Secure Identity for Enterprises

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

***Abstract*— Increasing threat intrusions to enterprise computing systems have led to a formulation of guarded enterprise systems. The approach was to put in place steel gates and prevent hostile entities from entering the enterprise domain. The current complexity level has made the fortress approach to security implemented throughout the defense, banking, and other high-trust industries unworkable. The alternative security approach, called Enterprise Level Security (ELS), is the result of a concentrated 15-year program of pilots and research. The primary identity credential for ELS is the 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, and a TLS 1.2 communication pipeline is established. This process was deemed to provide a high enough identity assurance to proceed. However, in some instances the PIV card is not available and a compatible approach is needed. Additionally, a derived credential may be used on mobile devices. This paper discusses multi-level authentication approaches designed to satisfy the level of identity assurance specified by the data owner, to add assurance to derived credentials, and to be compatible with the ELS approach for security.**

***Index Terms* — Identity, Authentication, Multi-Factor Authentication, Enterprise Level Security**

I. INTRODUCTION

Identity is complex. If you Google identity you can well come away more confused than when you first started. Identity commonly includes a number of attributes that may change over time, like job description, feelings, marital status, etc. In IT we need to separate things that are static from things that are temporal in nature. The simplest definition presented by Merriam-Webster [1] is probably the best place to start: "Identity is who someone is: the name of a person." For IT purposes, the identity is a label that is unique and attached to only one entity or object. This is the static part of identity, and it can be tied to something you know, something you have, or something you are. Each of these properties is an attempt to bind the name (and associated data and credentials) to the entity. For purposes of computation, this is the element we rely upon. There are familiar identities (recognized in my domain) and unfamiliar or less familiar identities (not recognized in my domain or not recognized at all). Establishing identity is a necessary first step in information technology security activities. These include:

Manuscript received 10 January 2018; revised 10 March 2018. 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,

Kevin E. Foltz, 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)

- Confidentiality. The confidentiality security service is defined as preventing unauthorized disclosure of data (both stored and communicated). Confidentiality services will prevent disclosure of data that is in storage or is transiting. One of the most common confidentiality mechanisms is cryptography. Familiar identities have the privilege of accessing data both stored and communicated.
- Integrity. The integrity security service includes: prevention of unauthorized modification of data (both stored and communicated), and detection and notification of unauthorized modification of data. One of the most common mechanisms for integrity is one-way hash algorithms, together with authoritative data source signature binding. Familiar identities have created or transmitted the data. Familiar identities need to know that unfamiliar identities have not modified the data.
- Availability. Availability is timely, reliable access to data and information services for authorized users. A popular attack is called denial of service (DOS), which attempts to make access unavailable. Of course, authorized users are familiar identities and unauthorized users are unfamiliar identities.
- Authenticity. Authenticity is that property that ensures that the identity of a subject or resource is the one claimed. Authenticity applies to entities such as users, processes, systems, and information.
- Non-Repudiation. Non-repudiation is that property that ensures actions are attributable to the identity of the entity that invokes them.

Identity also applies to the following familiar terms:

- Malicious Entities – Unfamiliar identities,
- Accountability – Attributing actions to familiar entities,
- Lines-of-authority – An accountability chain.

Many exploits take advantage of confusing identities through masquerade, man-in-the-middle (MITM), and other approaches. In a secure environment, actions that provide access and privilege should always be preceded by a strong identity check. Since we know the enemy is present, we must avoid any mechanisms that get in between known, vetted identities, like proxies and portals. These often confuse the identity issue and lead to vulnerabilities and exploits.

Naming provides the initial address for most entities and is one of the fundamental abstractions for dealing with complexity. The name for an entity (an individual, organization, or facility) provides a pointer into a set of labeled properties, managed by a registration service. Names provide the handle by which further certificates

Sidebar

Identity and Naming—Case Study

When Sam Cinco first came to XYZ Corporation, he was provided an identity in accordance with the company policy. His Name was Sam.Cinco3579. The resources people were careful about uniqueness in this case because there may well be another Sam Cinco. The extra digits ensured that his identity would always be unique. He first worked in corporate services, where he did a stellar job of providing payroll data and other financial services. After four years, his job was taken by a newer hire and he moved to the executive suite, where he provided services connected with dashboards for executive desktops. When he moved, the resources department recorded his new functional location, his new network affiliations, his equipment upgrades, and his new responsibilities. Sam is a Server Automated Machine made by Cinco, and there are twenty or so at XYZ Corporation. And although they are twenty, each is unique and accountable for its own actions. The resources department here was the IT resources department, and it had parallel function to the human resources department. Sam was more than an identity—he had a collection of attributes that provided his location, functionality, responsibilities, and privileges within the XYZ Corporation.

There are two points from this example:

- The first is that all entities, human or machine, need unique identities. Uniqueness cut across space and time, in that we do not wish old identities to be re-used even when the old identity is thought to no longer be part of the system. At best, reusing an identity can create an identification ambiguity and at worst it can open the system to masquerade attacks.
- The second is that identity is not enough. Attributes that may change over time must be available to each entity regarding its place, function, capabilities, and privilege.

Implications for Information Security

A number of implications can be derived from the case study.

