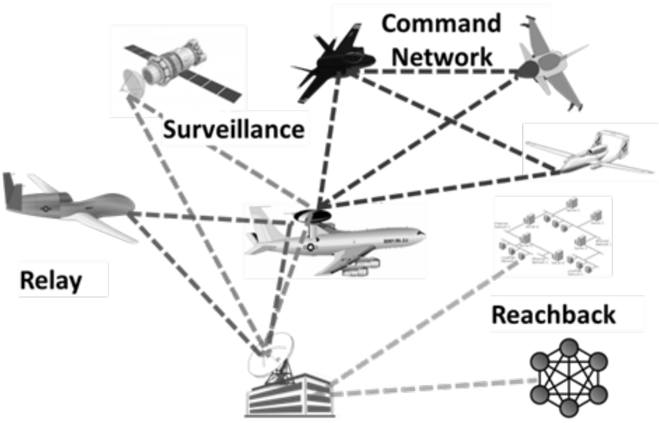

Fig. 2. Ad-hoc networking model used by ELS.

II. ENTERPRISE LEVEL SECURITY

A. Security Process Background

ELS is a capability designed to counter adversarial threats by protecting applications and data with a dynamic CBAC solution. ELS helps provide a high assurance environment in which information can be generated, exchanged, processed, and used. The ELS design is based on a set of high level tenets that are the overarching guidance for every decision made, from protocol selection to product configuration and use [7]. From there, a set of enterprise level requirements are formulated that conforms to the tenets and any high level guidance, policies, and requirements.

B. Design Principles

The basic tenets, used at the outset of the ELS security model [8] are as follows:

0. The ***zeroth*** tenet is that the *malicious entities are present* and can look at network traffic and may attempt to modify that traffic by sending malicious content to network assets. Current threat evaluation indicates that attacks are often successful at all levels; discovering these attacks and their consequences is problematic. In many cases, attackers may compromise and infiltrate before a vulnerability can be mitigated by software changes (patches).

1. The ***first*** tenet is *simplicity*. Added features come at the cost of greater complexity, less understandability, greater difficulty in administration, higher cost, and/or lower adoption rates that may be unacceptable to the organization.

2. The ***second*** tenet, and closely related to the first, is *extensibility*. Any construct we put in place for an enclave should be extensible to the domain, to the enterprise, and ultimately to cross-enterprise and coalition.

3. The ***third*** tenet is *information hiding*. Essentially, information hiding involves only revealing the minimum set of information to the requester and the outside world for making effective, authorized use of a capability.

4. The ***fourth*** tenet is *accountability*. In this context, accountability means being able to unambiguously identify and track what active entity in the enterprise performed any particular operation (e.g., accessed a file or IP address, invoked a service). Active entities include people, machines, and software process, all of which are named, registered and credentialed. By accountability, we mean attribution with supporting evidence.

5. This ***fifth*** tenet is *minimal detail* (to only add detail to the solution to the required level). This combines the principles of simplicity and information hiding and preserves flexibility of implementation at lower levels.

6. The ***sixth*** is the emphasis on a *service-driven* (rather than a product-driven) solution whenever possible. Services should be separated as stated in the separation of function tenant. This also allows simplification and information hiding.

7. The ***seventh*** tenet is that *lines of authority* should be preserved, and information assurance decisions should be made by policy and/or agreement at the appropriate level. One example is that data owners should implement sharing requirements even when the requirements come from a "higher authority."

8. The ***eighth*** tenet is *need-to-share* as overriding the need-to-know. Often effective health, defense, and finance

rely upon and are ineffective without shared information. Shared does not mean released, and the differences must be clear. However, judicious use of release authority and delegated access lead to a broader distribution of information. This leads to a more formalized delegation policy both within and outside of the enterprise.

9. The ***ninth*** tenet is *separation of function*. This makes for fewer interfaces, easier updates, maintenance of least privilege, reduced and easier identified vulnerabilities and aids in forensics.

10.The ***tenth*** tenet is *reliability*; security needs to work even if adversaries know how the process works. In setting up a large scale enterprise, we need to publish exactly how things work. Personnel, computer operations people, and vendors need to know how the system works, and this should not create additional vulnerabilities.

11.The ***eleventh*** tenet is to *trust but verify* (and validate). Trust should be given out sparingly, and even then trusted outputs need checking. Verification includes checking signature blocks, checking that the credential identities match (binding), checking the time stamps, checking to whom information is sent, checking that information received is identical to information sent. Validation includes checking issuing authority, checking certificate validity, and checking identity white lists and black lists.

12.The ***twelfth*** tenet is *minimum attack surface*. Fewer interfaces and minimized functionality in those interfaces decreases the exposure to threats.

13.The ***thirteenth*** tenet is *handle exceptions* and errors. Exception handling involves three basic aspects: logging the exception and restorative actions taken, alerting the Enterprise Support Desk (ESD) to all security related events, and notifying the user of the exception and restorative actions.

14.The ***fourteenth*** tenet is to *use proven solutions*. A carefully developed program of pilots and proofs of concepts must be pursued before elements are integrated into ELS. It is our intention to follow that process even when expediency dictates a quicker solution. Immediate implementation should always be accompanied by a roadmap for integration that includes this tenet.

15.The ***fifteenth*** tenet is *do not repeat old mistakes*. From a software point of view, this has many implications. First, never field a software solution with known vulnerabilities and exploits. There are several organizations that track the known vulnerabilities and exploits, and an analysis against those indexes should be required of all software. Second, a flaw remediation system is required. After a vulnerability analysis, fixes may be required, and after fielding, fixes will be required as new vulnerabilities and exploits are discovered. Third, from an operations standpoint, take time to patch and repair, including outputs from the flaw remediation and improvements in the Security Technical Implementation Guidelines.

Current paper-laden access control processes for an enterprise operation are plagued with ineffectiveness and inefficiencies. Given that, in a number of enterprises tens of thousands of personnel transfer locations and duties annually, delays and security vulnerabilities are introduced daily into their operations. ELS mitigates security risks while eliminating much of the system administration required to manually grant and remove user/group permissions to specific applications/systems. For government and defense early calculations show that ELS-enabled applications saved 90%-95% of recurring man-hours and eliminated up to 3 weeks of delay for access request processing [9]. While the perimeter-based architecture assumes that threats are stopped at the front gates, ELS does not accept this precondition and is designed to mitigate many of the primary vulnerability points at the application using a distributed security architecture shown in Figure 3.

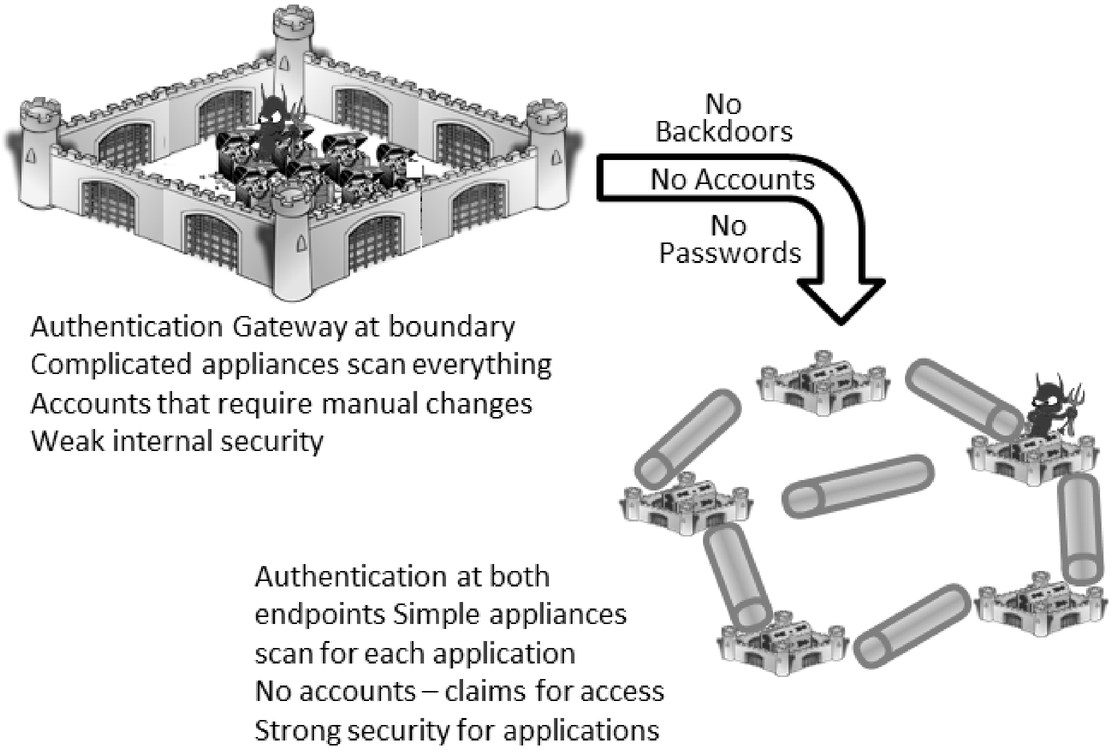
No
Backdoors
No Accounts
No
Passwords
Authentication Gateway at boundary
Complicated appliances scan everything
Accounts that require manual changes
Weak internal security
Authentication at both
endpoints Simple appliances
scan for each application
No accounts – claims for access
Strong security for applications

Fig. 3. The fortress approach has been replaced by a distributed security architecture.

