A Method for Access Authorisation Through Delegation Networks

Audun Jøsang[1] **Dieter Gollmann**[2] **Richard Au**[1]

[1]School of Software Engineering and Data Communications*
QUT, Brisbane, Australia
Email: {a.josang, w.au}@qut.edu.au

[2]Distributed Systems Security Group
Hamburg University of Technology, Germany
Email: diego@tuhh.de

Abstract

Owners of systems and resources usually want to control who can access them. This must be based on having a process for authorising certain parties, combined with mechanisms for enforcing that only authorised parties are actually able to access those systems and resources. In distributed systems, the authorisation process can include negative authorisation (e.g. black listing), and delegation of authorisation rights, which potentially can lead to conflicts. This paper describes a method for giving authorisations through a delegation network, and where each delegation and authorisation is expressed in the form of a belief measure. An entity's total authorisation for a given resource object and access type can be derived by analysing the delegation network using subjective logic. Access decisions are made by comparing the derived authorisation measure with required threshold levels, which makes authorisations non-categorical. By setting the threshold level higher than the assigned measure of a single authorisation, it is possible to require multiple authorisations for accessing specific resources. The model is simple, intuitive and algebraic.

1 Introduction

Access control models specify how access to systems and resources is granted as a function of authorisation, delegation and security policies. Most models are based on the traditional concepts of a *subject* requestor, *object* resource and *access type*, with which the *owner/custodian* of the object resource can specify that a given subject can access a specific object with a specific access type. The access control policy can be expressed e.g. in matrix form, as rules, logical expressions or as graphs.

We will use the term *access scope* as a compact way of describing what delegations and authorisations apply to. The term *access scope* is defined as follows:

Definition 1 (Access Scope) *Access scope is the combined set of object resource(s), and access type(s) in a given access authorisation.*

It is useful to separate between the *authorisation phase*, i.e. when the authorisation authority defines which object resources a given subject shall be able to access, and the *enforcement phase*, i.e. when the system during operations actually makes sure that a party requesting access, actually is authorised. As an example, in the Web Services Security (WS Security) Framework [29] these phases are handled by the *Policy Administration Point* (denoted PAP hereafter), and the *Policy Enforcement Point* (denoted PEP hereafter) respectively.

Authorisation is traditionally done by assigning binary positive access attributes, so that subsequent access control decisions will be categorically positive. In case of general default access rights, discretionary black listing of subjects can be done by assigning negative access attributes.

In centralised systems, where a single owner authority is responsible for granting authorisation, the authorisation and enforcement processes are relatively straightforward. Authorisation can for example be done by defining access attributes in an access control list (ACL) in matrix form. XACML[1] [28] is a formal XML based language for expressing access control policies. When an access request arrives, the enforcement is based on looking up the ACL to see if the subject has been authorised. This is illustrated in Fig.1 where the resource owner A authorises the subject E to access resources by defining an access policy at the PAP. The access request consists of the subject identity and authentication credentials in addition to the actual access instructions (not shown in the figure). The system authenticates the identity at the start of the session (not shown), and the PEP verifies that sufficient access authorisation has been defined at every subsequent access request during the session. It can be mentioned that in the WS Security Framework, the PEP relies on the *Policy Decision Point* for the actual access decision, but this level of detail is not needed for this presentation.

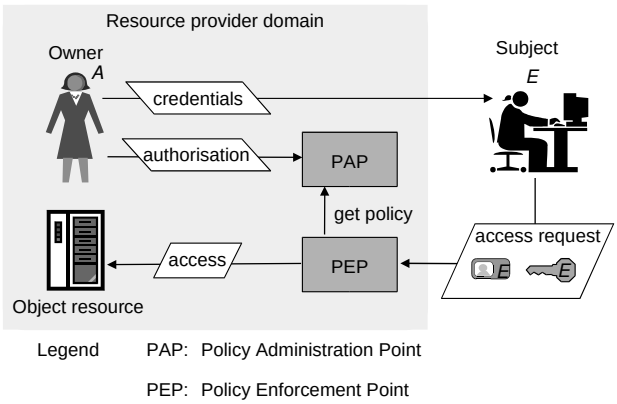

Figure 1: Access based on identity and local authorisation policies

*Support from the ARC Research Network Secure Australia acknowledged.

[1]eXtensible Access Control Markup Language

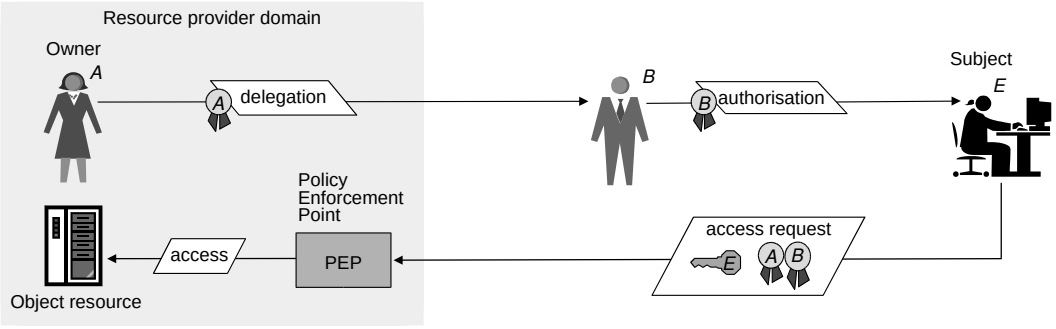

Figure 2: Access based on authorisation through a delegation path

In distributed and heterogeneous systems there might not be a central PAP; access authorisations can be issued by different parties, e.g in the form of digitally signed credentials, and a subject may thus collect credentials from a variety of sources, and later present those credentials (or refer to them) when making an access request.

The PEP therefore needs a method for obtaining access credentials from different sources, and an algorithm for processing them when deciding on a request; in particular, it may have to deal with contradicting credentials. There are two general options for analysing authorisation attributes issued in this way:

1. Design an algorithm based on case analysis that considers all possible reasons for granting or denying access. The challenge in this approach is the algorithm design, which may become complicated and difficult to manage and certify. Models in this category include (Bonatti *et al.* 2002) [6] and (Yao, 2003) [36].

2. Design a generic algorithm based on attributes coming with the credentials in the form of delegation and authorisation measures, and include external parameters such as risk and sensitivity for decision making. The challenges here is the assignment of measures in practical deployment and the computation of authorisation measures through delegation. Models in this category include (Brian *et al.* 2004) [35] and (Dimmock *et al.* 2004) [11] which combine risk with measures of access trust, but which do not consider delegation. Our method also uses this approach, and below we describe how such measures can be assigned, computed and interpreted in the context of access delegation.

Distributed authorisation can take place through delegation. This is common for example in business processes where managers can delegate to subordinates the capability of defining access policies to resources. In this scenario, which is illustrated in Fig.2, there is no central PAP, but delegations and authorisations take place in a distributed fashion.

