A GENERIC AUTHENTICATION LOA DERIVATION MODEL

Li Yao, Ning Zhang

Abstract One way of achieving a more fine-grained access control is to link an authentication level of assurance (LoA) derived from a requester's authentication instance to the authorisation decision made to the requester. To realise this vision, there is a need for designing a LoA derivation model that supports the use and quantification of multiple LoA-effecting attributes, and analyse their composite effect on a given authentication instance. This paper reports the design of such a model, namely a generic LoA derivation model (GEA- LoADM). GEA-LoADM takes into account of multiple authentication attributes along with their relationships, abstracts the composite effect by the multiple attributes into a generic value, authentication LoA, and provides algorithms for the run-time derivation of LoA. The algorithms are tailored to reflect the relationships among the attributes involved in an authentication instance. The model has a number of valuable properties, including flexibility and extensibility; it can be applied to different application contexts and support easy addition of new attributes and removal of obsolete ones.

1 Introduction

In a virtual organisational (VO) environment, services and data are provided and shared among organisations from different administrative domains and protected with dissimilar security policies and mechanisms. These services and data (collectively called resources hereafter) may have varying levels of sensitivity, thus requiring a more fine-grained access control solution. One way of achieving this is to link an authentication level of assurance (LoA) derived from a requester's authentication instance to the authorisation decision made to the requester.

Li Yao
School of Computer Science, University of Manchester e-mail: yaol@cs.man.ac.uk

Ning Zhang
School of Computer Science, University of Manchester e-mail: zhangn@cs.man.ac.uk

Electronic authentication (e-authentication) is an electronic process by which a remote user can be identified. Different authentication methods and processes provide different levels of assurance (LoA) in identifying a remote user. As defined by NIST [2], LoA reflects the degree of confidence in an authentication process used to establish the identity of an entity (an individual or a software component) to whom a credential was issued, and the degree of confidence that the entity using the credential is indeed the entity that the credential was issued to. In other words, LoA is an indicator of the strength of an authentication process. It is influenced by all the factors directly or indirectly associated to the process, including the method used for identity proofing, the authentication protocol/method used by the underlying authentication service and the environment under which the authentication is performed [2, 10, 13]. The extent to which an authentication event is coupled to an authorisation event should also be taken into account when LoA is established.

In a VO, or a large-scale distributed resource sharing environment, resources are likely to be more diversified and have varying levels of sensitivity. The existing approach to access control is a binary approach. A grant or deny authorisation decision is made merely based upon the verification outcome of the requester's identity credential. It is well-known that identity verification cannot always produce a perfect and reliable outcome. This approach to access control, disregarding the quality of authentication in authorisation decision making, cannot satisfy the need for effective and cost-efficient security provision in diversified resource sharing environments. To overcome this limitation, there is a need for the design and development of an adaptive authentication solution that allows the selection of different authentication methods with varying levels of assurance as matched with resource sensitivity levels at run-time.

This paper describes the design of an authentication model, called the generic e-authentication LoA derivation model (GEA-LoADM), to materialise our vision depicted above. The model supports the use and quantification of multiple LoA-effecting attributes in an authentication instance and derives an aggregate LoA for the given set of attributes at run-time. By grouping LoA attributes, analysing their mutual relationships and the composite effect on an authentication outcome, the authentication model is robust and more flexible than the existing binary authentication model. The major novel contributions of this paper include the identification and classification of LoA-effecting attributes (i.e. authentication factors) used in various e-authentication scenarios, the analysis of the mutual relationships and composite effect of these attributes, and the design of LoA derivation algorithms that derives an aggregate LoA for a given set of LoA attributes along with their respective LoA contributions and the mutual relationships.

The rest of this paper is organised as follows. Section 2 discusses related works and efforts on defining and using authentication assurance levels. Section 3 describes, in detail, the design of GEA-LoADM, including its architecture and architectural components. Section 4 presents aggregate authentication LoA derivation algorithms. Section 5 concludes the paper and outlines our future work.