C. Key Concepts

The key concepts for ELS are based on the tenets, as listed in braces, and address more specific architectural decisions than the core tenets [8]. These directly relate to requirements as well, forming a bridge between the high level tenets and the technical requirements.

1. **ELS specific concepts**
   - PKI credentials for active entities. {4, 5, 6, 7}
   - SAML for claims credential. {8}
   - TLS v1.2 for end-to-end confidentiality, integrity, and authentication.{9}
   - STS as trusted entity for generating authorization credentials.
   - Exceptions shall be documented with a plan and schedule to become compliant.
2. **Naming**. A standard naming convention and process is applied to active entities. {2, 4, 11}
3. **AuthN Claims**. Authentication is implemented by a verifiable identity claims-based process. {0, 2, 4, 11}
4. **AuthN Vetting**. Identity claims are tied to a strong vetting process. {0, 4, 11}
5. **AuthN Verify IDs**. Active entities verify each other's identity. {0, 4, 11}
6. **AuthN Private Key**. Identity is verified by proof of ownership of the private key associated with an identity claim. {4}
7. **AuthN Own Behalf.** Active entities act on their own behalf. {0, 1, 12}
8. **AuthZ Data Owner Rqt**. The claims objective requirement is provided by the data owner. {7, 8}
9. **AuthZ Access and Privilege**. Service providers use identity and authorization credential claims to determine access and privilege. {0, 1, 2, 3, 8, 11, 13}
10. **AuthZ Claims Service**. A trusted entity examines the attributes of an entity and determines if the claims objective requirement is satisfied. {2, 3, 5, 6, 9}
11. **AuthZ Claims Imply Access**. A claim in an authorization credential is a statement that an access requirement has been satisfied. {1, 3, 5, 8, 11}
12. T **AuthZ Claims**. Authorization is implemented by a verifiable identity, access, and privilege claims-based process. {0, 2, 3, 4, 8, 11}
13. **AuthZ Additional Info**. As part of the requirement definition, the data owner may request additional information about the requesting entity. {1, 2, 11, 12}
14. **AuthZ Credential Creation**. Authorization credentials are created by a trusted entity for a specific requester, a specific target resource, and a specific level of access. {0, 6, 9, 10}
15. **Web Functionally**. Functionality is to be provided through web services. {6}
16. **Non-Custom**. It is undesirable to work a point solution or custom approach. {1, 2, 5, 14}
17. **Delegation**. A formalized delegation policy both within and outside of the enterprise is required. {0, 2, 4, 7, 11}
18. **Trust**. The ability to be verified and validated is a requirement for trusted entities. {0, 4, 11}
19. **Confidentiality**. Active entity interactions require confidentiality of data/content exchanged. {0, 3, 10}
20. **Guarantees**. Integrity, authenticity, timeliness, and pedigree are to be guaranteed. {0, 2, 4, 10, 11}
21. **Monitoring**. Monitoring is a required element of cyber security. {0, 4, 10, 11, 13}
22. **Sanitize**. Eliminate or mitigate malware. {0, 15}

D. Mapping of Tenants and Key Concepts for Requirements

The model and concept tracing leads to some very specific requirements as shown in Figure 4 below. These are a subset of requirements for any service and are not meant to be all of the requirements for ELS; they instead supplement those requirements for each technical area as shown below.

1. Active entities shall be named in accordance with enterprise naming standard.
2. Active entities within the enterprise shall have unique identities.
3. Active entities shall use credentials from approved certificate issuing authorities.
4. Active entity communication shall use two-way end-to-end PKI authentication.
5. No active entity shall be anonymous.
6. Reusable or third party authentication tokens shall not be used.
7. Traditional single sign-on (forward passing of identity) is not allowed.
8. Private keys shall be stored in tamper proof, threat mitigating storage to which only the associated entity has access.
9. No impersonation of active entities through sharing of private keys or issuing of duplicate credentials.
10. No entity shall act "on behalf of" any other entity. No proxies or portals are allowed, as they cause ambiguity in identity.
11. Active entity authentication uses only primary or derived credentials. Derived credentials must be based upon the primary credential.
12. Any active entity without authorization credential claims shall only access identity-based services.
13. Active entities that act as a provider to any other active entity requester that requires claims shall have objective requirements in the enterprise registry.
14. Active entities that act as a requester of any other active entity that requires claims shall have attributes that support the computation of claims.
15. Each active entity that acts as a provider for any other active entity and requires claims shall have a SAML handler installed.
16. Access and privilege to applications and services shall be provided by SAML-based verifiable claims from a trusted STS.
17. Authorization credentials shall conform to least privilege so that only the relevant claims for the target are included.

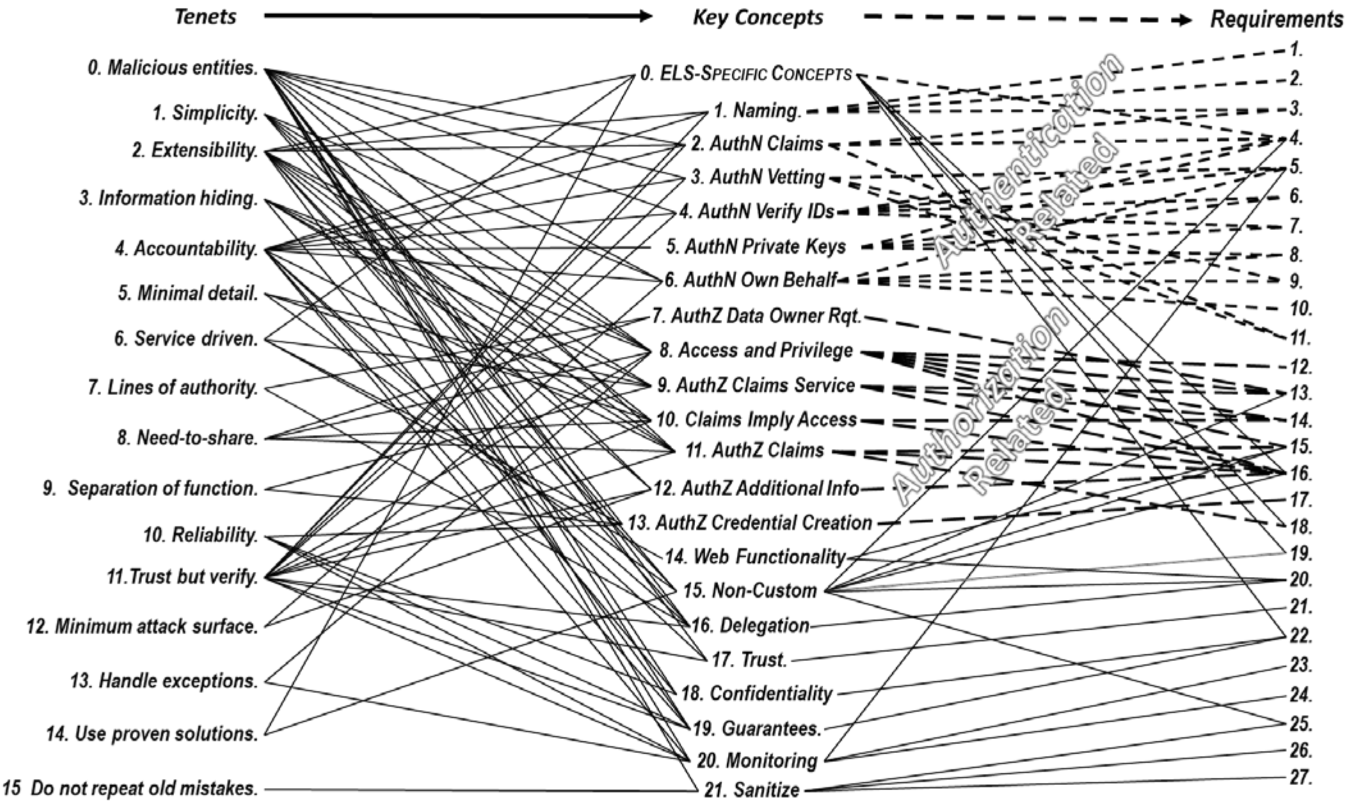

Fig. 4. Requirements are derived from the basic security model.