Access delegation depends on transitive paths of delegation arcs. Each arc can be represented by a credential in the form of an attribute certificate. This model was implemented by AT&T Research Laboratories in PolicyMaker [5], and later enhanced in KeyNote [3]. REFEREE [9] is a system based on PolicyMaker aimed at controlling access to Web resources. In this way, access rights can be given through a delegation chain, as illustrated in Fig.2 where the resource owner A delegates to B to authorise subject E to access A's resources.

Fig.2 indicates that delegations and authorisations are issued in the form of certificates. When the subject presents these together with a specific access request, the PEP verifies that the delegation chain is valid for that specific access request. In other words, as long as the delegation chain is valid, it is not necessary to know the actual identity behind the access request [2]. It is here assumed that the public key used in the credential can be considered to be unique, which is feasible given that the name space for public keys is huge. It thus serves as an identifier for an anonymous entity that has been granted access through the delegation chain.

The access control model we describe in this paper is based on a *delegation network*, where multiple delegation paths can exist between the owner and the subject. This approach is illustrated in Fig.3 where, for simplicity, the certificate with index N represents the combined certificates from the whole network. Fig.3 indicates that the identity is supplied with the access request, but this is not strictly necessary if the subjects are allowed to remain anonymous when accessing resources. In a sense, the PAP can be considered distributed across the delegation network in this model.

In a delegation network it is possible that two delegates can grant conflicting (e.g. positive and negative) authorisations to the same party, so that a method for conflict resolution is needed in the decision logic. Examples of conflict resolution principles are *Negative (Positive)-takes-precedence*, *Strong-and-Weak*, *More-specific-takes-precedence*, *Time-takes-precedence* and *Predecessor-takes-precedence* [7, 13, 17, 32, 34]. Although some conflict resolution methods introduce weights [1, 33], or partial orders [26], most have in common that they use logical rules and principles for analysing binary authorisation and delegation statements.

Access control is often described in terms of trust, and the first common use of the term *trust management* was closely linked to the combination of authorisation, authentication and access control in distributed systems, as expressed by Blaze *et al.* (1996) [4]. The main idea behind their approach was that a system does not need to know the identities of those who are accessing its resources, only that they are "trusted" to do so. Although the meaning of trust in this context is not clearly defined, the approach allows verification of access credentials without necessarily authenticating entities in the traditional sense. It is assumed that the delegating party *trusts* the delegates, who in turn trust the parties they authorise. Blaze *et al.* defined trust management as:

> *"a unified approach to specifying and interpreting security policies, credentials, relationships which allow direct authorisation of security-critical actions."* [4]

Trust has become quite an overloaded concept [21], so that using it to describe access control can easily undermine, rather than promote, understanding and meaningful discussion. In order to avoid any confusion, it is important

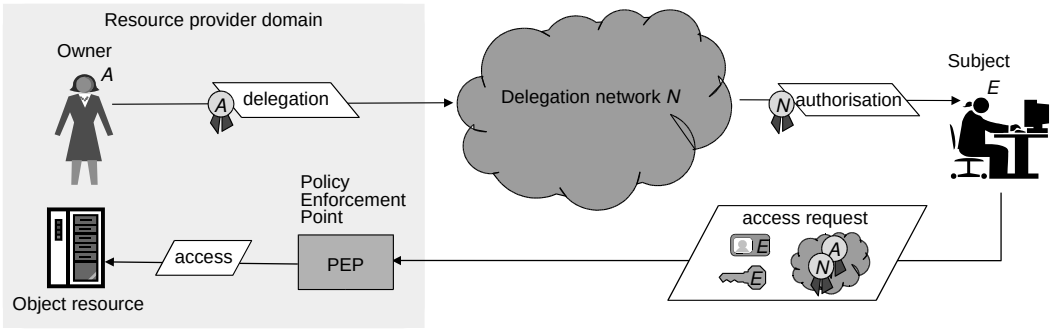

Figure 3: Access based on authorisation through a delegation network

to explicitly specify what is meant when trust is used in a specific context such as access control. The meaning of trust can be clarified with a *trust scope* defined as follows:

Definition 2 (Trust Scope) *Trust scope*[2] *is the specific type(s) of trust assumed in a given trust relationship. In other words, the trusted party is relied upon to have certain qualities, and the scope is what the trusting party assumes those qualities to be.*

In the context of access control, trust in a subject can therefore be interpreted as the belief that the subject will properly use, and not misuse, any access privileges he receives. It can be noted that the trust scope as defined above automatically emerges from the access scope as defined in Def.1, thereby making the meaning of trust in access control implicit. In access control models where trust is assumed to be binary (i.e. trusted or not trusted), the expression of trust is redundant, because that trust is uniquely determined by the scope of the actual delegations and authorisations.

When access trust is assumed to be discrete or continuous, and not just binary, specific measures must be expressed in addition to the access scope. Our method follows this approach, where beliefs in the subjects' reliability in handling and using specific access privileges get assigned to delegation and authorisation arcs.

In our model, every delegation and authorisation is expressed with a belief measure which can be seen as probability measures with an additional uncertainty dimension. The representation of beliefs will be described in more detail in Sec.6 and in the appendix. The delegation network can then be analysed algebraically with belief calculus in order to derive the *authorisation measure* expressed as a belief.

Since the derived authorisation measure is not binary, it does not categorically predetermine the access decision. Specific decision threshold levels can be defined according to the sensitivity or risk associated with the resource and access type. Access will be given if and only if the subject's derived authorisation measure is equal to or greater than the required access threshold level.

2 Characteristics of Delegation Networks

There is a close relationship between delegation transitivity and trust transitivity. Our model for analysing delegation networks is based on the trust network analysis method described in [19].

Trust transitivity means, for example, that if Alice trusts Bob who trusts Eric, then Alice will also trust Eric.

[2]The terms "trust context" [15], "trust purpose" [20] and "subject matter" [27] have been used in the literature with the same meaning.

It can be shown that trust is not always transitive in real life [8]. For example the fact that Alice trusts Bob to look after her child, and Bob trusts Eric to fix his car, does not imply that Alice trusts Eric for looking after her child, or for fixing her car. However, under certain semantic constraints [23], trust can be transitive, and a trust system can be used to derive trust. In the last example, trust transitivity collapses because the scopes of Alice's and Bob's trust scopes are different. Clearly, a transitive trust path requires a common trust scope.

Basic constructs of directed graphs can be used to represent delegation networks. Delegations and authorisations have in common that they are represented as arcs in the graph. However, a delegation is semantically different from an authorisation, in that it gives the right to delegate or to authorise, but it is not by itself an authorisation. In order for a directed delegation path to result in authorisation, it is required that the path must always have an authorisation as the last arc. This is expressed by the following criterion.

