TOWARDS USER AUTHENTICATION FLEXIBILITY

Laurent Gomez
SAP Research, SAP Labs France, 06250 Mougins, France

Ivonne Thomas
Hasso-Plattner-Institute, University of Potsdam, D-14440 Potsdam, Germany

Keywords: Access Control, Authentication, Subjective Logic.

Abstract: In order to gain access to a resource protected by an authorization service, a user can be required to authenticate. Traditionally, user authentication is performed by means of a combination of authentication factors, statically specified in the access control policy of the authorization service. In this paper, we propose to improve the flexibility of user authentication by enabling to authenticate using authentication factors at his disposal. Authentication factor are any piece of information used to assess the identity of a user. Capitalizing on opinion metric from subjective logic (Josang, 2001), the authorization service specifies an authentication level to be reached in order to gain access to a resource.

1 INTRODUCTION

In order to gain access to a resource protected by an authorization service, users can be required to authenticate. Traditionally, user authentication is performed by means of a combination of authentication factors (e.g. two-factor authentication (Schneier, 2005)) statically specified in the access control policy of the authorization service. Authentication factor are meant as any piece of information used to assess the identity of a user. In this paper, we propose to improve the flexibility of user's authentication by enabling to authenticate using different authentication factors at his disposal. Depending on his context, the user may have access to different authentication services. To that effect, the authorization service specifies an authentication level to be reached in order to get access to a resource. Resource owner's authentication preference are thus comprised in an authentication level policy.

The remainder of the paper is organized as follows. In section 2, we outline our approach. Section 3 introduces a new operator to combine authentication level combination and specifies authentication level policy. In section 4, we propose an access control model leveraging the concept of authentication level. We explain how a user can satisfy a required authentication level by means of his available authentication services. Related work is discussed in section 5 and section 6 presents the conclusion.

2 STATEMENT OF GOALS

In this section, we outline our approach to improve the flexibility of users authentication. In this approach, users are responsible for choosing the proper authentication factors to meet authentication requirements defined by the authorization service.

We envisioned the following authentication process, as depicted in Figure 1: (1) A user wants to gain access to a resource protected by an authorization service. The authorization service responds to the user with an obligation stating an authentication level to be reached. (2) The user attempts to reach the expected authentication level by means of combining authentication factors, using available authentication services, at his disposal. (3) The user then forwards a combination of authentication factors acquired to the authorization service, which then checks if they meet the required authentication level.

In order to simplify user authentication, we hence define three goals for a solution: (i) authentication level specification can be done by resource owner, (ii)

Gomez L. and Thomas I. (2007).
TOWARDS USER AUTHENTICATION FLEXIBILITY.
In *Proceedings of the Second International Conference on Security and Cryptography*, pages 59-66
DOI: 10.5220/0002122400590066

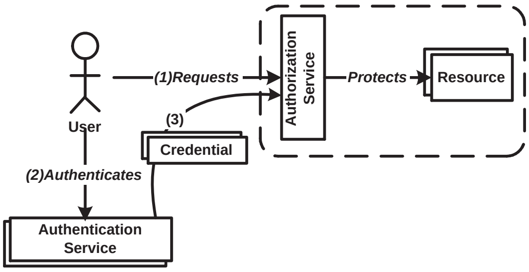


Figure 1: Architecture Overview.

the authentication level specified can be met by legitimate users and (iii) enforcement of an access control can be done based on a specified authentication level. Regarding authentication level specification, a resource owner should be able to define first their preferences of authentication factors, as well as the authentication level required to get access to their resource. Finally, the user should be able to determine if they can satisfy the required authentication level with a combination of available authentication factors.

2.1 Methodology

Each authentication factor is associated with an authentication level. The latter is the mapping of authentication factors to confidence values. Resource owner may specify their preferences in authentication factors by means of authentication level. The user is then able to combine available authentication factors in order to reach the expected authentication level.