18. Each active entity that acts as a provider for any other active entity and requires claims shall have a list of trusted providers of claims and a process for verifying and validating that those claims were provided by a trusted provider.
19. Each expedient solution shall have an accountable decision authority and a roadmap for bringing the solution into the basic approach for security.
20. Delegation services shall be used for (and only for) assignment of duties not based on existing attributes and for temporary assignments not met by existing claims and attributes.
21. Active entities that are designated trusted shall have credentials that are verifiable. "Verifiable" is meant to include the means for verifying the currency and integrity of the credential and the source of the credential.
22. Active entity interactions shall occur over end-to-end TLS v1.2 connections.
23. Active entities shall have provisions for logging security relevant events.
24. Monitoring of active entities shall be performed.
25. A carefully developed and executed program of pilots and proofs of concepts shall precede integration into ELS.
26. Active entities shall evaluate inputs for consistency with intended function before acting on any input.
27. Active entities shall evaluate outputs for consistency with intended function before transmitting any output.

E. Security Principles

The ELS design addresses five security principles that are derived from the basic tenets:

- Know the Players: This is done by enforcing bi-lateral end-to-end authentication.
- Maintain Confidentiality: This entails end-to-end unbroken encryption (no in-transit decryption/payload inspection).
- Separate Access and Privilege from Identity: This is done by an authorization credential.
- Maintain Integrity: This is done by ensuring that you received exactly what was sent.
- Require Explicit Accountability – This is accomplished by monitoring and logging transactions.

Know the Players

In ELS, the identity certificate is an X.509 Public Key Infrastructure (PKI) certificate [10]. This identity is required for all active entities, both person and non-person (e.g., services), as shown in Figure 5.

PKI certificates are verified and validated. Ownership is verified by a holder-of-key check. Supplemental authentication factors (in combination with a PKI credential) may be required from certain entities, such as identity confirming information or biometric data. In certain extreme cases, an identity for a person may be built without a PKI credential [11], and a temporary certificate is issued for compatibility. The temporary certificate has a short life allowing for the establishment of a few sessions before it expires.

Figure 6 shows that ELS establishes end-to-end Transport Layer Security (TLS) [12] encryption and never gives away private keys that belong uniquely to the certificate holder. Many security instantiations include passing of private keys and breaking the encryption in order to examine content

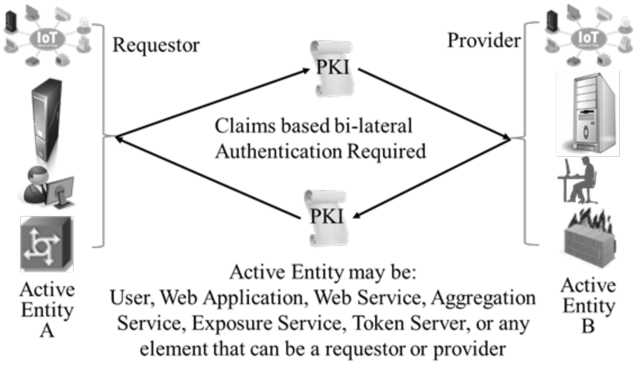
Requestor
Provider
PKI
Claims based bi-lateral
Authentication Required
PKI
IoT
IoT
Active Entity may be:
User, Web Application, Web Service, Aggregation
Service, Exposure Service, Token Server, or any
element that can be a requestor or provider
Active
Entity
A
Active
Entity
B

Fig. 5. Bi-lateral authentication is required

Maintain Confidentiality

. This type of security examination is not allowed in ELS. Although TLS is the preferred implementation care must be taken to avoid known vulnerabilities in implementation algorithms [13]. The cited reference is based upon a common shortcut used to improve performance known as the Chinese Remainder Theorem (CRT) [14-16]

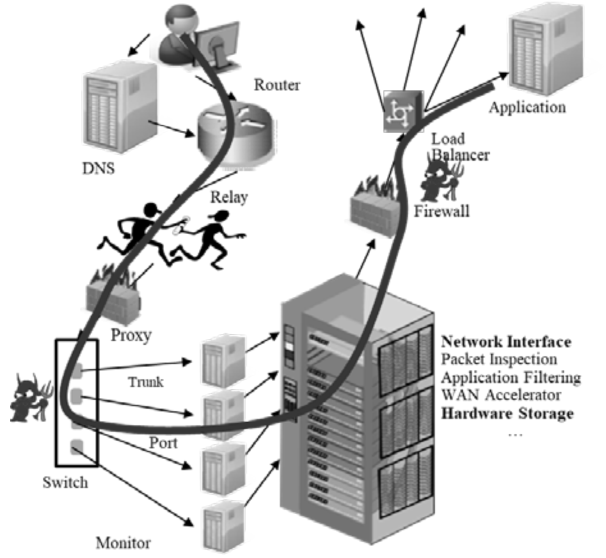

Fig. 6. End-to-end encryption is required.

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, claims are recomputed based on new associated attributes (see section III), allowing immediate access to required mission information. As shown in Figure 7, access control credentials utilize the SAML (SAML authorization tokens differ from the more commonly used single-sign-on (SSO) tokens, and in ELS, are not used for authentication) [17]. SAML tokens are created and signed by an enterprise STS. The signatures are verified and validated before acceptance. The credentials of the signers also are verified and validated. The credential for access and privilege is bound to the requester by ensuring a match of the identity used in both authentication (PKI exchange) and authorization credentials.

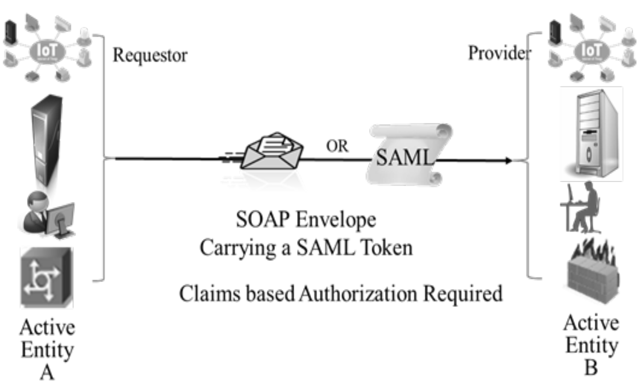

Fig. 7. Claims-based authorization is required.

Maintain Integrity

Integrity is implemented at the connection layer by end-to-end TLS Message Authentication Codes (MACs), see Figure 8. MACs are separately encrypted hashes using a shared secret passed during the crypto exchange. If the TLS crypto were broken, the packet may be modified. However, without the shared secret, the MAC cannot be modified. This provides an extra mitigation against modification. Chained integrity, where trust is passed on transitively from one entity to another, is not used as it is not as strong as employing end-to-end integrity. At the application layer, packages (SAML tokens etc.) are electronically signed, and electronic signatures are verified and validated [18].

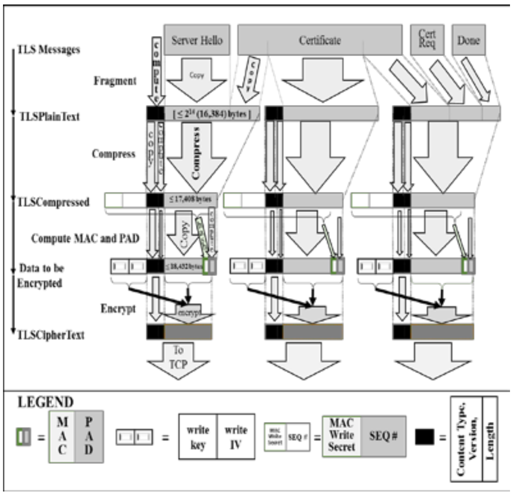

Fig. 8. Integrity measures are required.

Require Explicit Accountability

All active entities with ELS are required to act on their own behalf (no proxies or impersonation allowed). As shown in Figure 9, 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 to an enterprise relational 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. The details of this activity are provided in [19, 20].