Definition 3 (Authorisation Derivation) *Derivation of authorisation through a delegation path requires that the last arc represents authorisation, and all previous arcs represent delegation.*

An access scope can be general, such as a whole database with all possible access types, or narrow, such a single record with only read access. In that sense, an access scope can be a subset of another. For a delegation chain to be valid, we require a common access scope along the delegation path. This is expressed with the following criterion.

Definition 4 (Access Scope Consistency) *A valid delegation path requires a common subset between the access scopes of the authorisation arc, and of all previous delegation arcs in the path. The derived authorisation scope is then the largest common subset.*

Trivially, every arc in the path can have the same access scope. When the two above requirements are satisfied, it is possible to grant authorisation through a transitive delegation path, or through a delegation network which can be represented as a directed graph consisting of multiple paths. Fig.3 indicates that the resource owner plays the role of the source, and the subject plays the role of the sink, meaning that the owner and the subject must be considered part of the delegation network.

A delegation path stops with the first authorisation arc encountered. It is, of course, possible for a principal to both delegate and authorise the same subject, but that should be expressed as two separate arcs.

For comparison it can be mentioned that SPKI[3] [12] is a digital certificate framework where every delegation is

[3]Simple Public Key Infrastructure

also an authorisation. In general the role of being a delegate should be separated from the role of being authorised, and delegates should not be able to authorise themselves. Our model therefore requires that a node have two separate incoming arcs, one for delegation and one for authorisation, in order for that node to be both delegate and authorised at the same time.

In a delegation network it is possible that one party gives delegations and authorisations to multiple other parties. It is also possible that multiple delegates give delegations and authorisations to the same party. Fig.4 illustrates a simple example where the resource owner A delegates to B and D, and where both B and D delegate to C who in turn authorises the subject E. Fig.4 also indicates that this can be considered as an *indirect authorisation* of subject E by owner A.

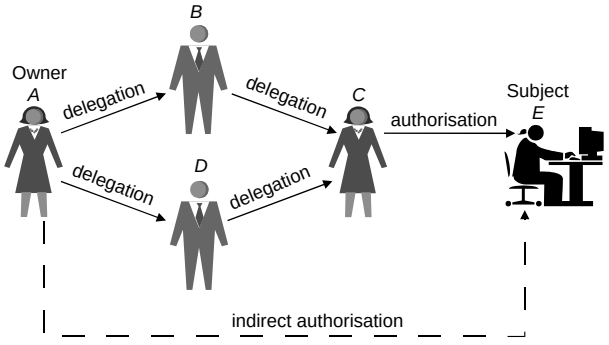

Figure 4: Delegation network with parallel paths

The existence of parallel delegation paths can lead to conflicts, for example, if one path dictates positive authorisation, and the other dictates negative authorisation. In our approach, the analysis and derivation of authorisation measures implicitly takes authorisation conflicts into account, and thereby eliminates the need for explicit conflict resolution methods.

3 Structured Notation

A single authorisation or delegation can be expressed as a directed arc between two nodes that represent the source and sink of that single arc. For example the arc $[A, B]$ means that A delegates or authorises B.

The symbol ":" will be used to denote the transitive connection of two consecutive arcs to form a transitive delegation path. The symbol "$\diamond$" will be used for the combination of two parallel paths, as it visually resembles a simple graph of two parallel paths between a pair of agents such as between A and C in Fig.4. The delegation network of Fig.4 can then be expressed in short notation as:

$$([A, E]) = \\ (([A, B] : [B, C]) \diamond ([A, D] : [D, C])) : [C, E] \tag{1}$$

where the access scope is implicit. Let the access scope e.g. be defined as:

σ: *"read-access to all staff records"*.

We will consider an authorisation to be *functional* because it actually empowers the subject to perform access functions. Similarly we consider a delegation to be a *referral* which does not empower the recipient to perform any access functions. We can therefore say that the access scope σ can have a functional, or a referral variant.

Let the functional (authorisation) variant of an access scope be denoted by "fσ" and the referral (delegation) variant by "rσ". A distinction can be made between initial *direct authorisation* and derived *indirect authorisation*. Whenever relevant, the access scope can be prefixed with "d" to indicate direct authorisation/delegation (dσ), and with "i" to indicate indirect authorisation/delegation (iσ). This can be combined with the referral or functional variant, so that for example indirect authorisation can be denoted as "ifσ". A reference to the access scope can then be explicitly included in the arc notation as e.g. denoted by $[A, E, \mathrm{if}\sigma]$, which can be read as *"indirect authorisation of subject E by owner A"*. The delegation network of Fig.4 is explicitly expressed as follows.

$$([A, E, \mathrm{if}\sigma]) = \\ (([A, B, \mathrm{dr}\sigma] : [B, C, \mathrm{dr}\sigma]) \diamond ([A, D, \mathrm{dr}\sigma] : [D, C, \mathrm{dr}\sigma])) \\ : [C, E, \mathrm{df}\sigma] \tag{2}$$

Fig.4 contains two paths. The graph consisting of the two separately expressed paths would be:

$$([A, E]) = \\ ([A, B] : [B, C] : [C, E]) \diamond ([A, D] : [D, C] : [C, E]) \tag{3}$$

A problem with Eq.(3) is that the arc $[C, E]$ appears twice. Although Eq.(1) and Eq.(3) consist of the same two paths, their combined structures are different. Some computational models would be indifferent to Eq.(1) and Eq.(3), whereas others would produce different results depending on which expression is being used. When implementing the serial ":" as binary logic "AND", and the parallel "$\diamond$" as binary logic "OR", the results would be equal. However, when implementing ":" and "$\diamond$" as probabilistic multiplication and comultiplication respectively, the results would be different. It would also be different in our method which uses subjective logic operators for transitivity and parallel combination. These operators are described in the appendix. In general, it is therefore desirable to express graphs in a form where an arc only appears once. This will be called a *canonical expression*.

Definition 5 (Canonical Expression) *An expression of a delegation graph in structured notation where every arc only appears once is called canonical.*

With this structured notation, arbitrarily large delegation networks can be explicitly expressed in terms of arc, scope and other attributes such as measure and time.

4 Delegation Network Analysis

A general delegation network is based on directed delegation/authorisation arcs between pairs of nodes. With no restrictions on the possible arcs, delegation paths from a given source X to a given target Y can contain loops and dependencies, which could result in inconsistent calculative results. Dependencies in the delegation graph must therefore be controlled when applying calculative methods to derive measures of authorisation. *Normalisation* and *simplification* are two different approaches to dependency control. Normalisation is for example a property of the PageRank algorithm proposed by Page *et al.* (1998) [30], and used by the Google search engine to rank search results. An other example of normalisation is the EigenTrust algorithm proposed by Kamvar *et al.* (2003) [25], which is aimed at deriving global reputation scores in P2P communities, with the purpose of assisting members in

choosing the most reputable peers. The advantage of normalisation is that general directed graphs can be analysed without modification. The disadvantage of normalisation is that measures of belief, trust or reputation associated with arcs are relative, and therefore can not be interpreted in any absolute sense. As a consequence, models based on normalisation are unable to take negative measures into account. For example, the PageRank algorithm does not recognise when a web page gives warnings about fraudulent web pages, and simply includes it as a positive rating of the fraudulent web pages. Similarly, EigenTrust requires that a negative sum of ratings be truncated to a zero sum of ratings.