2 Related Works

The concept of authentication LoA has been around since 2000 when the UK Office of the e-Envoy (now the CabinetOffice e-Government Unit) first initiated the effort on defining authentication LoA and on issuing guidance on using some specific types of identification and authentication methods to achieve appropriate levels of assurance so as to ensure that on-line government services are protected properly. This initial effort was then followed up by the US Office of Management and Budget (OMB) that defined e-Authentication guidance for federal agencies [10]. In this guidance, four authentication assurance levels, Levels 1 through to 4, are defined in terms of the consequences of authentication errors and misuse of credentials. The lowest, Level 1, denotes little or no confidence in the validity of an asserted identity, and the highest, Level 4, denotes very high confidence in the asserted identity's validity. While this OMB guidance specifies criteria for determining the authentication assurance levels required for specific on-line services and transactions based upon the risks in each service and transaction category, NIST (US National Institute of Standards and Technology) further defined technical requirements for implementing these four assurance levels in its Special Publication 800-63 [2]. Similar efforts have also been made by the Japanese Government [6], the Australian Government [1], and the Canadian Government [3] as part of their e-government initiatives. These efforts either use, or adopt a similar specification as, the OMB/NIST guidelines mentioned above.

It is worth emphasising that all the LoA guidelines and efforts discussed above are centred on the user-to-system authentication use case scenarios. They do not consider machine-to-machine nor software-to-software authentication scenarios. Nor do they address the authentication of a person via a physical authentication mechanism, e.g. location-based or biometrics based services. In addition, issues related to how LoA may be fed into the authorisation process are also outside the scope of these efforts. There is also a lack of solutions to link LoA to authorisation decision making at run-time. Most of the existing authentication LoA efforts, such as the one recommended by the OMB/NIST, uses an off-line approach to LoA compliance. With this approach, LoA definitions are given as guidelines and the parties concerned are required to comply with these guidelines by conducting a risk assessment of the underlying system, mapping identified risks to an applicable assurance level, selecting appropriate authentication methods and technologies based upon the technical guidelines, and validating the implemented systems to make sure that it has achieved the required assurance level. This off-line approach to authentication assurance level conformance may be adequate for a static and homogeneous environment where resources and their sensitivity levels are pre-defined prior to run-time and the services are provided by a single service provider, such as the case in e-Government scenarios. This approach is certainly not sufficient for Grid computing or large-scale distributed resource sharing environments environments in which both service consumers and service providers are expected to be diversified and dynamic in nature.

The first and the only effort so far (to the authors best knowledge) on linking authentication LoA to authorisation decision making at run-time was made by the FAME-PERMIS (Flexible Authentication Middleware Extensions to the PERMIS) project team [www.fame-permis.org]. The project developed a software component that derives a LoA value based upon a user's authentication token presented to the authentication service, and asserts the value to a role-based access control decision engine run at the SP (Service Provider) side thus achieving LoA lined access control [9, 15]. However, the software is in a very basic form; it only implements the LoA definition versus token types as defined by the NIST guideline [2]. It does not consider the impacts of other LoA- effecting factors such as authentication models and credentials used in Grid applications. Nor does it consider the composite effect by multiple LoA- effecting attributes.

Some works [4, 5] on the estimation of trustworthiness of a user done in the ubiquitous computing community may be relevant to our work described here. However, the algorithms given are largely for the context of a ubiquitous computing environment. For example, [4] proposes a model to calculate the trustworthiness of a user's pervasive device, and [5] describes a parameterised authentication model for calculating the authentication reliability of authentication sensors in a sensor based networks. Both of these works are centred at a broad level of trust in a ubiquitous environment, whereas our work focuses on identifying authentication attributes in large-scale and dynamic distributed resource sharing environments such as Data Grids, analysing and quantifying the composite effect of these attributes on user identification and authentication assurance level and linking it to authorisation decision making at run-time. In our problem context, the design issues of flexibility and extensibility are more acute.

3 GEA-LOADM MODEL

3.1 Architecture overview

As shown in Figure 1, the GEA-LoADM model has a number of architectural components, which can largely be classified into the following groups, an off-line component, a real-time component and a global LoA- effecting attributes policy database (GLoA-APDB). The output of the model is consumed by a replying party (i.e. a service provider) that can be a shibboleth attribute authority [14], or an authorisation decision engine.