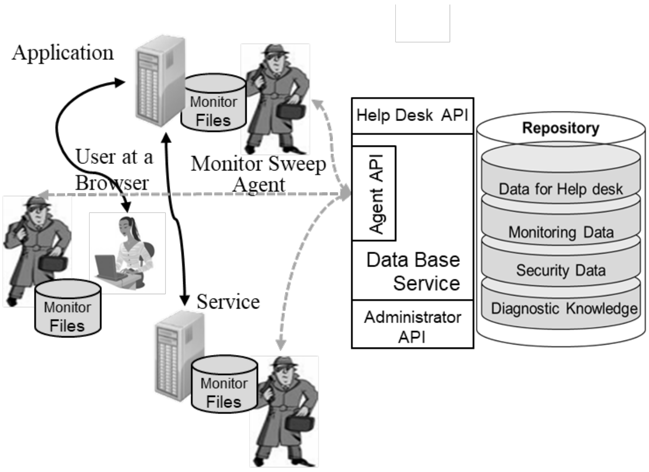

Fig. 9. Accountability is achieved through centralized monitoring.

III. MOBILE AD-HOC NETWORK SERVICES

The services described in this section are shown in Figure 10. These services are automated and seek only operator confirmations when and if required. They reside on each element participating in the networks shown in Figure 10. Each element in Figure 1 must participate in a handshake with the nexus (see section I) that identifies compatible protocols, waveforms, and drivers to establish a connection. These services act as the initial end-points for connection management. The connection is followed by a bi-lateral authentication and secure channel to the end-point device manager service [21]. The end-point device manager service is the entry point for the requester to access domain services. This must be followed by bi-lateral authentication at the device level. Basic services are shown on the left, building from basic hardware capabilities to supported protocols. Mobile ad-hoc network services are on the right, building from hardware and software management to the "Send Data" service that takes data and a destination as an input and sets up appropriate connections and initiates the communication using the supplied data. Arrows indicate dependencies, where arrows point from the service that is used to the service that uses it.

Certain members of the networks are designated as nexus. Nexus points may be located throughout the operational area. In order to be a nexus, the member must either have reachback to the enterprise, or in the case of disconnected, intermittent or limited bandwidth (DIL), it must be provisioned with all of the elements required to do enterprise business, including but not limited to:

- A fully functional Security Token Server (STS).
- A proper subset of the Enterprise Attribute Store (EAS).
- A claims repository that matches the elements of the EAS subset.
- A device management service capability.

Nexus elements seek out and provide a handshake to any other nexus points within range. The chaining of nexus points allows reach back from the local network to the enterprise when one or more of the nexus in the chain can reach a network node.

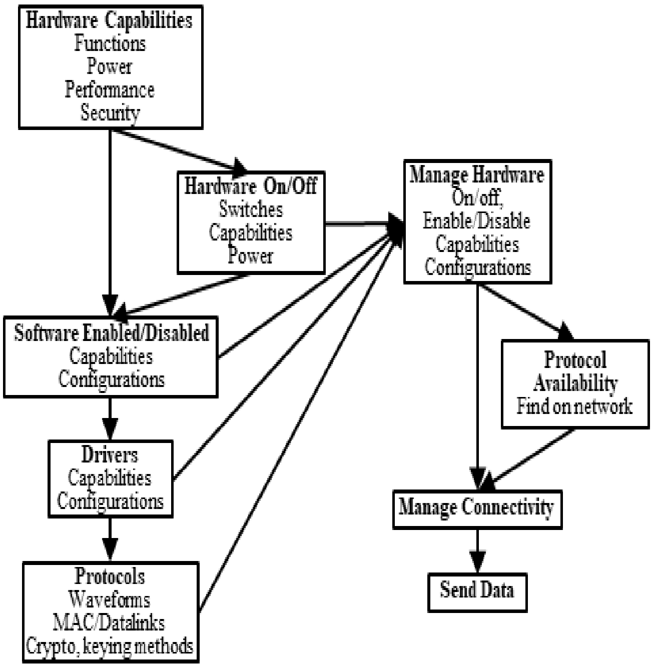

Fig. 10. Mobile ad-hoc services and dependencies for ELS.

The member must have full system capability and acts as the manager of ad-hoc sub networks. An end-point device manager service [21] must reside on a nexus, and a nexus must be part of each network and be the entry point for the requester to access domain capabilities. Designated nexus points is shown in Figure 11.

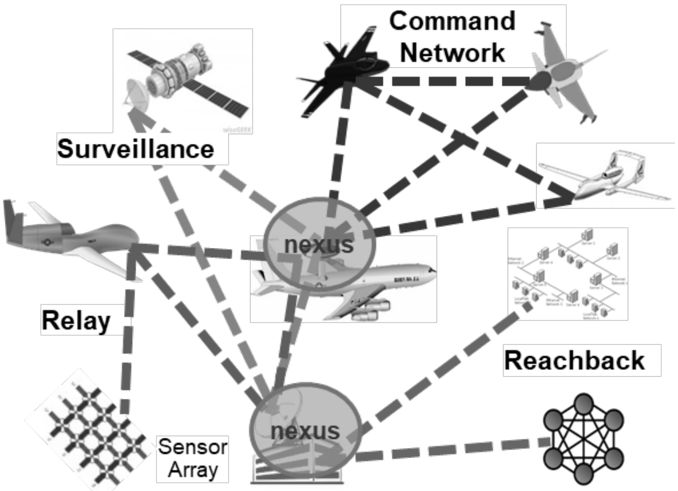

Fig. 11. Designated nexus members area necessity.

IV. NETWORK SERVICE DESCRIPTIONS

Network service descriptions are provided in the following sections.

A. *Detection of Hardware Capabilities*

This section describes the basic services that provide information about the available hardware and the software that directly controls it. These are typically duplicated for each piece of communication hardware in a device so that higher layer services have direct and independent control over each hardware interface. The interfaces to the hardware may be specific to the hardware so that higher layer services provide and use mediation services to interface with these lower-level services.

In order for a node to join a mobile or ad-hoc network, it must know that the network exists. This can be done by continuously polling for available connections or looking for connections when a request is made to connect. Polling involves more ongoing work and power but provides continuous feedback, while on-demand connection uses fewer resources but requires explicit instruction and incurs a delay. To bridge these two methods, a local service can be invoked that periodically polls for connections and provides the latest data to higher-layer services. This provides a configurable method to tradeoff between power and responsiveness across all possible connection types.

Connections at the lowest layer involve the hardware that actually does the signal generation and transmission. This hardware is controlled by drivers or other software that provide an interface to the operating system and local applications and services. The following information is of interest:

- Hardware capabilities that exist for a given device:
    - Capabilities that are supported,
    - Power and other performance that is supported;
- Hardware that is enabled or disabled by physical switches or other hardware mechanisms;
- Hardware that is enabled or disabled by software:
    - Capabilities that are enabled or disabled in the software;
- Hardware that has the appropriate software drivers and other code in place for use:
    - Capabilities that are supported by the drivers or software;
- Protocols that are supported for the hardware:
    - Waveforms,
    - Mandatory Access Control (MAC)/Datalink protocols and versions,
    - Crypto protocols, versions, keying methods.

All of these must be exchanged between ad-hoc participants and the nexus points. The nexus acts as the controller for sub network communications. Each of these translates into local services for mobile and ad-hoc networking. The services described in this section provide basic information about what networking is available, what could be made available, and the capabilities associated with what is and what could be available. In addition, some configuration of the lower layer hardware and software is made available through these services to other services.

The capabilities list for a device describes what hardware is available. This may take different forms. For some devices, it could provide a list of standard hardware regardless of what is currently available, such as standard-issue mass-produced units. Such a service would rely on outside or fixed data sets and not the system itself. Other services describe the hardware interfaces associated with the device. For example, a description of whether Universal Serial Bus (USB) 3.0 is supported or just USB 2.0 would be useful when deciding which hardware device to attach through a USB port. Such services could be offline, static, or based on querying the actual device to determine what is available. Other services describe what hardware is actually connected. Unlike some of the services described above that rely on fixed or external information sources, this service actually queries the system to determine what is connected. For hardware that is found, additional information can sometimes be provided, such as the capabilities of the hardware in terms of speed, power, or supported frequencies.

In some cases, hardware is available but switched off. This service provides information about the current state of such hardware. In some cases, hardware that is switched off is indistinguishable from hardware that is not present, but a distinction is made when possible. This allows a service to inform a user that a physical action must be taken to enable communication.