Our model is based on simplification which removes cycles and dependencies from a general directed graph, producing a so-called *directed series-parallel graph* which can readily be expressed in canonical form [14]. Delegation graph simplification can be done with the same method as in trust graph simplification described in detail in [19]. The advantage of simplification is that normalisation of the computational results no longer is needed, so that the derived values can be interpreted in an absolute sense, and not just as a relative rank such as in PageRank.

We will in the following assume that delegation graphs have been simplified so that they can be expressed in a canonical form.

5 Storage of Delegations and Authorisations

Authorisations and delegations must be protected from tampering. The two main alternatives for protection are *protected memory*, and *cryptographic protection*, e.g. in the form of digitally signed certificates.

This brings us to the issue of where delegations and authorisations are stored and how they are provided to the PEP. Two possibilities for storing delegations/authorisations are briefly outlined below.

1. **Storage at subject side.** Every time a new authorisation/delegation certificate is issued, the recipient receives it together with all valid delegation certificates that the issuer has previously received. By taking Fig.4 as example, assume that C has previously received the access delegations $[A, B]$ and $[B, C]$, and that C issues an authorisation to E. Then E will receive $[A, B]$, $[B, C]$ and the new $[C, E]$. Whenever a subject needs access to a resource object, he sends all the relevant authorisation/delegation certificates to the relevant PEP, that then analyses these to make the access decision. In this scenario, the PEP is a fully distributed function.

   The advantage of this approach is that it is totally distributed, and that every subject is able to derive his actual authorisation measure at any time.

   The disadvantage of this approach is that the same authorisation/delegation certificates are stored in multiple places, and that it is difficult to make every affected entity in a network aware of changes in authorisation/delegation measures. Another disadvantage is that it is difficult to enforce that all relevant certificates are always taken into account. A certificate that expresses negative authorisation of a given subject can represent an incentive for the subject to exclude it from the access request in an attempt to gain illegal access. Alternatively, if the subject in his job function is expected to perform a specific access that he, for some reason, does not want to execute, he can pretend not to have the necessary authorisation, in an attempt to escape from his duty.

2. **Storage at the owner side.** Whenever a new authorisation/delegation certificate is issued, it is sent to the PAP of the relevant resource. The PAP of a given resource will therefore have complete overview of all current authorisations/delegations at any time. Whenever subjects need access to resource objects, they simply submit their requests together with identifiers and authentication credentials to the PEP, who in turn calls the Policy Decision Point to make the access decisions. In this scenario, the PAP is co-located with the resources.

   The advantage of this approach is that each authorisation/delegation certificate only needs to be stored at one location, and that it is simple to make changes in authorisations/delegations. Another advantage is that subjects are unable to hide/suppress negative authorisations.

   The disadvantage of this approach is that the centralisation can represent a bottleneck, and that the subjects need to query the owner in order to know what actual authorisations they have.

Based on the above discussion, we believe that protected memory storage under the PEP at owner side represents the best solution, and this architecture will be assumed in the following.

6 Measures and Computation

Each isolated authorisation and delegation can be expressed individually as entries in a table, and an automated parser can establish valid authorisation delegation paths and graphs depending on the need.

Let us again consider the example delegation network of Fig.4, where delegations and authorisations are represented in the form of Table 1. In order to compute the authorisation measure, each delegation and authorisation arc must be assigned a measure that we will call belief and denote by ω.

Table 1: Authorisations (f) and delegations (r) of Fig.4

Arc	Scope	Variant	Measure
$[A, B]$	σ	r	w_1
$[A, D]$	σ	r	w_2
$[B, C]$	σ	r	w_3
$[C, E]$	σ	f	w_4
$[D, C]$	σ	r	w_5

A parser, as e.g. described in [19], can go through Table 1 to determine the delegation graph of Fig.4.

When analysing delegation networks, the delegation and authorisation measures must allow a *meaningful* interpretation and be *consistently* measured by all involved parties. By explicitly defining the access scope σ, the interacting parties are able to establish a common understanding of what the measures of delegation and authorisation relate to.

Delegations and authorisations have traditionally been considered binary in models proposed in the literature. In our method, they are expressed as continuous measures. While no natural or physical continuous units exist, our method expresses measures of delegation and authorisation as *beliefs*, which can be interpreted as probability value with an additional dimension that expresses the certainty or *confidence* of the probability value. This is described in detail in the appendix.

The belief measure associated with delegation or authorisation should be interpreted as the expected likelihood with which a specific delegation or authorisation *will*

not be misused by the delegate/authorised party. Derived authorisation measures can then be directly used in quantitative risk assessments for making dynamic access control decisions. The advantage of using belief measures, as opposed to probability measures, is that it allows delegation network analysis involving transitivity and parallel paths, and that access to resources can be tuned according to the associated sensitivity or risk level.

Initial assignment of delegation/authorisation belief measures can be done in various ways. Statistical data, e.g. from reputation systems, can be directly translated into belief measures. Alternatively the belief measure assignments can be determined on a subjective basis by the delegating/authorising party. Fuzzy verbal categories [31] can be mapped to belief measures to simplify the human cognitive task of determining belief measures. It is important to note that belief measures and subjective logic are compatible with probability measures and probability calculus.

The confidence of the authorisation measure derived from a given path should decrease as a function of the length of the path. A consequence of this is that long transitive delegation paths result in a reduced confidence measure, not in negative authorisation. The *discounting operator* of subjective logic, described in the appendix, satisfies this requirement, and is also the operator used for computing transitive delegations in our method. When using the discounting operator, denoted by '$\otimes$', the expression for deriving belief from the graph of Fig.2 is written as

$$\omega_E^A = \omega_B^A \otimes \omega_E^B \tag{4}$$

Arbitrary deep delegation paths are allowed, so the delegation depth control must be exercised by the resource owner through the authorisation threshold that the PEP will use when an access request is received. This can be seen in contrast to other delegation depth control solutions that have been proposed and implemented. For example, in the PolicyMaker and Keynote schemes, no delegation depth control exists, so that there is no difference between the authorisation strength resulting from short or arbitrary long delegation paths. SDSI/SPKI [12] has a simple scalar or binary "delegation" parameter that controls the delegation depth, meaning that when the parameter reaches a predefined threshold, further delegation is not permitted. The X.509 standard [16] provides control through the use of basic constraints, name constraints, policy constraints etc., but their management is considered to be complex.