The off-line component, called a Global LoA-effecting Attributes Policy Manager (GLoA-APM), is responsible for identifying LoA-effecting attributes and calculating the weightings among additive attributes. It comprises two further functional modules, the Global LoA-effecting Attributes Hierarchical Structure (GLoA-AHS), and the Global LoA-effecting Attributes Weightings Allocation Module (GLoA-AWAM). GLoA-AHS is responsible for identifying all the LoA-effecting attributes in a given authentication context/environment, constructing a hierarchical LoA-effecting attributes structure (such as the one shown in Figure 2), and categorising the attributes into different groups and levels based on their mutual relationships. These tasks are expected to be undertaken manually by an authentication administrator or access policy decision maker based on their security policies and access control requirements. GLoA-AWAM is responsible for calculating LoA weightings for additive attributes (additive LoA attributes refer to those LoA attributes that are in an elevating relationship, i.e. the aggregated LoA value measuring the composite effect of a set of additive LoA attributes on authentication assurance level is not lower than any of the individual component LoA values in the attribute set). The weightings, along with other related information, including the attributes hierarchical structure, the indicators of the relationships among different attributes, component LoA values are all stored in GLoA-APDB. The working mechanisms of, and the methodology used in the design of these functional modules are detailed in Sections 3.2.

The real-time component has two functional modules, a LoA-effecting Attributes Collection Module (LoA-ACM), and an Authentication LoA Derivation Module (ALoA- DM). The LoA-ACM module first receives a notification of the set of contributing LoA-effecting attributes involved in an authentication event/instance from authentication services. It then fetches the component LoA values corresponding to each of the attributes in the attribute set, along with their respective weightings, from GLoA-APDB. Next, LoA-ACM sends the contributing attributes names along with their relationships, component LoA values and weightings to ALoA-DM. Once these parameter values are obtained, ALoA-DM calculates an aggregated LoA using a LoA derivation algorithm corresponding to the settings of this authentication instance. The design details of LoA-ACM and ALoA-DM are described in section 3.4 and 3.5, and the LoA derivation algorithms are discussed in section 4.

GLoA-APDB is a database storing all the LoA-effecting attributes identified by GLoA-AHS, their relationships, component LoA values and additive LoA attributes weightings. The technical details of this module is described in section 3.3. In the

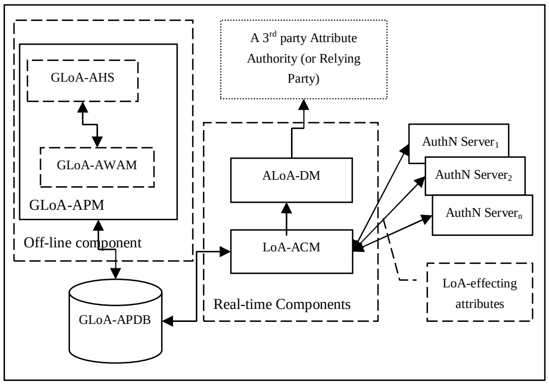

Fig. 1 GEA-LoADM architecture

following subsections, we further describe the designs of the architectural components and how they interact with one another.

3.2 Global LoA-effecting Attributes Policy Manager (GLoA-APM)

GLoA-APM has two functional modules, each performing some well defined tasks. The first module, GLoA-AHS, identifies and classifies LoA effecting attributes and organises them into a hierarchical structure based upon their mutual relationships. The second module, GLoA-AWAM, provides the algorithms that can systematically and scientifically assess and calculate the weightings of additive LoA-effecting attributes for a given authentication model.

Performing these tasks requires a thorough analysis and evaluation of the underlying authentication context/environment and access control policies, which can be a time-consuming process. Therefore, GLoA-APM is also termed as an offline component, meaning that its functional tasks should be performed prior to the execution of authentication procedures.