In addition to hardware switches, there are ways to enable and disable communication hardware through the use of software. This can be through an application, the operating system, registry items, or device driver settings. A service is provided to describe the current state of the communication hardware and allow changing this state as permitted through software. In addition to a simple on/off switch, software can provide detailed capability and configuration information, such as frequencies, versions, protocols, security settings, and many others.

In order to use the communication hardware, appropriate drivers and other software must be available and correctly functioning. This service checks hardware for proper operation and reports the status of the hardware and its drivers. This service may simply examine the driver and perform what amounts to static analysis of the system, or it may actually attempt to use the system and check that it responds appropriately. This service provides not just information about the system, but information about how it is currently operating. This includes whether the device is functioning, as well as which of its capabilities are working, such as transmission speeds, error rates, or power consumption, and potentially how well they are working.

This service provides information about particular protocols that run over different communication hardware. The protocols of interest are the protocols specific to the communication hardware. For example, a Wi-Fi protocol service would provide information about the Wi-Fi protocol, not IP or TCP. This service provides information about which protocols are supported by the hardware and which versions of each of these supported protocols are available. Additional information includes which frequencies, waveforms, data link, or MAC layer protocols are supported, and what type of cryptography or other cryptographic protections are available.

B. *Detection of Network Opportunities*

This service provides the ability to test enabled hardware for its protocol support at the network layer. This goes beyond the protocol-based services discussed in the previous section, which apply to the hardware protocols. It looks, for example, for Dynamic Host Configuration Protocol (DHCP) servers, network gateways, DNS servers, and other services that would be available in the presence of a network. These are the services that will be used for web service and web application requests. It is important to know whether these services are available, and to what extent they are provided. Knowledge about whether the connection is local or connected to other networks provides important information about the type of connection that can be used by other services.

This service includes tests for proxies, gateways, and other forms of network intermediaries. For example, proxies can be detected by accessing known sites and checking the certificate provided through Hypertext Transfer Protocol Secure (HTTPS). If it does not match the known good certificate, then a proxy is in the middle. This informs decisions about which network to use, as networks with proxies make ELS communication impossible by preventing end-to-end authentication through TLS, but they would be acceptable for low-security traffic.

C. Selection of Waveforms and Protocols

This service is used to turn hardware on and off in order to use a specific set of communication hardware. In some cases, this capability can function fully in software using the software interfaces described in the previous section. In cases in which physical action is required, a notification to a human or other interface, such as a machine or robot, is required to initiate the hardware action. In either case, the goal is to have the appropriate hardware on and enabled and everything else off or disabled. This can be for power conservation, stealth, or just a general security practice to reduce unneeded interfaces.

In addition to just turning hardware on and off, this service allows configuration of the hardware, including selection of frequencies, protocol versions, waveforms, and other hardware-level information. This service acts somewhat like a mediation service that provides a standard interface for higher-level protocols to manage the underlying hardware. It translates the hardware and low-level software controls into standard interfaces for the higher layers. This enables a consistent treatment of communication channels and re-use of higher-layer services across the enterprise and different devices within it. This service dynamically maintains a set of connections that provide an optimal allocation of resources to available potential connections based on provided performance metrics. For example, if high-speed connectivity to a particular IP address is desired, the service may continuously poll for available connections and choose the fastest one that has connectivity to the desired endpoint. Other parameters can be weighed against each other as well, such as power consumption, cost, and combinations such as power per bit or power per bit/sec. Additional inputs would be required for this service to operate effectively, including power consumption models, pricing models, and latency and throughput measurements and models.

This service uses the Manage Hardware service to actually make changes to the system and its connectivity. It uses a set of defined metrics, measured and provided information about the available networks and connections, and optimization logic to make decisions about how to invoke Manage Hardware to best provide what is desired.

This service not only determines which protocols are available, as described above, but also performs handshakes and information exchanges to establish IP addresses, secure connections, and other functions that actually initiate protocols for connectivity. Examples include Dynamic Host Configuration Protocol (DHCP) requests, Domain Name System (DNS) queries, and other protocols that are common first steps toward data transfer after initial basic connectivity is established. Any ongoing “ping”-type communication is handled by this service as well to establish and update what protocols are available.

D. Service Discovery

Lower-level service discovery is addressed by the Protocol Availability service, but a separate method must be used for ELS web services. In a connected network the claims query service is used to determine a list of all applications and services to which an ELS requester entity has claims or access through identity. In a DIL mobile ad-hoc environment, this service may not be accessible, but a local copy may be available. If so, this can be used for service discovery. This local copy must be hosted in a canonical place that is accessible to anyone on the network so that it can be used as an initial access point to any other ELS services and applications available in the local environment. Although the claims query service is not part of mobile ad-hoc services (it is part of the ELS suite of services), it is mentioned here for context. For all communication, the Send Data service is used to choose the hardware, protocol, and associated settings to provide the data transmission and receiving of any associated responses.

This service provides network communication based on any request and uses available connections to send and receive data. Software on a device calls this service to perform any network-based communication, and this service handles all network requests, sets up appropriate connections if available, and takes care of sending the requests and receiving the responses. It notifies the end entity making requests of the status of the current connections. It uses the metrics and parameters for performance, cost, and power as input and passes these on to the Manage Connectivity service to allow it to maintain a set of appropriate connections for communication. However, the Send Data service can override these settings based on current requests. For example, if cost and power are a primary concern, most communications will be disabled by Manage Connectivity. However, when a short high-priority message must be sent on a hardware module that is disabled, Send Data can override the default settings and make performance for that communication a priority for the duration required for the communication.

E. Query/Response Capabilities

Like the service discovery described above, query and response capabilities are based on ELS. After mobile ad-hoc services are used to establish connectivity ELS queries can proceed. If network connectivity provides access to the EAS and other network resources, then a standard ELS query can follow. If the local network is isolated and has its own EAS instance, then the local instance can be used to provide ELS-based access to local resources. If the local network is isolated and does not host its own instance of EAS, then access is limited to the non-ELS services provided on the local network. For intermittent connectivity, asynchronous messaging may be offered as a service even if synchronous communication is not, as asynchronous communication can be queued until connectivity returns. As with service discovery, the Send Data service handles the sending and receiving of data over the appropriate connections. The following sections describe the steps in setting up a connection. It is expected that this service will handle all of these either directly or indirectly using the previously mentioned services.

F. Network Broadcast

The first step for a mobile or ad-hoc connection is for the network to identify itself to the mobile node. This is typically done through some sort of network broadcast that identifies the transmitter, the network it represents, its address, the protocols supported, the security offered and required, and other relevant information. For Wi-Fi, for example, a beacon message is sent 100 times per second with this type of information. In some cases, this function is disabled or limited. For Wi-Fi, the Service Set Identifier (SSID) can be hidden so that only nodes that explicitly request the proper ID are allowed to connect. The beacons can be disabled entirely so that the mobile node must know of the network's existence in advance in order to connect. Other techniques exist to either hide connections or make detection and connection more difficult for unauthorized entities. These are more difficult to implement on wireless networks because the communications are broadcast to an entity in the vicinity, making replay attacks possible. In general, security protocols are a more robust method of limiting access than simple message-content, formatting, or timing-based methods. Wi-Fi Protected Access (WPA) for Wi-Fi and IPSec for IP-based network layer communications. For wired networks, security is often minimal, allowing anyone with physical access and connectivity to use available network services. An Ethernet connection usually is initiated automatically when a wire is plugged in to an Ethernet port. Higher-layer services may require further actions for access, but the lower-level connectivity provides little, if any, security.

G. System Discovery

After the network identifies itself, if it chooses to do so, the mobile node must discover what is available and how to connect [22-24]. With current systems, many possible network connections are available, such as satellite, Wi-Fi, Military Link Systems, broadband, and others. The networks provide information about different connections, and the node must make sense of this and discover which networks are accessible, which protocols and options are supported, which security is supported and sufficient to meet policy requirements, and which connections support higher-layer applications. ELS requires bi-lateral authentication, but it may be based on identity for access.

H. Joining a Network

The mobile node, though some internal logic, determines which network to join and initiates a "request to join" handshake [25-26]. This may involve the exchange of identification information, it may include security parameter negotiation, and it may include protocol negotiation. Wi-Fi often includes security information. Link systems use device profiles to set the message formats and protocols. In any case, this is where the connection from the mobile node to the network node is established, along with any required parameters.

As part of the request to join, physical layer attributes may be collected, such as signal strength, noise level, signal quality, multi-path parameters, location information, and supported waveforms and formats. Wi-Fi 802.11n and 802.11ai support beamforming, allowing the multiple antennas at the transmitter and receiver to be used to determine the direction of transmission, which can boost the signal in the vicinity of the communicating entities and reduce it elsewhere. This allows reduced power, slightly increased security, and potentially better use of available network resources by reducing interference with other transmissions.