Combination of multiple parallel delegation paths, should result in an increased confidence in the derived authorisation measure. The *consensus operator* of subjective logic, also described in the appendix, satisfies this parallel combination requirement, and is also the operator for computing parallel combinations used in our method. When using the consensus operator, denoted by '$\oplus$', together with the discounting operator, the expression for the derived belief from the graph of Fig.4 is written as:

$$\omega_E^A = ((\omega_B^A \otimes \omega_C^B) \oplus (\omega_D^A \otimes \omega_C^D)) \otimes \omega_E^C \tag{5}$$

Allowing positive and negative authorisations simultaneously is a complicating factor when combining delegation paths. In most models proposed in the literature, paths with negative authorisation must be specified separately, and methods for resolving conflicts in case of both positive and negative paths must be explicitly designed.

In the examples of Sec.7 delegation and authorisation measures are expressed as beliefs, allowing the belief calculus of subjective logic to be used when analysing delegation networks. In subjective logic, there is no need to treat negative and positive authorisations separately, as they are both implicitly handled by the belief calculus.

Time and validity periods are information elements that should be specified as a part of authorisations and delegations. These elements are necessary not only to allow the granting parties to change authorisations and delegations over time, but also in order to enable a PEP to make access decisions based on the most recent authorisations/delegations available.

7 Examples

The mathematical operators used in the following examples are described in the appendix. The actual belief measures are unrealistically low in the examples. The reason for using low measures is to allow nice graphical visualisations for humans. The PEP would of course be indifferent to the aesthetics of graphical illustrations.

7.1 Tuning Authorisation Requirements

When subjective logic is used to derive authorisation measures, it is possible to delegate in such a way that a subject needs authorisations from multiple parties in order to be able to access a given resource object.

Let the access decision threshold be defined by T, meaning that the probability expectation value of the derived authorisation opinion must be at least T. The owner A can then grant delegations to $B_1,\ B_2,\ \ldots,\ B_n$ in such a way that a minimum subset of the delegates need to authorise a given subject.

Let for example $T = 0.9$, and the access policy for a specific resource object be that at least three authorisations are needed. This can be achieved by giving delegations with opinion values $(0.6,\ 0.0,\ 0.4,\ 0.5)$ to each B_i. The parallel combination of two such delegations produces the opinion $(0.75,\ 0.00,\ 0.25,\ 0.50)$ with expectation value 0.88 which is insufficient for a positive access decision. Adding a third delegation produces the opinion $(0.82,\ 0.00,\ 0.18,\ 0.50)$, with expectation value 0.91, which is sufficient for a positive access decision. This assumes that the delegates authorise with opinion weights $(1.0,\ 0.0,\ 0.0,\ 0.5)$.

The delegates can apply the same principle to subdelegates, so that rather complex structures of minimum sets of delegations and authorisations can be constructed. Suitable delegation measures can be determined as a function of the threshold level and the required number of minimum authorisations.

It is also possible to tune the threshold level for positive access decision, e.g. as a function of information sensitivity or risk assessment related to the source object. With the authorisation measures of the example above, a threshold $T = 0.85$ would require two authorisations, and a threshold $T = 0.80$ would require only one authorisation.

7.2 Derivation of Authorisation Measures

This numerical example is based on the delegation graph of Fig.4. Table 2 specifies the corresponding delegation/authorisation values expressed as subjective opinions.

The threshold level for a positive access decision is $T = 0.8$, meaning that the probability expectation value of the derived authorisation measure must be greater or equal to 0.8 in order for the access decision to be positive.

Notice that the table includes time stamps, and that there are two entries for the arc $[A, B]$. Assume that $\tau_1 < \tau_2$, meaning that the last entry for $[A, B]$ is the most recent. By applying the discounting and consensus operators to the expression of Eq.(1), the derived authorisation measures can be computed[4].

[4]The Subjective Logic API, available at http://security.dstc.com/spectrum/ , was used to compute the derived values.

Table 2: Example delegation and authorisation measures with reference to Fig.4

Arc	Measure	Time
$[A, B]$	$\omega_B^A = (0.9,\ 0.0,\ 0.1,\ 0.5)$	τ_1
$[A, D]$	$\omega_D^A = (0.9,\ 0.0,\ 0.1,\ 0.5)$	τ_1
$[B, C]$	$\omega_C^B = (0.9,\ 0.0,\ 0.1,\ 0.5)$	τ_1
$[C, E]$	$\omega_E^C = (0.9,\ 0.0,\ 0.1,\ 0.5)$	τ_1
$[D, C]$	$\omega_C^D = (0.3,\ 0.0,\ 0.7,\ 0.5)$	τ_1
$[A, B]$	$\omega_B^{'A} = (0.0,\ 0.9,\ 0.1,\ 0.5)$	τ_2

- **Case a.** First assume that A derives the authorisation measure of E at time τ_1, in which case the first entry for the arc $[A, B]$ in Table 2 is used. The expression for the derived measure and the numerical result is given below.

$$\begin{aligned} \omega_E^A &= ((\omega_B^A \otimes \omega_C^B) \oplus (\omega_D^A \otimes \omega_C^D)) \otimes \omega_E^C \\ &= (0.74,\ 0.00,\ 0.26,\ 0.50) \end{aligned} \tag{6}$$

with probability expectation value $\mathrm{E}(\omega_E^A) = 0.87$. With the access decision threshold $T = 0.8$, it can be seen that $\mathrm{E}(\omega_E^A) > T$, so that the access decision will be positive in this case.

With Eq.(11) and Eq.(13), the derived authorisation measure can be translated into a beta PDF visualised Fig.5 below.

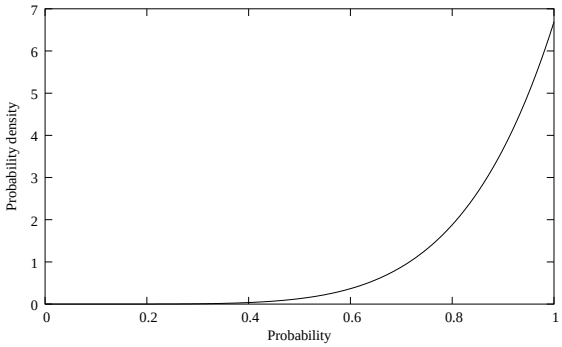

Figure 5: Case a: $[A, E]$ expressed as beta$(6.7,\ 1.0)$

- **Case b.** Let us now assume that, based on a new delegation issued at time τ_2, B's delegation measure is suddenly reduced to that of the last entry for $[A, B]$ in Table 2. As a result of this A needs to update the derived authorisation measure of E, and computes:

$$\begin{aligned} \omega_E^{'A} &= ((\omega_B^{'A} \otimes \omega_C^B) \oplus (\omega_D^A \otimes \omega_C^D)) \otimes \omega_E^C \\ &= (0.287,\ 0.000,\ 0.713,\ 0.500) \end{aligned} \tag{7}$$

with probability expectation value $\mathrm{E}(\omega_E^{'A}) = 0.64$. With the access decision threshold $T = 0.8$, it can be seen that $\mathrm{E}(\omega_E^{'A}) < T$, which means that the access decision will be negative in this case. The updated authorisation measure can be mapped to the beta PDF illustrated in Fig.6.