Following our approach, a suitable metric for authentication levels is required. For this purpose we apply subjective logic (Josang, 2001), which supports the assignment of a confidence value to properties such as authentication factors. In addition, it is sufficiently extensible to allow us to define a new operator for authentication level combination. Following this foundation, based on a resource owner's confidence in authentication factors, we are able to define an authentication level policy which maps authentication levels to authentication factors. Furthermore, relying on authentication level requirements to resources, we specify an access control policy featuring an authorization service level. Finally, we define a procedure so that a user can satisfy the authentication level requirements with a combination of authentication factors, relying on his available authentication services and the defined authentication level policy.

2.2 Subjective Logic

Subjective logic is a theoretical framework based on Dempster-Shafer theory (Shafer, 1976). In subjective logic, we manipulate opinions about a proposition P. An opinion is represented by the 4-tuple (b,d,u,a). a represents the *a priori* probability of P to be true in absence of opinion. As we only consider binary state space for P, we set a to 1/2. b, d and u represent the belief that P is true, the belief that P is false, and the uncertainty is the amount of belief that is not committed to the truth or falseness of P's respectively. The range of those four values is [0,1] where b+d+u=1. The opinion of a subject A about a proposition P is defined as $\omega_P^A = b + au$.

Moreover, the subjective logic framework provides a set of logical operators for combining opinions. Subjective logic provides traditional operators such as conjunction, disjunction and negation which corresponds to AND, OR and NOT logical operators between propositions. Subjective logic supports also non-traditional operators such as average or discount of opinions (Josang, 2001).

3 AUTHENTICATION LEVEL

In this section, we explain how we capitalize on subjective logic in order to define and combine authentication levels.

3.1 Authentication Factor

As depicted in figure 2, an authentication factor is delivered by an authentication service which implements an authentication mechanism. Each authentication mechanism is rated, based on some intrinsic characteristics called criterion. For example, password authentication can be characterised by the password length.

Traditionally, existing authentication factors are divided in three categories: what a user knows (e.g. password), what a user has (e.g. credentials), and what he is (e.g. biometry) (Pfleeger, 1997). In figure 3, we illustrate an upper view of authentication mechanisms classification with three classes: token-based (e.g. X.509 certificate, Kerberos (J.T Kohl, 1994) ticket), knowledge-based (e.g. password) and biometry (e.g. iris, finger print). This classification is not exhaustive and can be extended to other authentication mechanisms classes (e.g. time-based authentication). In figure 4, we depict text-based authentication mechanism which is a subset of token-based authentication mechanisms. Moreover, this figure shows criterion identified for some text-based authentication mechanisms. The reason for adding criterion on authentication mechanisms is to instantiate them in a fine-grained matter. For example, it enables us to clearly differentiate two authentication factors (e.g. X.509 certificate signed by ABC and DEF) by means of authentication mechanism criterion (e.g. signature issuer).

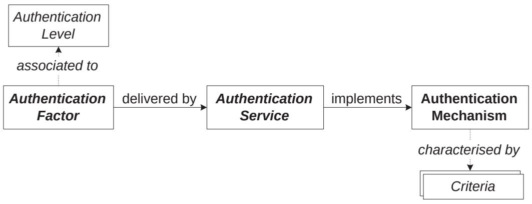


Figure 2: Authentication Factor.

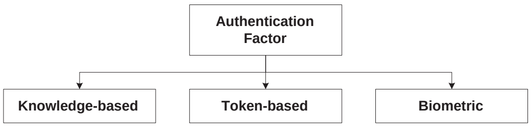


Figure 3: Authentication Classification.

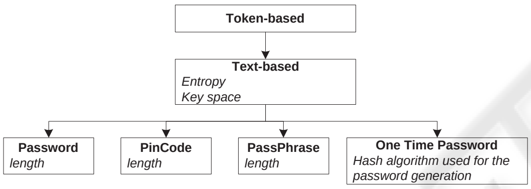


Figure 4: Text-Based Classification.