Other more advanced techniques may allow the use of multipath and complicated urban obstacles to be used to enhance channel security, quality, power efficiency, and data rates. The transmitter sends a test signal to the receiver, which then relays the received signal properties back to the transmitter. The transmitter can then reshape the transmission to "invert" the environmental distortion and allow positive reconstruction of signals at the receiver. Listeners at other physical locations will not be able to properly reconstruct the signal. This allows lower power transmission, better signal to noise, and potentially better privacy against eavesdroppers.

V. Other Considerations

There are several other processes that need to be considered as discussed below.

A. Exchange of Certificates

One important part of the request to join includes the exchange of certificates. The certificates are assigned to devices and allow authentication based on a trusted certificate authority. For ELS, certificates are stored in hardware, such as a Hardware Security Module (HSM) [27] or PIV Card [28]. For lower layer exchanges, the device Trusted Platform Module (TPM) [29] is the preferred location. Each device is equipped with a TPM or TPM-like hardware certificate and key store, which is used to authenticate to the network or to the mobile node when required.

For mobile devices without hardware stores, a derived credential may be used for the certificate exchange. This derived credential is issued by a trusted registration authority (RA) in the enterprise. The derived credential uses the same original certification as the primary credential. If the primary credential is revoked for reasons relating to certification, the derived credential is also revoked, as its certification is no longer secure. If the primary credential is revoked due to issues specific to the credential instance, then the derived credential may remain valid independently. Revocation of the derived credential similar may or may not lead to revocation of the primary credential, based on the reasons for revocation.

B. Device Requirements

Devices allowed to join enterprise networks are registered and managed by the enterprise use restrictions. All devices have a PKI certificate (certificate authority (CA) issued PKI or derived) in hardware storage (preferably in a TPM). The device and the domain controller perform bi-lateral PKI-based mutual authentication before establishment of the channel to the end-point device manager service. The device may also contain one or more individual user certificates (CA issued PKI or derived) that are activated when the user signs on to the device. The device may be required to register with the enterprise domain and report attestation from the TPM and other data such as location (where appropriate).

After joining the network and properly authenticating, it may be desirable to set up an end-point device manager service connection to a remote network. This provides an IP-layer secure tunnel through which higher-layer data can be sent. The initial network connection only applies to the link layer or device-to-device connection.

The end-point device manager connection uses machine certificates to authenticate the mobile node to the end-point device manager server and the end-point device manager server to the mobile node. The end-point device manager server then makes internal network services available to the mobile node. Particular attention must be paid to which nodes are allowed to connect to the end-point device manager server. The devices must have controls, through mobile device management or some other verifiable machine hardware and software integrity checks that ensure that the device is protected from compromise to a level comparable to that of the internal nodes on the network.

C. Discovery of Services

After connecting through the end-point device manager or just to the local network, service discovery can begin. This starts the use of higher-layer protocols, which talk using various protocols over TCP, UDP, or other transport layer protocols. All active entities must have a credential to initiate a request (derived credentials for entities residing on mobile platforms are permitted). For example, the requester may use a known URL, such as the EAS Claims Query service to retrieve a list of available services. These services are provided based on the requesting entity's identity, as provided in a CAC, a PIV card, an NPE certificate, or derived credential, HSM, or other certificate or key store.

Service discovery [22-25] can be initiated locally for DIL environments with a local cache of the claims repository and EAS Claims Query service. The claims query service may be modified to provide identity-based access-only claims. For mobile devices that are provided network connectivity to the primary EAS instance, no cache is required and a normal request is sent. Discovery may be accomplished initially using a Claims Query service. The initial handshake is bi-lateral PKI mutual authentication. This service is identity-based and returns links to claims for service that the requester has. The requester must know the local Uniform Resource Locator (URL) for that service.

D. Request for Service

When access to the EAS is established, the request for service can be sent to the desired application or service or a link in the Claims Query Service return page may be executed. The EAS-provided link redirects to an STS, which provides authorization information in a SAML and then redirects back to the service. The service's ELS handler processes the request and allows access.

Mobile and ad-hoc networking requires some level of performance to support higher-layer protocols and applications [30]. In some cases, poor wireless links or intermittent connectivity prevent the networking protocols from functioning well enough to support the higher-layer protocols. In other cases, the implementation of the protocols is inefficient, uses improper configuration, or adds extra components that reduce performance, such as monitoring or filtering. Those factors under the control of the implementer must combine with those not under control to provide a level of service that supports higher-level protocols and applications appropriate for the network and network participants

SUMMARY

We have reviewed the mobile ad-hoc issues in a high assurance security system. We have also described an approach that relies on high-assurance architectures and the protection elements they provide through PKI. The basic approach becomes compromised when identity is not verified by a strong credential for unique identification (such as holder-of-key in a PKI or a credential derived from that credential). The PKI usage is so fundamental to this approach that we have provided non-certificated users a way to obtain a temporary PKI certificate based upon their enterprise need and the level of identity assurance needed to provide access and privilege to applications [31]. The process is fully compatible with ELS and works as a complement to existing infrastructure. This work is part of a body of work for high-assurance enterprise computing using web services. Elements of this work are described in [32-52]. This work has raised a number of issues as well as identifying primary capabilities. First among these are the number and types of hardware and protocols that will be supported. Work has begun on the layer 7 services necessary to implement an ad-hoc networking capability while maintaining the high level of security in ELS. No firm date for implementation has been established, but a target date for capabilities demonstration is in the 2020-2021 time frame as shown in figures 12 and 13.

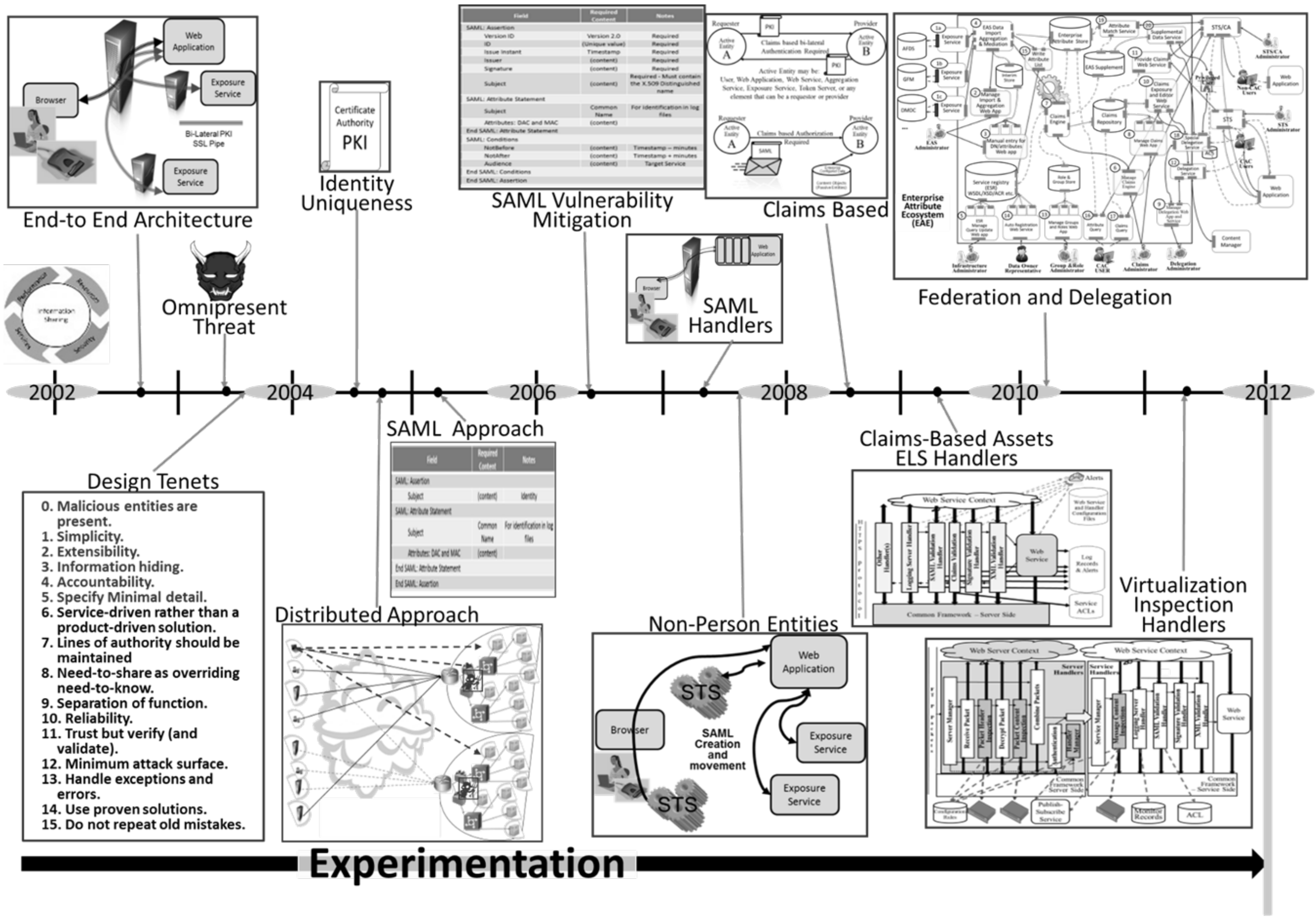
End-to End Architecture
Web Application
Browser
Exposure Service
Bi-Lateral PKI SSL Pipe
Exposure Service
Certificate Authority
PKI
Identity Uniqueness
SAML Vulnerability Mitigation
Claims Based
Federation and Delegation
Enterprise Attribute Ecosystem (EAE)
Omnipresent Threat
SAML Handlers
2002
2004
2006
2008
2010
2012
SAML Approach
Design Tenets
0. Malicious entities are present.
1. Simplicity.
2. Extensibility.
3. Information hiding.
4. Accountability.
5. Specify Minimal detail.
6. Service-driven rather than a product-driven solution.
7. Lines of authority should be maintained
8. Need-to-share as overriding need-to-know.
9. Separation of function.
10. Reliability.
11. Trust but verify (and validate).
12. Minimum attack surface.
13. Handle exceptions and errors.
14. Use proven solutions.
15. Do not repeat old mistakes.
Distributed Approach
Claims-Based Assets ELS Handlers
Web Service Context
Common Framework – Server Side
Non-Person Entities
Web Application
STS
Browser
SAML Creation and movement
Exposure Service
Exposure Service
STS
Virtualization Inspection Handlers
Web Server Context
Web Service Context
Experimentation