3.2.1 Global LoA-effecting Attributes Hierarchical Structure (GLoA-AHS)

As mentioned, the GLoA-AHS module is responsible for:

- managing (i.e. adding, deleting and classifying) LoA-effecting attributes;
- assigning component (or attribute) LoA values to each of the attributes; and
- constructing the attributes into a hierarchical structure based on their mutual relationships.

The first two tasks are authentication context dependent. They are also dependent on access policies that are, in turn, influenced by factors such as asset values and the underlying risks in the access environment. We have examined and extended the attributes identified by NIST [2] and OASIS [13], and produced a generic set of LoA-effecting attributes. In addition, we have examined the mutual relationships among these attributes and organised them into a hierarchical structure, as shown in Figure 2. From the figure, it can be seen that the structure highlight the mutual relationship among the group of attributes located at the same level. This structured approach to LoA-effecting attributes' identification, classification, and organisation is an essential step towards the determination of their respective weightings on, and the derivation of, the overall confidence level for an authentication instance, in a scientific manner. This structure has a number of additional merits. For example, it is flexible and extensible. Any emerging LoA-effecting attributes can be easily added into the structure, and any obsolete ones can be removed from it without affecting other levels in the hierarchy. Also, once constructed, a GLoA-AHS instance for a given authentication setting will only need to be revised when there is any change in the authentication attributes at any level.

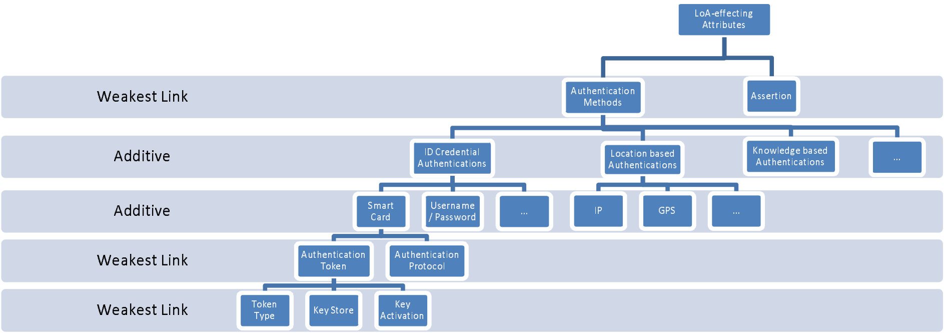

Fig. 2 An exemplar GLoA-AHS structure

3.2.2 LoA-effecting Attributes Weighting Allocation Module (LoA-AWAM)

When calculating an aggregate LoA for a group of attributes that are in an additive relationship, their respective weightings should be determined first. The GLoA-AWAM module uses AHP pair-wise comparison technique [11] to calculate the relative weightings of the attributes. For a group of n additive attributes in the same level, X = x_1, x_2,...,x_n, at a given level in a GLoA-AHS structure, the LoA-AWAM module works as follows [12, 7]:

1. Based on the fundamental scale (developed by [11], and is used to represent the intensity of importance among the attributes), the decision maker inputs the comparison values a_{ij}=x_i/x_j; i,j$\subseteq$ $[1...n]$, where x_i and x_j are the i^{th} and j^{th} attributes in the set, and the algorithm constructs matrix:

$$A = (a_{ij})_{n\times n} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}.$$

2. Compute the principle eigenvalue λ_{max} and the corresponding eigenvector W = $[w_1, w_2, ..., w_n]$.
3. Check for consistency.
4. If matrix A is consistent or acceptably consistent, the algorithm derives the normalised eigenvector W'=$[w'_1, w'_2, ..., w'_n]$ from W, and W' is the normalised weight for the set of attributes X.
5. Repeat steps (1)-(4) above for every attribute groups located at additive levels in the GLoA-AHS hierarchy.

3.3 Global LoA-effecting Attributes Policy Database (GLoA-APDB)

GLoA-APDB is a database containing three tables, storing, respectively, the GLoA-AHS data structure, the LoA-effecting attributes along with their component LoA values and weightings (for additive attributes), and aggregate LoA values and the corresponding information in the case of successful LoA derivation for an authentication event. The table for storing the GLoA-AHS data is called the Hierarchy Table. The second table, called the Attribute Table, stores AttributeNames, ComponentLoAValues, Weightings and RelationshipTypes of the LoA effecting attributes. These two tables store all the information required by the GLoA-APM module. The third table is named as the Aggregated LoA Table and it is for logging LoA information related to authentication events. That is, if an authentication event is successful, the Table stores the aggregated LoA value calculated for the event along with the corresponding contributing LoA-effecting attributes.