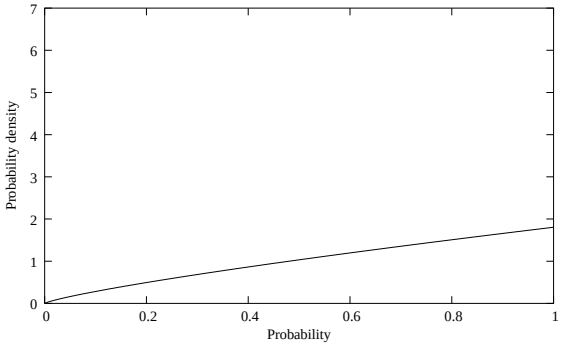

Figure 6: Case b: $[A, E]'$ expressed as beta$(1.8,\ 1.0)$

It can be seen that the authorisation illustrated in Fig.5 is relatively strong, but that the authorisation in Fig.6 approaches the uniform distribution, and therefore is very uncertain. The interpretation of this is that the negative delegation introduced in the $[A, B]$ arc in case b) has rendered the path $[A, B] : [B, C] : [C, E]$ useless, i.e. when B gets a negative delegation certificate, then whatever delegation B gives is completely discounted. It is as if B had not delegated anything at all. As a result, the derived authorisation measure of E must be based on the path $[A, D] : [D, C] : [C, E]$ which was already weak from the start. Thus negative delegation does not produce a negative derived authorisation, but rather more uncertainty in the derived authorisation measure.

The only way to propagate a negative authorisation intact through a transitive delegation path is when the last authorisation arc is negative and all the previous delegation arcs in the path are positive. Thus, a negative authorisation needs positive delegations in order to propagate through transitive delegation network.

8 Conclusion and Further Work

Access authorisation and delegation is traditionally based on categorical rules applied to binary statements. This paper describes a framework for setting flexible access control policies by applying subjective logic to reason about authorisation and delegation in transitive delegation networks. The main advantages of our approach are 1) the authorisation process can be distributed, 2) access decisions can be dynamically determined as a function of the sensitivity/risk of the object to be accessed, and 3) authorisation conflicts are implicitly handled by the decision logic, and 4) that it is possible to authorise principals in such a way that many authorisations are needed to access a give resource.

Controlling access to web services is a good candidate where our method could be applied. The owner of a given resource object can for example delegate to business partners the ability to grant access to that resource object. Our method can be combined with access control frameworks such as WS Security, and with formal access policy languages such as XACML. When using XACML, the access scopes and belief measures associated with delegations and authorisations can simply be encoded as XML attributes, and communicated over the Internet. In order for the proposed method to be practical, we see the need for specifying an application layer protocol to be used in the communication between delegates and resource owners when delegations and authorisations are issued.

References

[1] I. Agudo, J. Lopez, and J.A. Montenegro. A Representation Model of Trust Relationships Supporting Delegation. In Nikolau C., editor, *The Proceedings of the Third International Conference on Trust Management*, Paris, May 2005.

[2] R. Au, H. Vasanta, K-K.R Choo, and M. Looi. A User-Centric Anonymous Authorisation Framework in E-commerce Environments. In *Proceedings of the Sixth International Conference on Electronic Commerce (ICEC'04)*, 2004.

[3] Matt Blaze, Joan Feigenbaum, and Angelos D. Keromytis. KeyNote: Trust Management for Public-Key Iinfrastructures. In *Proceedings of the 1998 Secure Protocols International Workshop*, Cambridge, England, 1998.

[4] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proceedings of the 1996 IEEE Conference on Security and Privacy*, Oakland, CA, 1996.

[5] Matt Blaze, Joan Feigenbaum, and Martin Strauss. Compliance Checking in the PolicyMaker Trust Management System. In *Proceedings of Financial Crypto*, 1998.

[6] P. Bonatti, S. De Capitani di Vimercati, and P. Samarati. An Algebra for Composing Access Control Policies. *ACM Transactions on Information and System Security (TISSEC)*, 5(1):1–35, February 2002.

[7] S. Castano, M. Fugini, G. Martella, and P. Samarati. *Database Security*. Addison-Wesley, 1994.

[8] B. Christianson and W. S. Harbison. Why Isn't Trust Transitive? In *Proceedings of the Security Protocols International Workshop*. University of Cambridge, 1996.

[9] Y.-H. Chu, J. Feigenbaum, B. LaMacchia, P. Resnick, and M. Strauss. REFEREE: Trust Management for Web Applications. *World Wide Web Journal*, 2:127–139, 1997.

[10] M.H. DeGroot and M.J. Schervish. *Probability and Statistics (3rd Edition)*. Addison-Wesley, 2001.

[11] N. Dimmock et al. Using Trust and Risk in Role-Based Access Control Policies. In *Proceedings of the 9th ACM Symposium on Access Control Models and Technologies (SACMAT)*, New York, June 2004.

[12] C. Ellison et al. *RFC 2693 - SPKI Certification Theory*. IETF, September 1999. url: http://www.ietf.org/rfc/rfc2693.txt.

[13] W. Essmayr, F. Kastner, G. Pernul, S. Preishuber, and A. Tjoa. Access Controls for Federated Database Environments. In *roc. Joint IFIP TC 6 & TC 11 Working Conf. on Communications and Multimedia Security*, Graz 1995.

[14] P. Flocchini and F.L. Luccio. Routing in Series Parallel Networks. *Theory of Computing Systems*, 36(2):137–157, 2003.

[15] T. Grandison and M. Sloman. A Survey of Trust in Internet Applications. *IEEE Communications Surveys and Tutorials*, 3, 2000.

[16] ITU. *Recommendation X.509 v3, The Directory: Authentication Framework (also known as ISO/IEC 9594-8*. International Telecommunications Union, Telecommunication Standardization Sector(ITU-T), June 1997.

[17] S. Jajodia, P. Samarati, V. S. Subrahmanian, and E. Bertino. A unified framework for enforcing multiple access control policies. *SIGMOD Rec.*, 26(2):474–485, 1997.

[18] A. Jøsang. A Logic for Uncertain Probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, June 2001.

[19] A. Jøsang, E. Gray, and M. Kinateder. Simplification and Analysis of Transitive Trust Networks. *Web Intelligence and Agent Systems*, 4(2):139–161, 2006.

[20] A. Jøsang, R. Ismail, and C. Boyd. A Survey of Trust and Reputation Systems for Online Service Provision. *Decision Support Systems*, 43(2):618–644, 2007.

[21] A. Jøsang, C. Keser, and T. Dimitrakos. Can We Manage Trust? In P. Herrmann et al., editors, *Proceedings of the Third International Conference on Trust Management (iTrust)*, Versailes, May 2005.

[22] A. Jøsang and D. McAnally. Multiplication and Comultiplication of Beliefs. *International Journal of Approximate Reasoning*, 38(1):19–51, 2004.