Fig. 12. ELS early experimentation timeline ran for ten years.

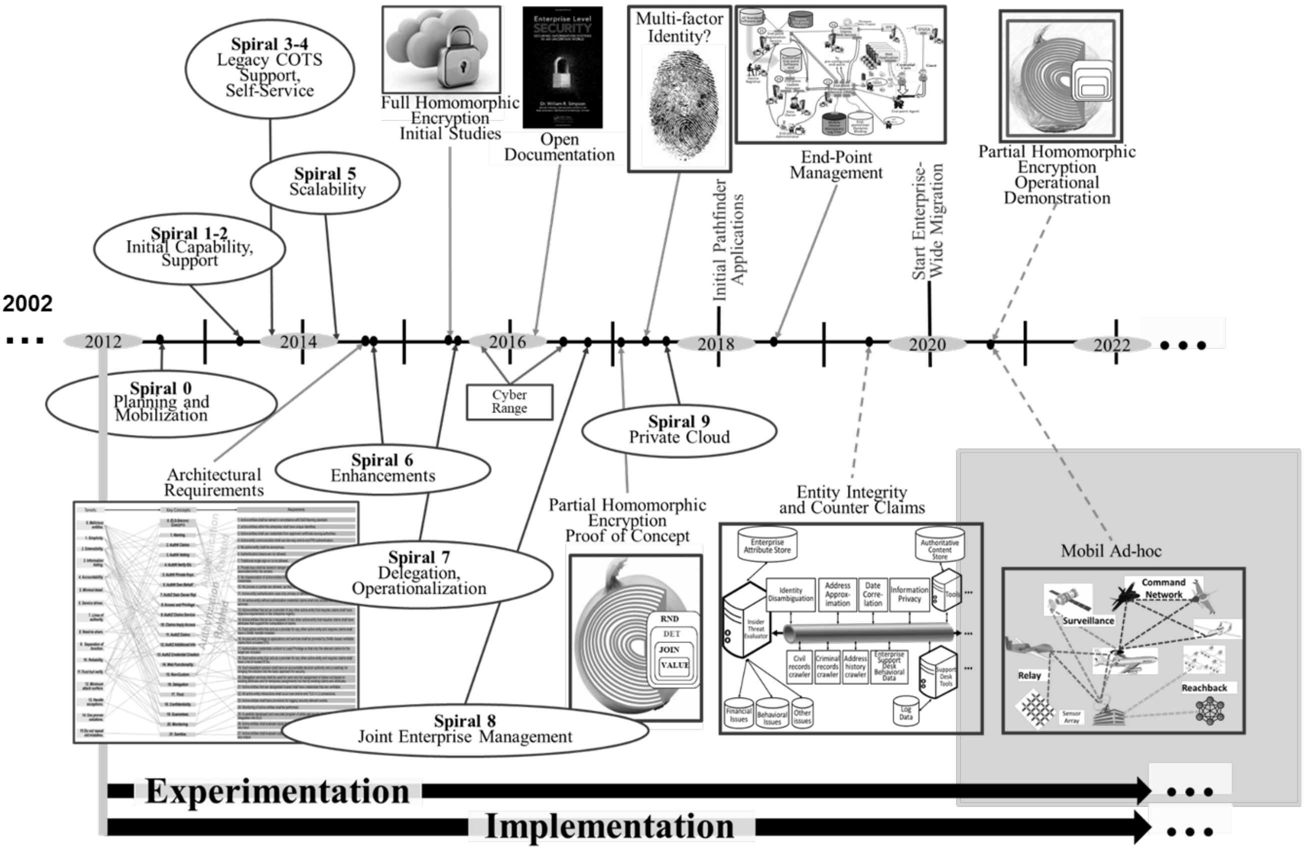

Fig. 13. The ELS implementation timeline – seven years and counting.

REFERENCES

[1] The OSI Model's Seven Layers Defined and Functions Explained, https://support.microsoft.com/en-us/kb/103884.

[2] WEP vs WPA Encryption, NETGear Support, http://kb.netgear.com/app/answers/detail/a_id/20043/~/wep-vs-wpa-encryption?cid=wmt_netgear_organic

[3] Understanding Voice and Data Link Networking, .Northrop Grumman's Guide to Secure Tactical Data Links, http://www.northropgrumman.com/Capabilities/DataLinkProcessingAndManagement/Documents/Understanding_Voice+Data_Link_Networking.pdf

[4] MANET Definition, http://techterms.com/definition/manet

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

[6] William R Simpson, and Kevin E. Foltz, "Mobile Ad-hoc for Enterprise Level Security," Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2018, 23-25 October, 2018, San Francisco, USA, pp172-177

[7] William R. Simpson and Kevin Foltz, Proceedings of The 20th World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI, "Enterprise Level Security - Basic Security Model", Volume I, WMSCI 2016, Orlando, Florida, 8-11 March 2016, pp. 56-61.

[8] Kevin E. Foltz and William R. Simpson, Proceedings of The 20th World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI, "Enterprise Level Security – Basic Security Model," Volume I, WMSCI 2016, Orlando, Florida, 8–11 March 2016, pp. 56–61.

[9] Email from Rudy Rihani, Project Manager, Accenture Corporation, dated March 6, 2016, Subject: " manpower savings with ELS"

[10] X.509 Standards

a) DoDI 8520.2, Public Key Infrastructure (PKI) and Public Key (PK) Enabling, 24 May 2011

b) JTF-GNO CTO 06-02, Tasks for Phase I of PKI Implementation, 17 January 2006

c) X.509 Certificate Policy for the United States Department of Defense, Version 9.0, 9 February 2005

d) FPKI-Prof Federal PKI X.509 Certificate and CRL Extensions Profile, Version 6, 12 October 2005

e) RFC Internet X.509 Public Key Infrastructure: Certification Path Building, 2005

f) Public Key Cryptography Standard, PKCS #1 v2.2: RSA Cryptography Standard, RSA Laboratories, Oct 27, 2012

g) PKCS#12 format PKCS #12 v1.0: Personal Information Exchange Syntax Standard, RSA Laboratories, June 1999; http://www.rsa.com/rsalabs/node.asp?id=2138 PKCS 12 Technical Corrigendum 1, RSA laboratories, Feb 2000

[11] William R Simpson, and Kevin E. Foltz, "Secure Identity for Enterprises," IAENG International Journal of Computer Science, vol. 45, no. 1, pp 142-152, ISSN: 1819-656X, February 2018.