3.4 LoA-effecting Attributes Collection Module (LoA-ACM)

The LoA-ACM module performs three tasks. Firstly, it interacts with all the authentication services involved in an authentication event to identify contributing LoA-effecting attributes. Secondly, it queries GLoA-APDB to obtain the component LoA values and weightings of the attributes. Thirdly, it sends all the data fetched from GLoA-APDB to ALoA-DM that then derives the aggregated authentication LoA value for the event.

3.5 Authentication LoA Derivation Module (ALoA-DM)

ALoA-DM receives a set of LoA-effecting attributes along with their component LoA values and weightings for an authentication event from LoA-ACM and derives an aggregated authentication LoA value for the event. The derivation is done by using either of the two algorithms detailed in Section 4. Once the aggregated LoA value is calculated, the LoA-effecting attributes along with the aggregated LoA value will be stored in GLoA-APDB for auditing purposes and for future references. Optionally, these data may be stored in a third party attribute directory or an attribute authority for consumptions by other relying parties. For example, the data may be sent to the attribute authority in the Shibboleth system for attribute assertion [14, 15], or to the attribute authority for creating and assigning an attribute certificate.

4 ESTIMATING THE COMPOSITE EFFECT OF MULTIPLE LOA-EFFECTING ATTRIBUTES

4.1 The Method Overview

As discussed in section 3, for any given authentication system, there will be a set of multiple LoA-effecting attributes, and the attributes can be organised into a GLoA-AHS structure. Using the structure, we can estimate the composite effect (i.e. aggregated LoA) of these attributes. This is done in a bottom-up manner. Assuming that there are *m* levels (levels 1, ..., *m*) in the structure. From the bottom level *m*, based upon the relationship (the weakest link, or the additive) of the attributes at the level, an aggregated LoA derivation algorithm (corresponding to the relationship) is used to calculate the aggregated LoA for this level. This aggregated LoA value is then used as the component LoA of the connected attribute at the level immediately above, i.e. Level (*m*-1). This process continues until the top level, i.e. Level 1, of the structure is reached, and the aggregated LoA value at Level 1 is the overall confidence level, i.e. the aggregated LoA, for the entire authentication event.

Obviously, for different relationships among multiple attributes, different LoA derivation algorithms should be used. The following two subsections discuss the weakest link relationship algorithm and additive relationship algorithm respectively.

4.2 The $ALoA_{WL}$ Algorithm

The $ALoA_{WL}$ (Aggregated LoA for the Weakest Link relationship) algorithm discussed in this section is designed for estimating an aggregated LoA value given a set of attributes that are in the weakest link relationship. Assume that there is a group of attributes $\{a_1, a_2, ..., a_n\}$ at level *k* and their respective component LoA values are $\{LoA_{a1}, LoA_{a2}, ..., LoA_{an}\}$, and that these attributes are in the weakest link relationship. The composite effect of these attributes on the authentication assurance level should be the lowest component LoA value in the set. Mathematically, this can be expressed as:

$$ALOA_{(WL,level-k)} = min(LoA_{a1}, LoA_{a2}, ..., LoA_{an});, \qquad (1)$$

where *min* is the minimum function, and $ALOA_{(WL,level-k)}$ is the aggregated LoA value for level k with attributes in the weakest link relationship.

From this discussion, it can be seen that the derivation of an aggregated LoA value for a group of attributes that are in the weakest link relationship only requires the attributes component LoA values.