3.2 Opinion Determination

Resource owners determinate their opinions on authentication services and mechanisms. Following Covington et al's (M. Covington, 2004) approach, we distinguish subjective aspect (e.g. reputation (S Ganeriwal, 2004) on authentication service and mechanism) from concrete aspects (e.g authentication mechanism criterion). The subjective aspects of an opinion are based on the past experience with a given authentication service or mechanism, while the concrete aspects are derived from measurable elements which characterize an authentication service or mechanism. Table 1 proposes few subjective and concrete aspects for opinion determination of authentication service and mechanism.

For the sake of readability, s denotes a subjective aspect, and c a concrete aspect. In (M. Covington, 2004), the authors propose the following combination for determining an opinion ω based on those two parameters:

Table 1: Opinion Determination.

	Subjective	*Concrete*
Service	Reputation Trust	Quality of Service Domain
Mechanism	Reputation Trust	criterion

$$\omega = (b,d,u,a) \ where \begin{cases} b = s \cdot c \\ d = s \cdot (1-c) \\ u = 1-s \end{cases}$$

Belief is then defined as a combination of subjective and concrete aspects whereas uncertainty is defined as the opposite of subjective aspect. Based on this combination of subjective and concrete aspects, resource owners can then determine their opinion in authentication services and mechanisms.

3.3 Authentication Level of a Single Authentication Factor

We first consider authentication level of a single authentication factor. As described in section 3.1, an authentication factor is delivered by an authentication service which implements an authentication mechanism. To determine authentication level of an authentication factor, we propose to combine its associated authentication service and authentication mechanism opinions. Thus we consider two opinions $\omega_{P_{as}}$ and $\omega_{P_{am}}$ where:

- P_{as}="The authentication service is trustworthy enough to be used for authorization".
- P_{am}="The authentication mechanisms is trustworthy enough to be used for authorization".

In order to calculate combined opinion on authentication service and mechanism, we propose to define a new combination operator:

$$\omega_{P_{af}} = \omega_{combine}(\omega_{P_{as}}, \omega_{P_{am}})$$

where P_{af} is the proposition "The authentication factor is reliable enough to be used for authorization".

The combination operator aims at leveraging the influence of the best opinion between $\omega_{P_{as}}$ and $\omega_{P_{am}}$. It consists of a smooth increase of the maximum between the two opinions, depending on the distance $|\omega_{P_{as}} - \omega_{P_{am}}|$. The next section is dedicated to the definition of this combination operator.

3.4 Combination Operator

Regarding combination of authentication level, we tend to always increase combined authentication level

$\omega_{combine(\omega_a,\omega_b)}$ of two authentication levels ω_a and ω_b. This increase must be proportional to their maximum and to their distance. Thus, the combination operator has to fulfill the following requirements:

- **RE1**: $\omega_{combine}(\omega_a,\omega_b) \geq max(\omega_a,\omega_b)$.
- **RE2**: The more $|\omega_a - \omega_b|$ tends to zero, the more $\omega_{combine}(\omega_a,\omega_b)$ increases.
- **RE3**: $\omega_{combine}(\omega_a,\omega_b)$ is proportional to $\max(\omega_a,\omega_b)$.

With requirement **RE1**, we express the fact that the combination of two opinions always results in an increase of opinion. In case of $\min(\omega_a,\omega_b)$=0, $\omega_{combine}(\omega_a,\omega_b)$ is equal to the lower bound, $\max(\omega_a,\omega_b)$. **RE2** reflects the fact that the closer the $\min(\omega_a,\omega_b)$ is to $\max(\omega_a,\omega_b)$, the bigger the combination acceleration of ω_a and ω_b has to be. Finally, **RE3** specifies that the result of the combination is bounded by $\max(\omega_a,\omega_b)$.

Based on **RE1**, **RE2** and **RE3**, we define combination between two authentication levels as follows:

Let ω_a and ω_b be agent's opinion about two distinct propositions a and b. Let $\omega_{combine}(\omega_a,\omega_b)$ be the opinion such that :

$$\omega_{combine}(\omega_a,\omega_b) = \min(1, \max(\omega_a,\omega_b) + \varepsilon(\omega_a,\omega_b))$$
$$\textit{where } \varepsilon(\omega_a,\omega_b) = (\omega_a \cdot \omega_b)^{(2-\omega_a-\omega_b)}.$$

$\omega_{combine}(\omega_a,\omega_b)$ is called the combination of ω_a and ω_b representing the agents' opinion about the combination of a and b being true.