[12] TLS family Internet Engineering Task Force (IETF) Standards

a) RFC 2830 Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security, 2000-05

b) RFC 3749 Transport Layer Security Protocol Compression Methods, 2004-05

c) RFC 4279 Pre-Shared Key Ciphersuites for Transport Layer Security (TLS), 2005-12

d) RFC 5246 The Transport Layer Security (TLS) Protocol Version 1.2, 2008-08

e) RFC 5289 TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM), 2008-08

f) RFC 5929 Channel Bindings for TLS, 2010-07

g) RFC6358 Additional Master Secret Inputs TLS, 2012-01

h) RFC 7251 AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS, 2014-06

i) RFC 7301 Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension, 2014-07

j) RFC 7457 Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS), 2015-02

[13] Michael Ortisi, "Recover a RSA Private Key from a TLS Session with Perfect Forward Secrecy", BlackHat 2016, August 2016. https://www.blackhat.com/docs/us-16/materials/us-16-Ortisi-Recover-A-RSA-Private-Key-From-A-TLS-Session-With-Perfect-Forward-Secrecy-wp.pdf

[14] "RSA Algorithm for public key encryption", https://chronichacker.wordpress.com/2011/04/04/rsa-algorithm-for-public-key-encryption/

[15] "Using the CRT with RSA", http://www.di-mgt.com.au/crt_rsa.html

[16] "The Chinese Remainder Theorem", http://www.di-mgt.com.au/crt.html#chineseremaindertheorem

[17] Organization for the Advancement of Structured Information Standards (OASIS) open set of Standards

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

b) P. Mishra et al. Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, March 2005.

c) S. Cantor et al. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, March 2005

[18] William List and Rob Melville, IFIP Working Group 11.5, Integrity In Information, Computers and Security, Volume 13, Issue 4, pp. 295–301, Elsevier, doi:10.1016/0167-4048(94)90018-3, 1994.

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

[20] 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.

[21] William R. Simpson, and Kevin E. Foltz, Lecture Notes in Engineering and Computer Science, "Enterprise End-point Device Management," In Process, Proceedings of the World Congress on Engineering, July 2018, Imperial College, London, pp. 331-336.

[22] S. Ghemawat, H. Gobioff, and S.-T. Leung. The Google filesystem. In SOSP, 2003.

[23] G. Graefe. Query evaluation techniques for large databases.ACM Comput. Surv., 25(2), 1993.

[24] J. Hammerbacher. Managing a large Hadoop cluster. Presentation, Facebook Inc., May 2008.

[25] P. Mishra and M. H. Eich. Join processing in relational databases. ACM Comput. Surv., 24(1), 1992.

[26] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins. Pig latin: A not-so-foreign language for data processing. In SIGMOD, pages 1099–1110, 2008.

[27] Hardware security module, Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Hardware_security_module

[28] Common Access Card (CAC) http://www.cac.mil/common-access-card/

[29] Trusted Platform Module (TPM) Summary, http://www.trustedcomputinggroup.org/trusted-platform-module-tpm-summary/

[30] D. A. Schneider and D. J. DeWitt. A performance evaluation of four parallel join algorithms in ashared-nothing multiprocessor environment. In SIGMOD, 1989.

[31] William R. Simpson, and Kevin E. Foltz, "Assured Identity for Enterprise Level Security," Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2017, 5-7 July, 2017, London, U.K., pp. 440-445

[32] William R. Simpson, Coimbatore Chandersekaran and Andrew Trice, “A Persona-Based Framework for Flexible Delegation and Least Privilege,” Electronic Digest of the 2008 System and Software Technology Conference, Las Vegas, Nevada, May 2008.

[33] William R. Simpson, Coimbatore Chandersekaran and Andrew Trice, “Cross-Domain Solutions in an Era of Information Sharing,” The 1st International Multi-Conference on Engineering and Technological Innovation: IMET2008, Volume I, Orlando, FL, June 2008, pp. 313–318.

[34] Coimbatore Chandersekaran and William R. Simpson, “The Case for Bi-lateral End-to-End Strong Authentication,” World Wide Web Consortium (W3C) Workshop on Security Models for Device APIs, 4 pp., London, England, December 2008.

[35] William R. Simpson and Coimbatore Chandersekaran, “Information Sharing and Federation,” The 2nd International Multi-Conf. on Engineering and Technological Innovation: IMETI2009, Volume I, Orlando, FL, July 2009, pp. 300–305.

[36] Coimbatore Chandersekaran and William R. Simpson, “A SAML Framework for Delegation, Attribution and Least Privilege,” The 3rd International Multi-Conf. on Engineering and Technological Innovation: IMETI2010, Volume 2, pp. 303–308, Orlando, FL, July 2010.

[37] William R. Simpson and Coimbatore Chandersekaran, “Use Case Based Access Control,” The 3rd International Multi-Conference on Engineering and Technological Innovation: IMETI2010, Volume 2, pp. 297–302, Orlando, FL, July 2010.

[38] Coimbatore Chandersekaran and William R. Simpson, “A Model for Delegation Based on Authentication and Authorization,” The First International Conference on Computer Science and Information Technology (CCSIT-2011), Springer Verlag Berlin-Heildleberg, Lecture Notes in Computer Science, 20 pp.

[39] William R. Simpson and Coimbatore Chandersekaran, “An Agent Based Monitoring System for Web Services,” The 16th International Command and Control Research and Technology Symposium: CCT2011, Volume II, Orlando, FL, April 2011, pp. 84–89.

[40] William R. Simpson and Coimbatore Chandersekaran, “An Agent-Based Web-Services Monitoring System,” International Journal of Computer Technology and Application (IJCTA), Vol. 2, No. 9, September 2011, pp. 675–685

.

[41] William R. Simpson, Coimbatore Chandersekaran, and Ryan Wagner, “High Assurance Challenges for Cloud Computing,” Lecture Notes in Engineering and Computer Science: Proceedings World Congress on Engineering and Computer Science 2011, WCECS 2011, 19–21 October 2011, San Francisco, USA, pp. 61–66.

[42] Coimbatore Chandersekaran and William R. Simpson, “Claims-Based Enterprise-Wide Access Control,” Lecture Notes in Engineering and Computer Science: Proceedings World Congress on Engineering 2012, WCE 2012, 4-6 July 2012, London, U. K., pp. 524–529.

[43] William R. Simpson and Coimbatore Chandersekaran, “Assured Content Delivery in the Enterprise,” Lecture Notes in Engineering and Computer Science: Proceedings World Congress on Engineering 2012, WCE 2012, 4–6 July 2012, London, U. K., pp. 555–560.

[44] William R. Simpson and Coimbatore Chandersekaran, “Enterprise High Assurance Scale-up,” Lecture Notes in Engineering and Computer Science: Proceedings World Congress on Engineering and Computer Science 2012, WCECS 2012, 24-26 October 2012, San Francisco, USA, pp. 54–59.

[45] Coimbatore Chandersekaran and William R. Simpson, “A Uniform Claims-Based Access Control for the Enterprise,” International Journal of Scientific Computing, Vol. 6, No. 2, December 2012, ISSN: 0973-578X, pp. 1–23.

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

[47] Simpson, William R., and Kevin E. Foltz, “Enterprise Level Security: Insider Threat Counter-Claims,” Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2017, 25–27 October, 2017, San Francisco, USA, pp. 112–117.

[48] William R. Simpson and Kevin E. Foltz, Proceedings of the 22nd International Command and Control Research and Technology Symposium (ICCRTS), “Escalation of Access and Privilege with Enterprise Level Security,” Los Angeles, CA. September 2017, pp. TBD.

[49] William R. Simpson and Kevin E. Foltz, Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017), Volume 1, pp. 177–184, Porto, Portugal, 25–30 April, 2017, “Enterprise Level Security with Homomorphic Encryption,” SCITEPRESS – Science and Technology Publications.

[50] Kevin E. Foltz and William R Simpson, “Enterprise Considerations for Ports and Protocols,” Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2016, 19–21 October, 2016, San Francisco, USA, pp.124–129.

[51] Kevin E. Foltz and William R Simpson, “Simplified Key Management for Digital Access Control of Information Objects,” Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2016, 29 June–1 July, 2016, London, U.K., pp. 413–418.

[52] F Kevin E. Foltz and William R. Simpson, Wessex Institute, Proceedings of the International Conference on Big Data, BIG DATA 2016, “Access and Privilege in Secure Big Data Analysis,” 3–5 May 2016, Alicante, Spain, pp. 193–205.