1. We need some way to recognize an identity as being part of our eco-system. When John.Smith1234 wishes to interact with Sam.Cinco3579, each has to have a concrete way of identifying who the other is.
2. This is critical to the XYZ Corporation (the enterprise); it is up to the enterprise to provide credentials that can be used in securing this identity. A secret handshake may be possible, a Kerberos ticket may suffice, but the identity must be recognizable, and according to our tenets, it must be able to be verified and validated. Passwords and secret handshakes do not pass muster. Credentials that can be verified and validated must be provided by an enterprise authority or an enterprise trusted authority.
3. Point 2 requires a credential of some form for each and every entity that might take part in a transaction. This credential can have a relatively long life in that identity is unique and does not change over time.
4. Identity Credentials require a trusted identity credentialing agent, a registry, and software services associated with verification. They also require a way to challenge the presenter of a credential to prove that he is the owner of that credential.
5. Identity is only part of the issue. Knowing the identity of an entity is not the same as knowing the attributes of that entity. This implies that a provider of services in a transaction must have some measure of what the entity's attributes are, either the data itself or some claim.
6. Data or claims must be presented in a manner that can be verified and validated.
7. Point 6 requires an attribute or claims credential. This credential must have a relatively short life because entities move around and change their locations, functionalities, responsibilities, and privileges. Claims and attributes are not unique and change over time.
8. Care must be taken that attribute or claims credentials are not reused with the same or other identities.
9. Point 8 requires a trusted attribute or claims credentialing agent that can generate a credential at the time a request is made.
10. Point 9 requires a registry and software services associated with creation of the credential.
11. Point 9 also requires a way to bind the presenter of an attribute or claims credential to the same identity that presented and verified the identity credential.

and other data concerning the entity can be accessed. Integrity, data concerning the entity can be accessed. Integrity, quality, performance, and scalability all hinge on a solid design for directory namespace and schema.

Architecturally, software components such as the directory service agent (DSA) depend on the schema to provide access controls and security to directory data. A namespace design determines the structure of the directory information tree (DIT) and how an organization can partition the directory for replication.

We often take for granted the name associated with a uniform resource locator—an identity. The power of the web was fully realized when the web developers defined the uniform resource locator (URL). This is a registry that assigns names to a resource and enforces the uniqueness of that name [2]. This provides a way to specify where you wish to go without ambiguity and "…the relationship between uniform resource identifiers (URI)s, URLs, and uniform resource name (URN)s, describes how URI schemes and URN namespaces identities are registered…." The lack of ambiguity is the principle thing. Computers and their associated software get unpredictable when a label can mean more than one thing.

Some attributes are so important that they may alter the expected behavior of an identity. This can be typified by an individual with multiple roles. The example given often is a former active duty military person, who is an active National Guard member and a defense contractor. But the two aspects of the described person could be expressed as attributes related to multiple assignments in the enterprise. John.Jones123 may be an administrator for the payroll system, and a user of that system for his own data. Everybody is a user in that they fill in time sheets and check their leave, etc. These may not overlap (John.Jones123 may be administrator for the executive department and reside in the IT department). This may be by design, with enterprise policy preventing John.Jones123 from administering his own accounts. The software may give him all the privileges of both personas (this is difficult for the software to do and prone to error). In the event that the software is not that clever, the software will need to know which persona it is dealing with. At one level a persona could be considered another identity (say John.Jones123a). The enterprise could issue credentials for each persona and provide them to John.Jones123 to use as needed. However, this has a number of implications, not the least of which is maintaining duplicate attribute files, and tracking multiple credentials. The easiest way to handle these things is to ask John.Jones123 which persona he wants to be. Since John.Jones123 has both personas and the privileges that accrue to each, John should be asked to mitigate the ambiguity that the persona issue creates. If he were issued multiple credentials, he would decide which one to use. This paper is based in part on a paper published by WCE 2017 [3].

II. BACKGROUND

Adversaries continue to penetrate, and in many cases, already exist within, our network perimeter, i.e., 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 [6], but this assumption has been proven wrong by a multitude of reported network-related incidents. The previous statements are no longer controversial but a wise assumption for data and information security practitioners. Network attacks are pervasive, and nefarious code is present even in the face of system sweeps to discover and clean readily apparent malware. Reported breaches exceed 8,000 and a loss of over 1 billion records since 2005 and 1172 in 2017 alone [7]. The focus of this paper is on the security aspects of countering existing known and unknown threats based on robust identity and access management (IdAM) and on how this access control system can dynamically support mission information requirements.

A. *Current Approaches – A Brief Review*

When intercommunications between computers began, there were no security concerns, just a bunch of academics experimenting. As we began to actually make this happen, we found it useful and the would-be ne'er-do-wells would begin to deface our notes back and forth. As we organized into enterprises, the hackers got better and real resources were at risk.

Network operating systems generally require that a user be authenticated in order to log onto the network [6].This can be done by entering a password, inserting a smart card and entering the associated PIN, proximity of a nearfield device, providing a biometric verification or using some other means to prove to the system that you are who you claim to be. The network may provide an identity token that provides identity to applications and providers of service on the network. This token (sometimes called single-sign-on (SSO)), is vulnerable and subject to theft or forgery and may be replayed for nefarious activity. When logging on to a web site, you are not generally aware of whether you are logging into an application or a network because the magic is all behind the scenes. With ELS, the network is the medium and authentication is to the application or provider of services. This of course, must be bi-lateral so that both entities have an assurance of their communicating partner's identity.

For some networks, a message integrity process is invoked called Internet Protocol Security (IPSec). IPSec transmissions can use a variety of authentication methods, including ticket or token based approaches, public key certificates issued by a trusted certificate authority (CA), or a simple pre-shared secret key (a string of characters known to both the sender and the recipient). This type of authentication assumes the fortress model previously discussed. As with any authentication process, the requester and provider must support a common method.

For the remainder of this paper, we will discuss the requester/provider authentication process.