RE1 is fulfilled by the fact that $\varepsilon(\omega_a,\omega_b)$ is at least equal to 0 as ω_a and ω_b are in [0:1].

RE2 is fulfilled by the fact that $(2 - \omega_a - \omega_b)$ decreases while $\min(\omega_a,\omega_b)$ increases for a given $\max(\omega_a,\omega_b)$. This implies that the acceleration of $\varepsilon(\omega_a,\omega_b)$ increases.

RE3 is fulfilled by the fact that $\omega_{combine}(\omega_a,\omega_b)$ is a function of $max(\omega_a,\omega_b)$.

In figure 6, we depict the evolution of $\omega_{combine}(\omega_a,\omega_b)$ for five values of the maximum between ω_a and ω_b. We clearly demonstrate our three requirements. When the maximum between the opinions equals to 0.1, the increase of $\omega_{combine}(\omega_a,\omega_b)$ is smaller than the one with the maximum equals to 0.9 (**RE1** and **RE3**). Moreover $\omega_{combine}(\omega_a,\omega_b)$ increases progressively while $|\omega_a - \omega_b|$ tends to zero (**RE2**).

Figure 5 depicts the fact that a combination of two low opinions leads to almost no opinion increases. On the contrary, the combination with two strong opinions leads to a quick increase of combination.

$\omega_{mitigate}(\omega_a,\omega_b)$		ω_a		
		Low	**Medium**	**High**
ω_b	**Low**	Low	Medium	High
	Medium	Medium	Medium	High
	High	High	High	High

Figure 5: Combination Evolution.

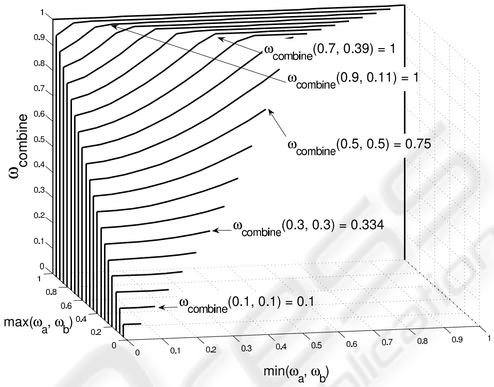


Figure 6: Combination Operator.

3.5 Authentication Level Combination

As far as the combination of authentication factors is concerned, we compute the authentication level of a combination of n authentication factors $\omega_{P_{caf}}$ as follows:

$$\omega_{P_{caf}} = \omega_{combine}(\omega_{P_{af_i}})_{i=0}^{n\geq 1} \qquad (1)$$

where

- The authorization service has an authentication level of $\omega_{P_{af_i}}$ on each authentication factor.
- $\omega_{combine}(\omega_{P_{af_i}})_{i=0}^{0} = \omega_{P_{af_0}}$
- $\omega_{combine}(\omega_{P_{af_i}})_{i=0}^{1} = \omega_{combine}(\omega_{P_{af_0}}, \omega_{P_{af_1}})$
- $\omega_{combine}(\omega_{P_{af_i}})_{i=0}^{n\geq 2} = \omega_{combine}(\omega_{combine}(\omega_{P_{af_j}})_{j=0}^{n-2}, \omega_{combine}(\omega_{P_{af_{n-1}}}, \omega_{P_{af_n}}))$

3.6 Authentication Level Policy

Resource owners specify their opinions on authentication services and mechanisms, $\omega_{P_{as}}$ and $\omega_{P_{am}}$ in the authentication level policy. The latter thus is composed of two sets

- { Authentication Service Description, $\omega_{P_{as}}$ } and
- { Authentication Mechanism Description, { Criterion Description, $\omega_{P_{am}}$ }*, $\omega_{P_{am}}$ }.