[23] A. Jøsang and S. Pope. Semantic Constraints for Trust Tansitivity. In S. Hartmann and M. Stumptner, editors, *Proceedings of the Asia-Pacific Conference of Conceptual Modelling (APCCM) (Volume 43 of Conferences in Research and Practice in Information Technology)*, Newcastle, Australia, February 2005.

[24] A. Jøsang, S. Pope, and M. Daniel. Conditional deduction under uncertainty. In *Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005)*, 2005.

[25] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina. The EigenTrust Algorithm for Reputation Management in P2P Networks. In *Proceedings of the Twelfth International World Wide Web Conference*, Budapest, May 2003.

[26] N. Li, J.C. Mitchell, and W.H. Winsborough. Design of a Role-Based Trust Management Framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 114–130. IEEE Computer Society Press, May 2002.

[27] G. Mahoney, W. Myrvold, and G.C. Shoja. Generic Reliability Trust Model. In A. Ghorbani and S. Marsh, editors, *Proceedings of the 3rd Annual Conference on Privacy, Security and Trust*, St.Andrews, New Brunswick, Canada, October 2005.

[28] OASIS. *eXtensible Access Control Markup Language (XACML) Version 1.0 3*. Organization for the Advancement of Structured Information Standards, 18 February 2003.

[29] M. O'Neill et al. *Web Services Security*. McGraw-Hill, New York, 2003.

[30] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.

[31] Simon Pope and Audun Jøsang. Analsysis of Competing Hypotheses using Subjective Logic. In *Proceedings of the 10th International Command and Control Research and Technology Symposium (ICCRTS)*. United States Department of Defense Command and Control Research Program (DoDCCRP), 2005.

[32] C. Ruan and V. Varadharajan. Resolving Conflicts in Authorization Delegations. In *Proceedings of the 7th Australasian Conference on Information Security and Privacy*. Springer, 2002.

[33] C. Ruan and V. Varadharajan. A Weighted Graph Approach to Authorization Delegation and Conflict Resolution. In H. Wang et al., editors, *Proceedings of the 9th Australasian Conference on Information Security and Privacy*. Springer, 2004.

[34] R.S. Sandhu and P. Samarati. Access Control: Principles and Practice. *IEEE Communications Magazine*, 32(9):40–48, 1994.

[35] B. Shand, N. Dimmock, and J. Bacon. Trust for Ubiquitous, Transparent Collaboration. In *Proceedings of the 2nd UK-UbiNet Workshop*, Cambridge, May 2004.

[36] Walt Teh-Ming Yao. *Trust Management for Widely Distributed Systems*. PhD thesis, University of Cambridge, 2003.

Appendix A: Belief Representation in Subjective Logic

Subjective logic [18] is a belief calculus that can be used for analysing delegation networks. In this appendix, we describe how delegation/authorisation measures can be expressed as beliefs.

Subjective logic uses a belief metric called *subjective opinion* to express beliefs. A subjective opinion denoted by $\omega_x^A = (b_x^A, d_x^A, u_x^A, a_x^A)$ is an ordered tuple where $b_x^A, d_x^A, u_x^A, a_x^A \in [0, 1]$. The parameters b, d, and u represent belief, disbelief and uncertainty, respectively where the following equation holds:

$$b_x^A + d_x^A + u_x^A = 1 \ . \tag{8}$$

The parameter a_x^A reflects the base rate of the belief in the absence of evidence. The default base rate can be set to $a_x^A = 0.5$. An opinion's probability expectation value is:

$$\mathrm{E}(\omega_x^A) = b_x^A + a_x^A u_x^A \ . \tag{9}$$

An opinion expresses the observing party A's belief in the truth of statement x, similarly to a probability, but opinions allow much richer forms of expression than probabilities do. In case of total ignorance, the probability expectation value is equal to the base rate a_x^A. When the statement x for example says *"Party B will not misuse his delegation/authorisation privileges"* then A's subjective opinion about x can be interpreted as a measure of how willing A is to delegate/authorise B, which can also be denoted as ω_B^A.

The opinion notation ω_B^A can be used to represent authorisation/delegation, where A and B are the source and sink respectively of the arc $[A, B]$.

The opinion space can be mapped into the interior of an equal-sided triangle, where, for an opinion $\omega_x = (b_x, d_x, u_x, a_x)$, the three parameters b_x, d_x and u_x determine the position of the point in the triangle representing the opinion. Fig.7 illustrates an example where the opinion about a proposition x from a binary frame of discernment has the value $\omega_x = (0.7,\ 0.1,\ 0.2,\ 0.5)$.

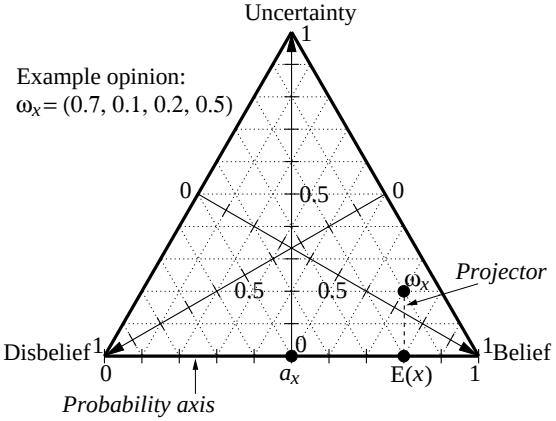

Figure 7: Opinion triangle with example opinion

Opinions can be ordered according the following rules by priority:

1. The opinion with the greatest probability expectation is the greatest opinion.
2. The opinion with the least uncertainty is the greatest opinion.

The probability density over binary event spaces can be expressed as beta PDFs (probability density functions) denoted by beta (α, β) [10], and opinions can be mapped to beta PDFs. The beta-family of distributions is a continuous family of distribution functions indexed by the two parameters α and β. The beta PDF denoted by $\mathrm{beta}(\alpha, \beta)$ can be expressed using the gamma function Γ as:

$$\mathrm{beta}(\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \tag{10}$$

where $0 \leq p \leq 1$ and $\alpha, \beta > 0$, with the restriction that the probability variable $p \neq 0$ if $\alpha < 1$, and $p \neq 1$ if $\beta < 1$. The probability expectation value of the beta distribution is given by:

$$\mathrm{E}(p) = \alpha/(\alpha + \beta). \tag{11}$$

Let r and s express the number of positive and negative past observations respectively, and let a express the *a priori* or base rate probability before any observations have been made, then α and β can be determined as:

$$\alpha = r + 2a \ , \qquad \beta = s + 2(1-a) \ . \tag{12}$$

A bijective mapping between the opinion parameters and the beta PDF parameters can be analytically derived [18] as:

$$\begin{cases} b_x = r/(r+s+2) \\ d_x = s/(r+s+2) \\ u_x = 2/(r+s+2) \\ a_x = \text{base rate of } x \end{cases} \iff \begin{cases} r & = 2b_x/u_x \\ s & = 2d_x/u_x \\ 1 & = b_x + d_x + u_x \\ a & = \text{base rate of } x \end{cases} \tag{13}$$

This means for example that a totally ignorant opinion with $u_x = 1$ and $a_x = 0.5$ is equivalent to the uniform PDF beta $(1, 1)$ illustrated in Fig.8.