The earliest form of protection was the use of passwords. We have gotten much more sophisticated with passwords over the years, regularly changing them, making them complex, and not using the same passwords for multiple purposes. As an identity method, these passwords were ok as long as we could keep the secret. The username/password unlocked an account with the target and the account had my privileges and assets and, etc. Passwords are easy. The trouble with passwords is that they are not safe. They must be strong, and they must be updated and they must be a kept secret by several entities, they must be transmitted, and they must be stored at multiple sites for logistical as well as security reasons. All this complexity means you either use simple passwords over and over, or you write your passwords down, or you trust all of your passwords to some single point of failure. The thieves are getting good at stealing them. Passwords falling into the wrong hands are one of the biggest causes of network vulnerability: 63% of known data breaches involved weak or stolen passwords [7].

Attempts to resolve the password issues include extended passwords [8] by using special characters or adding additional factors to the password, such as adding salt (random data that is used as an additional input to a one-way function that "hashes" a password or passphrase). Salt addresses recovery of multiple passwords from lists of hashed passwords. It forces an attacker to guess one at a time instead of all at once. The salt process is an attempt to thwart a password cracking method using rainbow tables [9]. Passwords and accounts are part of the fortress approach where we constantly renew the mantra of adding more and better software to sort, identify, and deny the unwanted from our enterprises. It turns out that the strong password is still a weak identity credential.

Attempts to thwart programs from trying to "guess" passwords include two or three-factor authentication (discussed later in this paper) and the captcha [10] which is supposed to be a test (visual, sound or other) that humans can easily pass but programs have difficulties. Most of these tests have been cracked [11]. If the enterprise uses its knowledge of the user to supplement the password with secondary factors the strength of authentication is greatly increased, but asking simple questions does not appear to be the solution. Many of these questions may be answered with modest research [12]. Better is the out-of-band query, where a message is sent to a phone number or e-mail, and a correct response would indicate the presenter of the password at least has the out-of-band device [13]. Mobile devices have recently begun to use the camera function to provide biometric authentication to the device. These devices use a fingerprint, face print or iris scan to verify the person who is in control of the device is the one registered to that device [14]. At least authentication to the device does not have to be transmitted and the digital representation is

less subject to theft. The jury is still out on these attempts to increase security [15].

A strong credential would be one where you alone keep the secret locked away where only you can get to it, and you identify yourself by proving that you have this secret locked away (not necessarily producing the secret).

One identity credential that partially satisfies these criteria is the One-Time Password (OTP) [16]. OTP is the provision of a single-use password that is provided at the time of use.

- Simple forms of the OTP include distributed lists (where a sequential list of user passwords are provided to the user –bookkeeping is a bit tedious here, and the lists may be stolen or intercepted).

- Somewhat more sophisticated forms include algorithmically produced codes (usually based upon some shared values between the requester and the provider). These suffer from control of secrecy issues in both the shared values and the algorithm.

- A more satisfactory OTP solution would include hardware provision of password generators tied to the user and synchronization between the user and the target of communication. Some Personal Identification Verification (PIV) (see next paragraph) include OTP generators included in their functionality. These have the ability to be registered, verified and revoked if lost, or stolen. These suffer from needing to be transmitted, making them available to a Man-in-the-Middle, and algorithm cracking or theft. Theft is less of a concern since the password is one time use. Algorithm theft is another issue. This latter has already been accomplished for one provider [17].

Another Identity credential that meets this criterion is the PIV card [18] or equivalent. This is the preferred credential for ELS. The PIV card uses PKI credentials and has a public certificate, issued by a recognized certificate issuing authority, with a public key and a private key locked into tamper-proof hardware. Identity is established when the holder of the PIV card can decrypt a message encrypted by his/her public key. The proof is called Holder-of-Key (HOK). If the user or the card is compromised, the certificate may be revoked [19]. While this does not solve all identity problems, it at least provides a strong credential that is more difficult to steal.

B. *The Enterprise Level Security Approach*

Enterprise Level Security (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. It is important to note that 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 [20]. From there, a set of enterprise-level requirements are formulated that conforms to the tenets and any high-level guidance, policies, and requirements. A working prototype has been developed and evaluated for security, functionality, and scaling issues.

The basic tenets, used at the outset of the ELS security model, are the following:

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 preserved
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.

These tenets are foundational to ELS and are described in detail in [21].

The current paper-laden access control processes for an enterprise operation are plagued with ineffectiveness and inefficiencies. In a number of enterprises, tens of thousands of personnel transfer locations and duties annually, which on a daily basis introduces delays and security vulnerabilities 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.

Early calculations show that for government and defense, 90–95% of recurring man-hours will be saved and up to 3 weeks in delay for access request processing will be eliminated by ELS-enabled applications [22]. While a 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 1.

This work is part of a larger body of work termed Consolidate Enterprise IT Baseline (CEITB). The security aspects of this baseline are termed Enterprise Level Security (ELS). The element and sub element locations within the baseline are shown in Figure 2. Each of the sub-elements must conform to both the CEITB and ELS requirements as applicable.