In figure 7, we show a simple authentication level policy based on XML. For the sake of readability, we skip intentionally namespaces and full descriptions of

```
<AuthenticationLevelPolicy>
 <AuthenticationServices>
  <AuthService URL="ABC"
    AuthMecha=’’X.509 Certificate’’>
   <Opinion>0,5</Opinion>
  </AuthService>
 <AuthenticationServices>
 <AuthenticationMechanisms>
  <AuthenticationMechanism
    Type=’’X.509 Certificate’’>
   <Opinion>0,3</Opinion>
   <Criterion="IssuedByFooBar">
    <Opinion>0,35</Opinion>
   </Criterion>
  </AuthenticationMechanism>
 <AuthenticationMechanisms>
</AuthenticationLevelPolicy>
```

Figure 7: Authentication Level Policy Sample.

```
<Policy>
 <Target>...
 </Target>
 <Rule Effect=’’Permit’’>
  <Target>
   <Subjects>Physician</Subjects>
   <Resources>Medical Data</Resources>
    <Actions>Read</Action>
   </Actions>
   </Target>
    <Condition>
         <Apply Fct=’’required_auth_lvl’’>
      <AuthLevel>0,75</AuthLevel>
     </Apply>
    </Condition>
  </Rule>
</Policy>
```

Figure 8: Access Control Example.

authentication services, mechanisms and criteria. We refer to an authentication service with an unique URL. An opinion in an authentication mechanism can be refined by means of authentication mechanisms criteria. In our example, we specify an opinion of 0.3 in an X.509 certificate, which is refined with the introduction of a criterion on an X.509 certificate such as issuer.

4 ACCESS CONTROL POLICY

In section 3, we have described how resource owners can define their opinions on authentication services and mechanisms in authentication level policy, relying on subjective logic. At authorization service level, we enforce an access control policy based on the preferences of resource owners. The purpose of our access control policy is to specify the required authentication level to get access to a resource. To that effect, requesters should be able to satisfy the authentication level requirement.

We first describe an access control policy based on XACML (OASIS, 2005) to specify resource owners' security preferences. We then propose an approach to satisfy an required authentication level by resource requesters. Finally, we enforce access control policy based on resource owners opinions on authentication factors.

4.1 Access Control Policy Definition

In order to avoid the burden of defining a new security policy language, we decided to reuse an existing one that will enable us to express our requirements related to authentication levels. XACML (eXtensible Access Control Markup Language) appears to be the most appropriate security policy language. It is an OASIS standard used to perform access control. It includes an XML-like policy language and a query language for access control enforcement and decision. XACML requests consist of a triple { *Subject*, *Resource*, *Action* }. A *Subject* tries to access to a *Resource* (e.g. file, web service) in order to perform an *Action* (e.g. read/write, invoke a method). The *Subject* is characterized by a set of attributes (e.g role, location). Based on this triple { *Subject*, *Resource*, *Action* }, a rule-based access control policy is enforced.

In the XACML policy definition, for a given 3-tuple { *Subject*, *Resource*, *Action* }, we add a rule definition where we define the required authentication level. The XACML policy sample in figure 8 illustrates the definition of an authentication level expectation for physicians who want to get access to patient medical information. The physicians have to authentication themselves with an authentication level up to 0.75.

When users request access to a protected resource, the authorization service has to extract the authentication level expectation for the given resource. Based on the authentication level policy of the authorization service, requesters try to achieve the authentication level requirement.

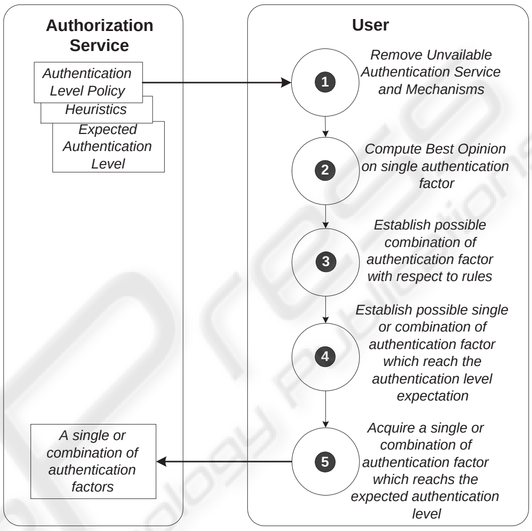
4.2 Meeting the Expected Authentication Level