4.3 The $ALoA_{AD}$ Algorithm

The design of the $ALoA_{AD}$ (Aggregated LoA for the additive relationship) algorithm that is required for estimating an aggregated LoA value given a set of attributes that are in an additive relationship is not as straightforward as the case for $ALoA_{WL}$. A scientific method that can take into account of the attributes' component LoA values as well as their respective weightings is required. Subjective Logic [8], defined to mathematically describe and quantify subjective beliefs, consists of a belief model named opinion model and set of operations for combining opinions. It can be used to define various operations for processing multiple opinions such as conjunction, disjunction, negation, consensus, recommendation and ordering. The $ALoA_{AD}$ algorithm employs the subjective logic opinion (SLO) model and its consensus operation to derive the aggregated LoA [8].

Using the SLO model, each of the additive attributes is transformed into an 'opinion' in the opinion model. For example, an attribute x's opinion about the aggregated authentication assurance level can be expressed as,

$$\pi_p^x = b + d + u = 1, b, d, u \subseteq [0,1] \tag{2}$$

Where π is the opinion function, p is the proposition which π has opinion to (in this case, p refers to the aggregated LoA), x is the attribute, and b, d, and u represent belief, disbelief and uncertainty, respectively.

We now need to determine the values for tuple $< b, d, u >$. Belief b refers to the level of trust in attribute x's opinion. It is set to a value in the range [0,1], where 0 stands for no certainty and 1 stands for absolute certainty. The level of trust in an authentication outcome (i.e. the meaning of b) obviously has a similar meaning as the component LoA (which refers to the level of confidence in an authentication outcome). However, as LoA values are scoped between 1 to 4, and b in the subjective logic uses a scale from 0 to 1, we need a transform method to transform LoA values from the scale of [1, 4] to values in the scale of [0, 1]. This scale transformation is done using the following mapping, b(0.25) = LoA_x(1), b(0.5) = LoA_x(2), b(0.75) = LoA_x(3), and b(1) = LoA_x(4).

Disbelief d refers to the level of accuracy in attribute x's opinion. It is usually used to measure the accuracy of some hardware-based authentication attributes such as the case in biometric authentication and hardware sensor- based authentication [5]. Unlike hardware-based authentication attributes, credential-based authentication attributes only have belief and uncertainty values, but not accuracy value. This is because, for credential based authentication, if the authentication outcome is successful, then the level of accuracy is taken as 100% (i.e. d = 0).

Based upon these considerations, for credential-based authentication attributes, we can define the opinion for attribute x as follows:

$$\pi_p^x = \{b + d + u\} = \begin{cases} b = LoA_x \\ d = 0 \\ u = 1 - LoA_x \end{cases}$$

The opinion definitions for cases where disbelief is not zero, such as the case of sensor or location based authentication method, will be addressed in our future work. Once the opinions of all the attributes involved are defined, we can calculate a combined opinion by using the consensus operation defined in [8].

This consensus operation assumes that the contributions (i.e. the weightings) by each opinion are the same. However, for different authentication events and in different access environments, the weightings of different additive LoA- effecting attributes are likely to be different, and these differences may influence the final LoA derivation result significantly. For example, consider the case where a smartcard authentication attribute with a component LoA value 3, and an IP authentication attribute with a component LoA value 1, are both used in an authentication event. If the ratio of their authentication impact/weighting is (1:1), then the calculated combined opinion will be {b=0.77, d=0, u=0.33}. However, if the ratio is (3:1), then the combined opinion will be different and b is expected to be higher than 0.77. Therefore, there is a need for a method to integrate the influence of various weightings into the algorithm. We do this by integrating the weighting of an attribute into its component LoA. The following describes this method.

Assume that w_i is the weighting, LoA_i is the original component LoA value, and LoA_{ai} is the adjusted LoA value, of attribute a_i. In other words, the effect of a_i's weighting on the final aggregated LoA is embedded into the adjusted component LoA value of attribute a_i, LoA_{ai}. It is worth noting that the sum of the weightings by all the attributes is always 1. Assume there are n attributes, if we take that the assumed contributions (or assumed weightings) by each of the attributes are always the same, and that each such weighting equals to 1/n, then the adjusted weighting for attribute a_i will be the difference between the real weighting, w_i, and the assumed weighting, 1/n. That is, the adjusted component LoA value for attribute a_i is

$$LoA_{ai} = LoA_i \times (1 + (w_i - 1/n)) \tag{3}$$

By integrating attributes' weightings into their respective component LoA values, the adjusted component LoA values can capture the effects of the attributes on the overall authentication assurance level of an authentication event in a more accurate manner. Then the consensus operation mentioned earlier can be used to derive the final aggregated LoA value.