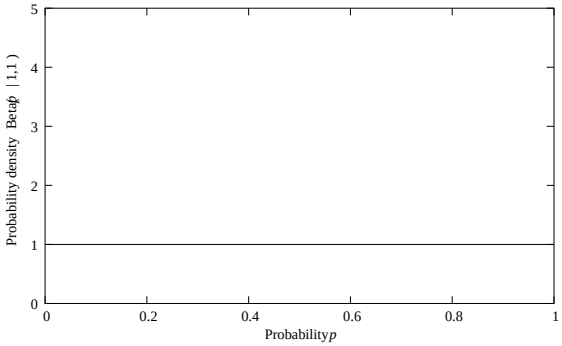

Figure 8: *A priori* uniform beta(1,1)

It also means that a dogmatic opinion with $u_x = 0$ is equivalent to a spike PDF with infinitesimal width and infinite height expressed by beta $(b_x\eta,\ d_x\eta)$, where $\eta \to \infty$. Dogmatic opinions can thus be interpreted as being based on an infinite amount of evidence.

When nothing is known except that the state space is binary (i.e. $a = 0.5$), the *a priori* distribution is the uniform beta with $\alpha = 1$ and $\beta = 1$. Then after r positive and s negative observations the *a posteriori* distribution is the beta PDF with the parameters $\alpha = r + 1$ and $\beta = s + 1$.

For example the beta PDF after observing 7 positive and 1 negative outcomes is illustrated in Fig.9, which also is equivalent to the opinion illustrated in Fig.7

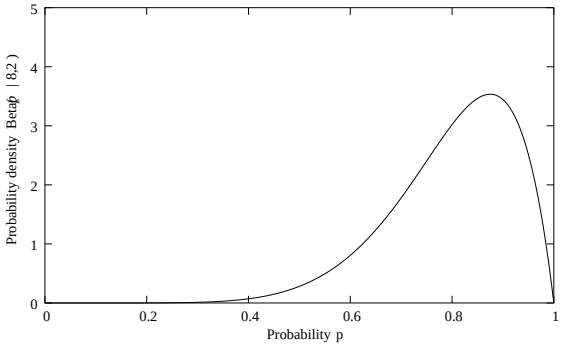

Figure 9: *A posteriori* beta(8,2) after 7 positive and 1 negative observations

By definition, the expectation value of the PDF is always equal to the expectation value of the corresponding subjective opinion. This provides a sound mathematical basis for combining opinions using Bayesian updating of beta PDFs.

Appendix B: Mathematical Operators for Delegation Network Analysis

Subjective logic defines a number of operators [18, 22, 24, 31]. Some operators represent generalisations of binary logic and probability calculus whereas others are unique to belief theory because they depend on belief ownership. By explicitly expressing belief ownership, it is possible to allow different principals to hold different and possibly conflicting beliefs about the same thing. Here we will only focus on the *discounting* and the *consensus* operators, because they are suitable for implementing the transitivity and parallel combination functions respectively. These operators, and their mathematical expressions, are described below[5].

[5]Online demonstrators at: http://security.dstc.com/spectrum/trustengine/

- **Discounting** is used to compute transitive delegations. Assume two agents A and B where A gives a measure of delegation to B, denoted by $\omega_B^A = (b_B^A, d_B^A, u_B^A, a_B^A)$. In addition B gives a measure of authorisation to C, denoted by $\omega_C^B = (b_C^B, d_C^B, u_C^B, a_C^B)$. A's indirect authorisation in C can then be derived by discounting B's authorisation of C with A's delegation of B. The derived authorisation is denoted by $\omega_C^{A:B} = (b_C^{A:B}, d_C^{A:B}, u_C^{A:B}, a_C^{A:B})$. By using the symbol '$\otimes$' to designate this operator, we can write $\omega_C^{A:B} = \omega_B^A \otimes \omega_C^B$. The opinion parameters are defined as follows.

$$\begin{cases} b_C^{A:B} = b_B^A b_C^B \\ d_C^{A:B} = b_B^A d_C^B \\ u_C^{A:B} = d_B^A + u_B^A + b_B^A u_C^B \\ a_C^{A:B} = a_C^B\ . \end{cases} \tag{14}$$

The effect of discounting in a transitive path is to increase uncertainty, i.e. to reduce the confidence in the expectation value.

- **Consensus** is used to fuse two (possibly conflicting) authorisations into one. Let $\omega_C^A = (b_C^A, d_C^A, u_C^A, a_C^A)$ and $\omega_C^B = (b_C^B, d_C^B, u_C^B, a_C^B)$ be authorisation of C by A and B respectively. The opinion $\omega_C^{A\diamond B} = (b_C^{A\diamond B}, d_C^{A\diamond B}, u_C^{A\diamond B}, a_C^{A\diamond B})$ is then called the consensus between ω_C^A and ω_C^B, denoting the authorisation that an imaginary agent $[A, B]$ would give C, as if that agent represented both A and B. By using the symbol '$\oplus$' to designate this operator, we can write $\omega_C^{A\diamond B} = \omega_C^A \oplus \omega_C^B$. The opinion parameters are defined as follows.

Case I: $u_C^A + u_C^B - u_C^A u_C^B \neq 0$

$$\begin{cases} b_C^{A\diamond B} = \frac{b_C^A u_C^B + b_C^B u_C^A}{u_C^A + u_C^B - u_C^A u_C^B} \\ d_C^{A\diamond B} = \frac{d_C^A u_C^B + d_C^B u_C^A}{u_C^A + u_C^B - u_C^A u_C^B} \\ u_C^{A\diamond B} = \frac{u_C^A u_C^B}{u_C^A + u_C^B - u_C^A u_C^B} \\ a_C^{A\diamond B} = a_C^A \end{cases}$$

Case II: $u_C^A + u_C^B - u_C^A u_C^B = 0$

$$\begin{cases} b_C^{A\diamond B} = (\gamma^{A/B}\, b_C^A + b_C^B)/(\gamma^{A/B} + 1) \\ d_C^{A\diamond B} = (\gamma^{A/B}\, d_C^A + d_C^B)/(\gamma^{A/B} + 1) \\ u_C^{A\diamond B} = 0 \\ a_C^{A\diamond B} = a_C\ . \end{cases}$$

where the relative weight $\gamma^{A/B} = \lim(u_C^B/u_C^A)$ (15)

The effect of the consensus operator is to reduce uncertainty, i.e. to increase the confidence in the expectation value. In case the subjective opinions are probability values ($u = 0$), Case II produces the weighted average.