In order to meet the required authentication level, users have to combine authentication factors which reach the required authentication level. To that effect, users have first to establish a list of potential combination of authentication factors. In this step, users determine the authentication services which would enable them to acquire authentication factors reaching the authentication level expectation. Second, users have to acquire authentication factors from authentication services. With acquisition of authentication factors, we always refer to the process of users authentication and receiving an security token. Finally, users have to check if the combination of acquired authentication factors actually reach the required authentication level.

Figure 9 depicts our approach to satisfy the authentication level requirements. In the first step, we propose to reduce the set of possible combinations of authentication factors without authenticating the user.

- **Step 1.**:
  From the authentication level policy, we keep only authentication services at requester's disposal. It remains a list of available authentication services and mechanisms for the requester.
- **Step 2.**:
  We compute the best opinion for each potential authentication factor. Without taking into account the criteria on authentication mechanisms, we extract for each authentication mechanisms the best reachable authentication level. We then combine the opinion on authentication services and the best opinion achievable out of its authentication mechanism.
- **Step 3.**:
  In addition to the authentication level policy and authentication level expectation, the authentication service imposes to the user a set of rules on combination of authentication factors. The goal of such rules is to prevent users to over combine authentication factors (e.g. *"no combination should consist of more than 3 authentication factors"*, *"each authentication factor must be delivered by different authentication services"*). The requester can remove non-compliant combination with those rules.
- **Step 4.**:
  Finally, we remove any combination of authentication factors that does not reach the authentication level.

If there is no possible combination of authentication factor remains, the requester can obviously not meet the authentication level requirement and the access to the requested resource is denied. In case of remaining combinations, the requester has to acquired each authentication factor by performing the authentication process (5). Each authentication factor is characterised by an authentication mechanism criteria which determine its final authentication level. The latter has to reach the authentication level requirement. For the sake of optimisation, we suggest that the requester caches temporary the acquired authentication factors. If the requester finally finds a combination of authentication factors which satisfies the authentication level expectation, she then forwards it to the authorization service.



Authorization Service
Authentication Level Policy
Heuristics
Expected Authentication Level
A single or combination of authentication factors
User
1
Remove Unvailable Authentication Service and Mechanisms
2
Compute Best Opinion on single authentication factor
3
Establish possible combination of authentication factor with respect to rules
4
Establish possible single or combination of authentication factor which reach the authentication level expectation
5
Acquire a single or combination of authentication factor which reachs the expected authentication level

Figure 9: Authentication Level Satisfaction.

4.3 Example of Authentication Level Satisfaction

In the following example, a user has two authentication services S_1 and S_2 which support M_1 and M_2 authentication mechanisms respectively. The authorization service defines an authentication level policy for S_1, S_2 and S_3 authentication services which support M_1, M_2 and M_3 respectively. M_1 may be characterised with two criteria C_{11} and C_{12}. Figure 10 assigns an authentication level to each authentication service and mechanism. In addition, the required authentication level is set to 0.6 and the only rule on combination of authentication factors is : *"Only one authentication factor per available authentication service"*.