5 CONCLUSION AND FUTURE WORK

This paper has discussed the concept of authentication level of assurance and the potential benefits in using it to achieve more fine grained access control. However, owing to the number, the variety and the complexity of the attributes concerned, quantifying their composite effect and deriving an aggregate assurance level given multiple authentication attributes for an authentication event is a very challenging research issue.

The paper has made some novel contributions in addressing this research issue by proposing a framework, by which an authentication assurance level as influenced by multiple attributes can be systematically estimated. This framework includes a Global LoA-effecting Attributes Hierarchical Structure (GLoA-AHS) by which a large number of LoA- effecting attributes can be organised into a hierarchical structure with distinctive mutual relationships. Two aggregated LoA derivation algorithms are designed to accommodate the identified relationships. With the use of these algorithms, along with the GLoA-AHS structure and additional architectural component, the framework is able to automatically derive a composite LoA value given a set of LoA-effecting attributes. The major advantage of this model is its ability to accommodate a complex set of attributes, and to provide a quantitative measure for authentication assurance levels in the face of the complex attributes. Our ongoing work includes prototyping and evaluating the framework, and extending it to accommodate more complex Grid authentication scenarios. The consequent data privacy protection is another research issue and how to safely employ users authentication information without misuse will be included in our future work.

References

1. Australian e-Government & Information Management Available at: http://www.finance.gov.au/e-government/index.html. cited 10 Oct 2008
2. Burr, W. E., et al.: Electronic Authentication Guideline. In: NIST Special Publication 800-63. Available via NIST. http://csrc.nist.gov/publications/PubsSPs.html cited 15 Oct 2008
3. Canadian e-authenticaiton, 2004 Available at: http://www.ic.gc.ca/epic/site/ecic-ceac.nsf/en/h_gv00090e.html. cited 10 Oct 2008
4. Creese, S., et al., Authentication for Pervasive Computing, the First International Conference on Security in Pervasive Computing, 2003, pp.116-129, Boppard, Germany, 2004.
5. Covington, J., et al, Parameterized Authentication, European Symposium on Research in Computer Security 2004, pp. 276C292, Sophia Antipolis, French Riviera, France 2004.
6. Japan, An overview of International Initiatives in the field of Electronic Authentication, 2005 Available at: http://www.japanpkiforum.jp/shiryou/e-auth_policy/overview_e-auth_ v07.pdf. cited 10 Oct 2008
7. Johnson, H. et al, A Decision System for Adequate Authentication, S.F.Page(s): 185- 185 Digital Object Identifier 10.1109/ICNICONSMCL.2006.9
8. Josang, A. et al. Legal Reasoning with Subjective Logic. Artificial Intelligence and Law, 8(4), pp.289-315, Kluwer 2000.
9. Nenadic A. et al., Fame: Adding Multi-Level Authentication to Shibboleth, IEEE Conference of E-Science and Grid Computing, Page(s):157 - 157, Amsterdam, Holland, 2006.
10. OMB Memorandum M-04-04, E-Authentication Guidance for Federal agencies. Available via OMB. http://www.whitehouse.gov/OMB/memoranda/fy04/m04-04.pdf. cited 10 Oct 2008
11. Saaty, T.L., ”Scaling method for priorities in hierarchical structures”, Journal of Mathematical Psychology 15/3 (1977) 234-281.
12. Saaty, T.L., How to make a decision: The analytic hierarchy process. European Journal of Operational Research. No: IC/1990/48, pp. 9-26, 1990.
13. SAML 2.0 Authentication Context specification Available via OASIS. http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf. cited 10 Oct 2008
14. Shibboleth Architecture technical overview, 2005, available at: http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-overview-latest.pdf.
15. Zhang, N., Yao, L. et al., doi: 10.1002/cpe.v19:9.