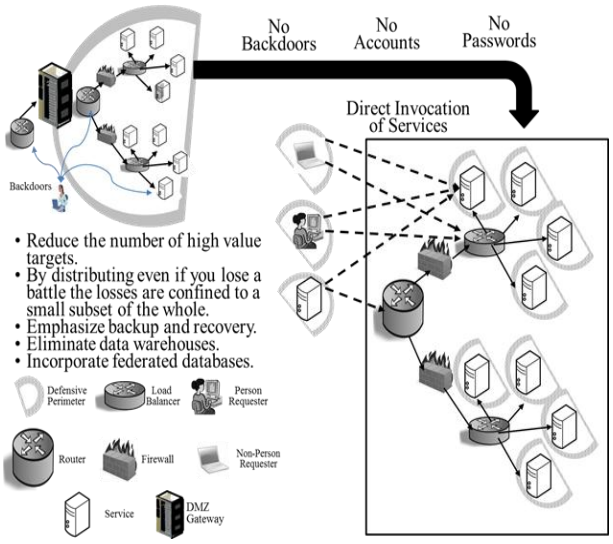

Fig 1. Distributed Security Architecture

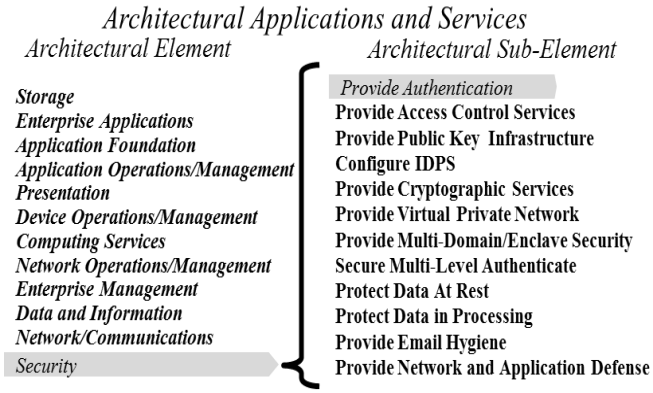

Fig 2 CEITB Architectural Element

III. ENTERPRISE LEVEL SECURITY

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. It is important to note that 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 [20]. From there, a set of enterprise level requirements are formulated that conforms to the tenets and any high level guidance, policies and requirements.

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. Early calculations show that for government and defense 90-95% of recurring man-hours are saved and up to 3 weeks in delay for access request processing are eliminated by ELS-enabled applications [22]. While 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 1.

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 – know that you received exactly what was sent;
- Require Explicit Accountability – monitor and log transactions.

ELS has been shown to be a viable, scalable alternative to current access control schemas [23]. A complete description of ELS basics is provided in [3].

IV. IDENTITY ISSUES IN ELS

Identity in the enterprise is a unique representation of an entity. For users, it begins with the human resources who maintain their files. The assigned identity is called the Distinguished Name (DN) and it must be unique over space and time. There may be five John Smiths in the enterprise, but only one John.Smith2534, UID=Finance, HID=Chicago. These and PKI information are normally encoded into a Personal Identity Verification (PIV) card for network access and provided to the entity for its use. Certain pieces of the information may be tagged as verifiable by DN for identity purposes, such as wife's middle name.

The PIV card alone is insufficient in many instances. The current model of device security is based upon a fortress approach with well defended entry points. When mobile devices began to proliferate, and in forms that were unanticipated, it became apparent that a separate management system was needed to secure the multitude of devices that were not under control in the computing center. Within the computing center a legion of administrators maintained servers, keeping them updated, patched and in proper configuration, but the mobile devices were not always on and connected and often nowhere near the administrators of the computing system. Several designs for Mobile Device Management (MDM) were provided [25-28] – many of these included provisions for devices provided by the enterprise members, known as Bring Your Own Device (BYOD) [29].

These devices operate on a derived certificate that is place in the mobile devices hardware tamper-proof Secure Key Storage and Use (SKSU) with attestation. One such standard for this function is the Trusted Platform Module (TPM) [30]. SKSU is the starting point of trust for enterprise registered devices. The SKSU manages a public/private key pair, the private key of which cannot be removed or copied from the SKSU. The public key is recorded in the device registry when the device is issued to a user. All future communications with the device are tied

back to this key pair. The device proves ownership of the private key in order to provide validated information about the device and its properties, such as installed or connected hardware, installed operating system, installed software, and configuration settings. The SKSU is integrated into the operating system in order to properly account for application and configuration changes. The SKSU is implemented at a sufficiently low level to prevent software attempts to subvert it. This is necessary in particular to prevent leakage of the private key. The SKSU on a mobile device has provisions for storage of derived PKI certificates for authorized users and temporary certificates for guest users [31].

These devices are often outside of the boundaries of the enterprise and the individual who has possession of the device may be in question. Authentication and binding of mobile devices typically requires a two-factor authentication since there is no separate hardware device for storage of private keys and the devices are generally physically accessible to non-vetted personnel. The second factor configured for the device is typically biometric (out-of-band is associated with the mobile device) with the biometric determined by device capabilities (face recognition, voice recognition, fingerprint, etc.). The call for second factor authentication comes from the Security Token Service (STS) upon recognizing the use of a derived credential.

There is also a need to allow users without PIVs some degree of access based on alternative authentication methods. PIVs may not be available to all, but also, the user device may not be capable of reading and using a PIV. Additional use cases include lost PIV, waiting for issuance of a PIV, or a user being unable to get a PIV compatible with the ELS certificate authority trust. Additionally, there are federation partners, contractors, and other vetted external individuals with short-term needs.

Each application ultimately decides what kind of authentication is strong enough (through a registration process with Enterprise Attribute Ecosystem (EAE)).

The creation of a non-PIV identity comprises three separate stages. The first stage is creation of a proposed identity. This value is provided by the user. The goal is to correlate this with the enterprise files. It may be an email, a common name, or simply a name. The second stage is creation of a candidate identity (starting point for identity determination), in which the proposed identity is paired with an enterprise identity, and a DN is determined.