- Step 1.:
  We remove all the unavailable authentication services and mechanisms from the authentication level policy as follows. Only S_1 and S_2 services, which refer to mechanisms M_1 and M_2 respectively, remains .
- Step 2.:
  We compute the best combination opinion out

```
<AuthenticationLevelPolicy>
 <AuthenticationServices>
  <AuthService URL="S1" AuthMecha=’’M1’’>
   <Opinion>0,5</Opinion>
  </AuthService>
  <AuthService URL="S2" AuthMecha=’’M2’’>
   <Opinion>0,4</Opinion>
  </AuthService>
  <AuthService URL="S3" AuthMecha=’’M3’’>
   <Opinion>0,7</Opinion>
  </AuthService>
 <AuthenticationServices>
 <AuthenticationMechanisms>
  <AuthenticationMechanism Type=’’M1’’>
   <Opinion>0,2</Opinion>
   <Criterion="C11">
    <Opinion>0,3</Opinion>
   </Criterion>
   <Criterion="C12">
    <Opinion>0,5</Opinion>
   </Criterion>
  </AuthenticationMechanism>
  <AuthenticationMechanism Type=’’M2’’>
   <Opinion>0,1</Opinion>
  </AuthenticationMechanism>
  <AuthenticationMechanism Type=’’M3’’>
   <Opinion>0,3</Opinion>
  </AuthenticationMechanism>
 <AuthenticationMechanisms>
</AuthenticationLevelPolicy>
```

Figure 10: Authentication Level Policy Example.

of each authentication service and its supported mechanism.

(S_1, C_{12})	$\omega_{combine}(\omega_{S_1}, \omega_{C_{12}})$ = 0.75
(S_2, M_2)	$\omega_{combine}(\omega_{S_2}, \omega_{M_2})$ = 0.4

- Step 3.:
  Based on the combination rule, we have the following combinations of authentication factors:

  $$\{(S_1,C_{12}); (S_2,M_2), ((S_1,C_{12});(S_2,M_2))\}.$$

- Step 4.:
  The following table corresponds to the best authentication level reachable by each combination of authentication factors:

(S_1, C_{12})	$\omega_{combine}(\omega_{S_1}, \omega_{C_{12}})$ = 0.75
(S_2, M_2)	$\omega_{combine}(\omega_{S_2}, \omega_{M_2})$ = 0.4
$((S_1,C_{12});$ $(S_2,M_2))$	$\omega_{combine}$ $(\omega_{combine}(\omega_{S_1}, \omega_{C_{12}}),$ $\omega_{combine}(\omega_{S_2}, \omega_{M_2}))$=1

  As the second entry does not reach the expected authentication level set to 0.6, we remove it.

- Step 5.:
  The user acquires authentication factor from S_1. The mechanism supported by S_1 is characterised by C_{11}. The corresponding authentication level is then $\omega_{combine}(\omega_{S_1}, \omega_{C_{11}})$=0.6. The user can send back the authentication factor to the authorization service.

4.4 Access Control Policy Enforcement

At the access control policy enforcement point, we consider two steps: validation of authentication factors and enforcement of access control policy.

Validation of an authentication factor consists of checking its timestamp or its signature. If any authentication factor is expired, signature is not valid, the combination of authentication factors is considered as invalid and access is denied.

Finally, the enforcement of access control policy is in charge of computing the authentication level associated to a combination of delivered authentication factors with respect to the defined authentication level policy. The computed authentication level must reach the authentication level requirement.

5 RELATED WORK

In the literature, several researchers have already proposed models for authentication factors metric. In (M.K Reither, 1999), the authors propose a set of principles for designing a metric for authentication factors. Nevertheless, they only focus on issuer of authentication factor and not on the authentication mechanism supported. In (W. Burr, 2006), an assurance level on authentication factors is defined in an arbitrary matter. It consists basically of a categorization of authentication mechanisms. At the contrary, our approach allows for a finer grained characterization of authentication factors. For example, a password-based authentication factor with a password length of 4 characters can be assigned to a different authentication level than one of length of 10 characters. Moreover, the authors do not propose any solution for combining authentication factor in order to achieve a better authentication level. (Al-Muhtadi, 2005) is closer to our approach by introducing the notion of confidence value in authentication mechanisms. The author uses the Gaia authentication framework which calculates the net confidence value of available Gaia authentication modules. It implies that the user has to authenticate himself by means of all available authentication mechanisms. Moreover the authors do not consider the use of heuristics for combining authentication mechanisms. In addition, the confidence in the service implementing the authentication mechanisms is not considered in like manner as the criteria on authentication mechanisms. To combine confidence values, the authors finally suggest to use the consensus operator from subjective logic. As depicted in figure 11, the consensus operator does not fulfill the requirements on opinions combination expressed in section 3.4. In figure 11, we show the evolution of consensus compared to combination of two opinions for max(ω_a, ω_b) equals to 0.1, 0.5 and 0.9. It is clear that, in the best case where uncertainty is maximised for the two opinions, consensus hardly rises above the max(ω_a, ω_b) (**RE1**). Moreover, the evolution of consensus is neither proportional to the distance between ω_a and ω_b (**RE2**) nor to the max(ω_a, ω_b) (**RE3**). In (M. Covington, 2004), the authors still propose to abstract authentication factor to subjective logic opinions. In order to calculate the confidence in a combination of authentication factors, the author also uses the consensus operator from subjective logic. Liberty Alliance (Liberty Alliance, 2005) introduces the notion of identity provider which is in charge of federating user identities. When users want to consume service, they authenticate to their identity provider by mean of an authentication context encapsulated in SAML assertions where the circumstance of the authentication (e.g. mechanism used, service) are described. With that additional information, the service provider can evaluate its trust in user's authentication. Moreover, the identity provider can still combine different authentication context. Nevertheless, the service provider still imposes the user to authenticate by using statistically defined authentication factors.

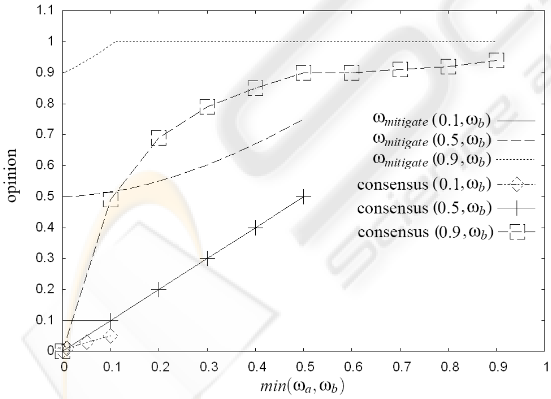


Figure 11: Consensus and Combination Operator.

6 CONCLUSION

In this paper, we propose an alternative approach to user authentication by means of a combination of authentication factors with a confidence value, so-called authentication level. We capitalize on subjective logic in order to define a trust metric for authentication level. Moreover, we define a new operator on subjective logic for mitigating opinions on combination of authentication factors. Our approach enables user to leverage the use of available authentication factors.

We are currently studying how to establish similarities between authentication factor by means of ontology in order to ease the definition of authentication level policy. We are also working on the extension of our approach to other type of information used for access control policy (e.g. contextual information).

Finally, we will implement our approach on a web service platform, and use a rule engine to optimize and ease user authentication to evaluate of such framework.

REFERENCES

Al-Muhtadi, J. (2005). *An Intelligent Authentication Infrastructure for Ubiquitous Computing Environments*. University of Illinois at Urbana-Champaign.

Josang, A. (2001). A logic for uncertain probabilities. In *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems.*

J.T Kohl, B.C Neuman, T. T. (1994). The evolution of the kerberos authentication system. In *Distributed Open Systems*. IEEE Computer Society Press.

Liberty Alliance (2005). Liberty Alliance Project.

M. Covington, M. A. e. a. (2004). Parametrized authentication. In *Proceedings of the 9th European Symposium on Research in Computer Security*. Springer.

M.K Reither, S. S. (1999). Authentication metric analysis and design. In *ACM Transactions on Information and System Security*. ACM Press.

OASIS (2005). XACML 2.0 - eXtended Access Control Markup Language.

Pfleeger, C. (1997). *Security in Computing*. Prentice-Hall, Inc.

S Ganeriwal, M. S. (2004). Reputation-based framework for high integrity sensor networks. In *SASN 04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*. ACM Press.

Schneier, B. (2005). *Two-Factor Authentication: Too Little, Too Late*. Communication of the ACM/Vol 48, No.4.

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton Univ. Press. Princeton, NJ.

W. Burr, D. Dodson, W. P. (2006). Electronic authentication guideline. In *NIST Special Publication 800 63*. National Institue of Standards and Technology.