As we will discuss, the process also takes steps to verify that the pairing between the proposed identity and the DN is owned by the individual making the request. The last stage is creation of the assured identity. The candidate identity becomes the assured identity when enough correlated information and personal verification about the candidate identity has a sufficient level of pairing with the enterprise identity that it can be trusted with access to an application using his/her claims that have been computed for his/her use.

V. SCALE OF IDENTITY ASSURANCE

If you search the literature for multi-factor authentication, you will find a predominance of processes based upon account-based systems and starting with username–password [32-39]. These systems intertwine the security issues of authentication and authorization. In fact, the popular definition of multi-factor authentication merges the two:

Multi-factor authentication (MFA) is a method of computer access control in which a user is only granted access after successfully presenting several separate pieces of evidence to an authentication mechanism:

"on the basis of something they know, something they have, something they are, or something they are" [40]

ELS separates the identity and access/privilege security issues. Thus there are no accounts and no usernames with passwords. Further, ELS uses no proxies and limits access to the enterprise attribute system, thus reducing the threat surface.

Each data owner will decide what the requirements for access and privilege to their data are, and this includes the level of assurance that is acceptable. ELS represents a strong identity assurance and will be assigned a value of .80 (values are arbitrary and subject to revision). It is assumed that if the data owner wishes strong identity assurance he will specify .70 or .75 as the identity assurance value (from the collection below, the value of .75 requires bio information in the absence of PIV). This will allow all enterprise users with a PIV to actually present access and privilege claims to the application.

The lowest level of identity assurance would come from self-assertion; however, we will require several additional factors for this minimum, including a presence in the enterprise catalog, verification by an out-of-band (OOB – phone or e-mail) method; and of course for authorization, claims must be available for the individual. This lowest level will be described as User Asserted Identity with OOB verification and assigned a value of .2, which should also be the minimum specified by a data owner. A total of seven identity cases were developed, as follows, with strengths shown in Table 1:

1. Bi-lateral AUTHN (Hard Token) – AUTHN Hard
2. Bi-lateral AUTHN (prior issued Soft Token) in protected store. – ATHN Soft
3. User Asserted Identity with Out-of-Band (OOB) verification – OOB
4. User Asserted Identity with OOB verification and with any Biometric factor – OOB Bio
5. User Asserted Identity with OOB verification and with any Biometric factor and with any non-biometric multi-factor verification – OOB Bio + 1mf.

6. User Asserted Identity with OOB verification and with any non-biometric multi-factor verification – OOB + 1mf
7. User Asserted Identity with OOB verification and with three non-biometric multi-factor verifications – OOB + 3mf

Enhanced Identity Assurance:

8. Hard token plus one non-biometric multi-factor verification – Hard token + 1mf
9. Hard token plus one biometrics authentication. – Hard token + 1bio
10. Hard token plus one biometric and one non-biometric multifactor verification – Hard token +1bio + 1mf

Table I
Multifactor Authentication Identity Assurance

Method	Comment	Id Assurance
1. AUTHN Hard	Standard ELS – Strong	0.80
2. ATHN Soft	Closest to ELS	0.70
3. OOB	A Start - Minimal	0.25
4. OOB Bio	Solid	0.50
5. OOB Bio + 1mf.	Strong	0.80
6. OOB + 1mf	Moderate	0.60
7. OOB + 3mf	Strong	0.70
Greater than Normal ID Assurance directed by Web Application		
8. Hard token +	Very Strong	0.85
9. Hard token ++	Very Strong	0.90
10. Hard token +++	Highest Value	0.95

VI. A TOKEN SERVER WITH CERTIFICATE AUTHORITY

In order to preserve the ELS paradigm, a temporary soft certificate needs to be provided and the user claims must be provided with a SAML credential through TLS. The user needs to be in the attribute system with claims for services sought.

A. Non PIV STS/CA Issued X.509

Non-PIV owners go to a special token server with certificate issuance authority (STS/CA) and provide a proposed identity. This may be email or full name, etc. The STS/CA calls a service that scans the Enterprise Attribute Store (EAS) and rejects any identity that it cannot find in EAS. The STS/CA then confirms that the requester is not an automated system (via Captcha, etc.). This avoids a number of threat vulnerabilities. The STS/CA then asks questions of the non-PIV user to resolve ambiguity (if present). For example, there are five Jon Smiths in the enterprise, but only one works in Finance. The STS/CA then establishes the DN. To this point, the identity is still a proposed identity. The STS/CA saves the DN attributes in separate temporary store and sets up a server side TLS.

The next step is a requirement, and non-PIV users must maintain an OOB contact for this. This OOB (one or more) is provided to the human resources for inclusion in the user's enterprise data. The token server resolves OOB (email, phone voice, phone text, etc.) communication methods for DN. We note that OOB means not on the network, and if the enterprise desk phone is part of the enterprise network, it does not work as OOB. Anyone without at least one OOB is rejected.

At this point the token server sends a one-time token (10 minutes or less life) to the OOB and requests input. No input or improper input will be rejected. A successful exchange results in the identity moving to a candidate identity.

The STS/CA will attempt to identify if the user is using a managed device (looking for bio capability like face or fingerprints). The STS/CA retrieves the claims from the enterprise claims store for the established DN, presents a choice from among the services the user has claims to, and asks for a selection. This establishes the application for later SAML transmission. The STS/CA choses the maximum and minimum identify assurance needed for claims.

The minimum identity assurance may not be achievable with the device, and a polite rejection is issued if so. Otherwise, the token server begins a multifactor verification, including biological, if applicable. Any multi-level failure leads to exit. If the multi-factor maximum achievable authentication for the identity assurance is successful, the identity becomes an assured identity. The STS/CA then creates and issues a temporary certificate, in the name of the assured identity DN, and sends this certificate and separately the private key to a specially configured application on the user's device for installation.

The temporary certificate contains the identity assurance and has a life of 90 minutes or less. Comments in the temporary certificate, specify the assurance level and the method for the application's use as appropriate. The temporary certificate may be reused for the life of the certificate by selecting any application (this will go to the normal STS for claims).

When the user selects an application, the token server posts a SAML through the browser to the application. The SAML is specifically for the audience (selected application). The temporary certificate is used for authentication to the application, and all else works as with normal ELS for an application. The interaction between the STS/CA and the attribute system is shown in Figure 3.

B. PIV USAGE OF THE STS/CA

A PIV user may be redirected to the STS/CA when the identity assurance requirement for the web application exceeds 0.80. The post will include the identity assurance value of the user (0.80), the identity assurance value sought, and the audience for the multi-factor authentication. The STS will use the user's PIV to authenticate, and the STS/CA will try to increase the identity assurance to the level sought by the application using the methods shown in table 1.

VII. REQUIRED ADDITIONAL ELEMENTS

From an ELS standpoint, accommodation of non-PIV users adds the following requirements:

- Data Owners must specify the level of assurance on applications when specifying requirements for access and privilege in the enterprise service registry.
- STS/CA for non-PIV Users needs to be developed.
- An additional service must be placed in the EAE for comparison of attributes in DN retrieval.
- STS/CA must have full crypto and key management capability (generating asymmetric key pairs).
- Device software is needed to install temporary certificates on the end user device.
- The application must recognize temporary certificates generated by the STS/CA (STS/CA must be placed in the trust store).
- The application must recognize SAML certificates provided by the STS/CA.
- The application must check signatures and timestamp, but there is no need for revocation checking of the temporary certificate.

Advantages of the new additions:

- The derived process in this paper is not username/password – there are no accounts and no storage of user data.
- The process will handle retirees, contractors, and temporary employees if they are included in EAS.
- The process will handle missing or forgotten PIV cards.
- Since DN is in EAS claims are computed for each DN in the enterprise stores.
- Claims may be from Delegation (recommend non PIV cannot delegate)
- All of the ELS software and handlers work without modification.
- The EAS has same attack surface as before.
- Temporary certificates expire out of system quickly.

However, the following disadvantages are noted:

- Only covers person entities (not for Non-Person Entities (NPE) – but an adaption may be possible for NPEs).
- Software certificates and keys from STS/CA may be extracted and shared – mitigated through device management and short validity window.
- Manipulation of identities is possible (OOB requires the threat to have an OOB device in EAS that is really not part of the network).
- The threat's ability to initiate exchange with STS/CA (takes on all comers – reconnaissance by threat entities is facilitated under these circumstances).
- Intercept of temporary credentials (transmission is in TLS – some mitigation).
- On-device recovery of temporary credential (short duration provides mitigation)
- Credential forging (signatures and timeouts are some mitigations).
- The current identity assurance process treats all biometric identifications the same. For future versions, we may wish to distinguish between the types of biometric.
- The current identity assurance process treats all multi-factor queries as the same. For future versions, we may wish to distinguish between the types of multi-factor queries

VIII. ADDITIONAL CONSIDERATIONS

At this point, we have established an identity, but we have not mentioned trust. Even though the identity is known to the enterprise, and may have access and privileges based upon his SAML credentials, trust is still an issue. Trust manifests itself in evaluation of insider threat potential.

> "An insider threat is a malicious threat to an organization that comes from people within the organization, such as employees, former employees, contractors or business associates, who have inside information concerning the organization's security practices, data and computer systems." [41].

From the IT standpoint, we have adopted the concept of veracity and tailored its definition to be more amenable to self-assessment in ELS environments. Entity Veracity is the degree to which an entity is worthy of trust as demonstrated by resistance to or avoidance of factors that denigrate trust or compromise reliability. Positive factors may enhance veracity, and negative ones may reduce veracity. Veracity is based upon recognized accomplishments and failures, along with the associated stress factors or other trust debilitating factors present. A history of actions in difficult circumstances provides strong evidence for or against veracity. This is a new area and is just beginning to be implemented. A preliminary model is presented in [42].

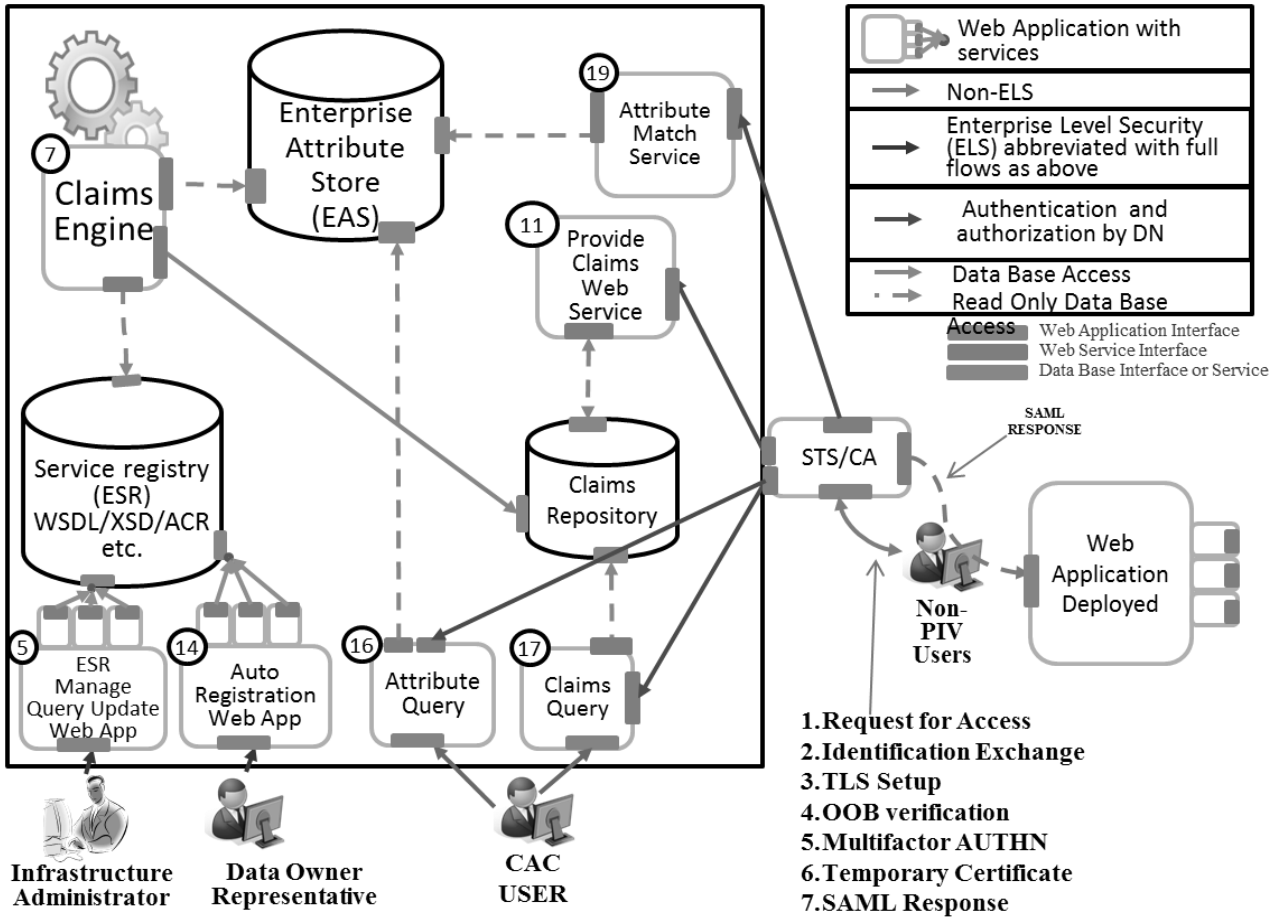

Fig 3. Partial Enterprise Attribute Ecosystem (EAE) for Non-PIV Users and Extended Identity Assurance

IX. SUMMARY

We have reviewed the identity 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). 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 on their enterprise need and the level of identity assurance needed to provide access and privilege to applications. The process is fully compatible with ELS and works as a complement to existing infrastructure. Finally we have reviewed a few of the advance topics related to identity that are currently being developed. This work is part of a body of work for high-assurance enterprise computing using web services. Elements of this work are described in [24, 43-58].

REFERENCES

[1] Merriam Webster online dictionary - for identity, http://www.learnersdictionary.com/definition/identity.

[2] Internet Engineering Task Force (IETF) Standards, RFC 3305: Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations, August 2002.

[3] 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., pp440-445

[4] 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, pg. 4.

[5] Identity Theft Resource Center (ITRC) Breach Report, http://www.idtheftcenter.org/images/breach/2017Breaches/ITRCBreachStatsReportSummary_2017.pdf, last accessed on 11/22/2017.

[6] TechRepublic, McAfee, Understanding and selecting authentication methods, https://www.techrepublic.com/article/understanding-and-selecting-authentication-methods/, last accesed on 27 November 2017.

[7] Verizon Communications, Verizon 2016 Data Breach Investigations Report, web reference, http://www.verizonenterprise.com/resources/reports/rp_DBIR_2016_Report_en_xg.pdf, last accessed on 11/22/2017.

[8] Open Web Application Security Project (OWASP) Foundation, https://www.owasp.org/index.php/Password_special_characters , April 2013, last accessed on 23 November 2017.

[9] Learn Cryptography, Password salting, https://learncryptography.com/hash-functions/password-salting, copyright 2017, last accessed on 23 November 2017.

[10] StackExchange, Information Security, 2-Factor Authentication vs Security Questions, web reference, https://security.stackexchange.com/questions/96884/2-factor-authentication-vs-security-questions, last accessed on 23 November 2017.

[11] IBM Corporation, Upgrade Your Security with Mobile Multi-Factor Authentication, https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WGW03242USEN&, last accessed on 23 November 2017.

[12] Katheleen Hickey, GCN Magazine, Biometric authentication growing for mobile devices, but security needs work, https://gcn.com/articles/2016/12/07/biometrics-maturity.aspx?admgarea=TC_Mobile, December 2016, last accessed on 23 November 2017.

[13] Liam M. Mayron , Arizona State University, Biometric Authentication on Mobile Devices, IEEE Security & Privacy, Volume: 13, Issue: 3, May-June 2015.

[14] Gemalto, One Time Password (OTP), https://www.gemalto.com/companyinfo/digital-security/techno/otp , last accessed on 23 November 2017.

[15] Goodin, Dan, Security Editor at Ars Technica, RSA SecurID software token cloning, https://arstechnica.com/information-technology/2012/05/rsa-securid-software-token-cloning-attack/ , May 2012, last accessed on 23 November 2017.

[16] National Institute of Technology and Standards, Computer Security Division, Applied Cybersecurity Division, Best Practices for Privileged User PIV Authentication, April 21, 2016, https://csrc.nist.gov/publications/detail/white-paper/2016/04/21/best-practices-for-privileged-user-piv-authentication/final, last accessed on 11/22/2017.

[17] Lawton, Steven, tom's IT PRO, Introduction to Public Key Infrastructure (PKI), March 2015, http://www.tomsitpro.com/articles/public-key-infrastructure-introduction,2-884.html, last accessed on 23 November 2017.

[18] Technical Profiles for the Consolidated Enterprise IT Baseline, release 3.0. Available at (CAC required) (currently working 4.0): https:// intelshare.intelink.gov/sites/ afceit/TB

[19] William R. Simpson, and Kevin E. Foltz, "Ports and Protocols Extended Control for Security" IAENG International Journal of Computer Science, volume 44, number 2, pp 227-240, May 2017, IJCS_44_2_12

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

[21] Technical Profiles for the Consolidated Enterprise IT Baseline, release 4.0. Available at (CAC required) (currently working 5.0): https:// intelshare.intelink.gov/sites/ afceit/TB

[22] Briefing prepared by Accenture Corporation, "USAF Enterprise Level Security, Spiral 5, Codeless Migration of Legacy .NET Applications, High Performance Claims Engine and Performance Test Results," dated 27 September 2013.

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

[24] IBM Corporation, web reference, "Mobile Device Management (MDM)", https://www.ibm.com/security/mobile/maas360/mobile-device-management, last accessed on 18 November, 2017.

[25] AT&T Business, web reference, "CYBERSECURITY SOLUTIONS- Mobile Security", https://www.business.att.com/solutions/Family/cybersecurity/mobile-security/, last accessed on 18 November, 2017.

[26] PC Magazine, web reference,The Best Mobile Device Management (MDM) Solutions of 2017, Web reference,https://www.pcmag.com/article/342695/the-best-mobile-device-management-mdm-software, last accessed on 18 November, 2017.

[27] MindWireless – Strategic Telecom Management, web reference, "Enterprise Mobility Management", Web reference, https://mindwireless.com/services/enterprise-mobility-management, last accessed on 18 November, 2017.

[28] TPM Main Specification Version 1.2, Revision 116, 1 March 201, TCG Published, available at: https://trustedcomputinggroup.org/wp-content/uploads/ TPM-Main-Part-1-Design-Principles_v1.2_rev116 _01032011.pdf

[29] Ferraiolo, H. , et al, NIST Special Publication 800-157, "Guidelines for Derived Personal Identity Verification (PIV) Credentials, December 2014, Web reference, http://dx.doi.org/10.6028/NIST.SP.800-157

[30] Sabzevar, Alireza Pirayesh, and Angelos Stavrou. "Universal multi-factor authentication using graphical passwords." Signal Image Technology and Internet Based Systems, 2008. SITIS '08. IEEE International Conference on. IEEE, 2008.

[31] Gordon, Whitson (3 September 2012), "Two-Factor Authentication: The Big List Of Everywhere You Should Enable It Right Now," LifeHacker, Australia. Retrieved 1 November 2012.

[32] Lamport, Leslie, "Password authentication with insecure communication," Communications of the ACM 24.11 (1981), pp. 770–772.

[33] Bauckman, Dena Terry, Nigel Paul Johnson, and David Joseph Robertson, "Multi-Factor Authentication," U.S. Patent No. 20, 130, 055, 368, 28 Feb. 2013.

[34] Bhargav-Spantzel, Abhilasha, et al. "Privacy preserving multifactor authentication with biometrics," Journal of Computer Security 15.5 (2007), pp. 529–560.

[35] Aloul, Fadi, Syed Zahidi, and Wassim El-Hajj, "Two factor authentication using mobile phones," AICCSA 2009, IEEE/ACS International Conference on Computer Systems and Applications, 2009, IEEE, 2009.

[36] The Failure of Two-Factor Authentication, (Bruce Schneier, March 2005). Web reference, https://www.schneier.com/blog/archives/2012/02/the_failure_of_2.html

[37] Alzomai, Mohammed, Bander AlFayyadh, and A. Josang, "Display security for online transactions: SMS-based authentication scheme," Internet Technology and Secured Transactions (ICITST), 2010 International Conference.

[38] Liou, Jing-Chiou, and Sujith Bhashyam, "A feasible and cost effective two-factor authentication for online transactions," 2010 2nd International Conference onSoftware Engineering and Data Mining (SEDM), IEEE, 2010.

[39] Arsenault, Ryan, TechPro Essentials, Aberdeen Essentials, The Devil's Dictionary for IT and IT Security: Two-Factor Authentication, Sep 16, 2015, Web reference, http://www.aberdeenessentials.com/techpro-essentials/the-devils-dictionary-for-it-and-it-security-two-factor-authentication/, last accessed on 23 November 2017.

[40] Shaw, Eric; Fischer, Lynn; Rose, Andrée, Insider Risk Evaluation and Audit, 2009, Department of Defense Personnel Security Research Center, http://www.dtic.mil/docs/citations/ADA563910, last accessed on 23 November 2017.

[41] William R. Simpson, 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, pp112-117.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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, San Francisco, USA, 19–21 October 2011, pp. 61–66.
[52] 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, London, U. K., 4-6 July 2012, pp. 524–529.
[53] 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, London, U. K., 4–6 July 2012, pp. 555–560.
[54] 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, San Francisco, USA, 24-26 October 2012, pp. 54–59.
[55] 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.
[56] 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.