Policy Languages and their Suitability for Trust Negotiation

Martin Kolar, Carmen Fernandez-Gago, Javier Lopez
Network, Information and Computer Security Lab
University of Malaga, Spain
Email: {kolar,mcgago,jlm}@lcc.uma.es

Abstract

Entities, such as people, companies, institutions, authorities and web sites live and exist in a conjoined world. In order to live and enjoy social benefits, entities need to share knowledge, resources and to cooperate together. The cooperation brings with it many new challenges and problems, among which one is the problem of trust. This area is also important for the Computer Science. When unfamiliar entities wish to cooperate, they do not know what to expect nor whether they can trust each other. Trust negotiation solves this problem by sequential exchanging credentials between entities, which have decided to establish a trust relationship in order to reach a common goal. Entities specify their own policies that handle a disclosure of confidential information to maintain their security and privacy. Policies are defined by means of a policy language. This paper aims to identify the most suitable policy language for trust negotiation. To do so, policy languages are analysed against a set of criteria for trust negotiation that are first established.

1 Introduction

Entities in our world, in order to live and enjoy the benefits of our civilisation, need to cooperate together and share many resources, such as knowledge, services, products and jobs. This sharing or exchanging requires fair conditions for all parties, so that everybody can use shared resources equally and receive fair consideration for each contribution. One of the biggest issues is trust. Entities need to know who they can trust when cooperating, to prevent deception and abuse. They need to be sure the others are going to behave as expected [1]. Trust is usually built up over a long period of time, as entities get to know each other better. In this case, the main constructor of trust is experience, which is gained with each interaction and if the output is positive, trust increases. This way trust is built step by step. An entity supposes that others generally do not change too much over time, so its previous experience with them is also relevant for the future. An entity expects others to provide similar outputs in the future, as they have provided in the past. This expectation, positive or negative, is de facto trust.

However, trust cannot always be established in this natural way. Nowadays with online environments, our communication capabilities have been widely extended and the number of participating entities is enormous. Because there are so many of them, their anonymity tend to increase. However, entities still need to be sure about the provision of their resources, information, etc. to other parties [2]. For this reason suitable approaches for establishing trust and maintaining confidence should be used. Trust negotiation establishes trust without the need of a direct previous experience with an entity. It is a credentials exchange process between two entities resulting in the establishment of trust [4, 5], where entities authenticate themselves by disclosing their private information. This information may be signed by an authority assuring their genuineness and authenticity. Generally, entities involved in trust negotiations need to control access over their data. This can be achieved by a definition of policies in a policy language. Policy languages provide a suitable way to express various types of policies and cover by them diverse aspects of trust negotiation, such as security and privacy. During trust negotiation, policies are read and evaluated in order to make credential disclosure decisions.

For the purpose of this paper, policy languages are observed for their suitability of usage for trust negotiation. This requires analysing the trust negotiation criteria that comprise a set of requirements needed to efficiently handle trust negotiation. The policy languages are analysed by classifying their attributes. Then, the attributes are checked against the identified criteria and if a match is found, the language is marked as supporting the criterion. The more criteria the policy language supports, the more suitable it is. Some criteria may be more important than others and some may even be essential. As a result, one or more policy languages suitable for trust negotiation can be identified. This paper identifies the general criteria for trust negotiation and analyses and classifies policy languages according to the selected criteria. It is important to carefully select the possibly suitable languages for trust negotiation that are to be classified. This classification is helpful to make a decision, which policy language should be chosen for a trust negotiation model. An engineer designing such model can view attributes of the analysed languages and thus select the most suitable language according to his needs.

The remainder of this paper is organised as follows: In Section 2, related work on trust, trust negotiation, policy languages and their classification is presented. Section 3 describes and explains the general concepts of trust negotiation and Section 4 identifies and presents the trust negotiation criteria. In Section 5, policy languages are analysed and matched against the criteria identified in Section 4. Finally, Section 6 concludes the paper and outlines future work.

2 Related Work

Entities experience trust on an everyday basis as they relate to each other and make decisions. Gambetta [1] defines trust as a subjective probability, by which an individual A expects another individual B to perform a given action, on which its welfare depends. This definition supposes that the trustor is dependent on the trustee and the trustee is reliable. According to Jøsang [2], there are two common definitions of trust called *reliability trust* and *decision trust*. Reliability trust can be interpreted as the reliability of something or somebody and decision trust as a magnitude of willingness of one entity to be dependent on another in a given scenario. The entity should feel relatively secure and comfortable about the other, even though it must accept possible negative consequences. Grandison and Sloman [3] define trust related to a given context: "Trust is the firm belief in the competence of an entity to act dependably, securely, and reliably within a specified context." From these definitions it is clear that the area of trust is quite diverse and it is difficult to define a single, standard and general trust definition covering all possible aspects and scenarios.

Trust establishment is a process of creating trust between two entities. Winsborough [4] claims that the mainstream approaches presume that the entities already know each other. Two standard approaches are used: *Identity-based*, when an entity is authenticated based on its known identity and *capability-based*, when an entity possesses capabilities needed by the requester. However, this approach does not work well in open systems, e.g. online environments, where the entities are anonymous and their attributes are unknown. In this case, trust negotiation can be used. It belongs to the trust-based decision-model concept [7] and is a process of incrementally establishing trust by exchanging credentials, one by one between two entities. The exchange process continues until the required trust level is reached [4, 5]. Credentials are private resources of an entity that contain sensitive information and can lead to identify the entity or to disclose facts about it. According to Winsborough [4] credentials or property-based digital credentials are the on-line analogues of paper credentials that people carry in their wallets. They appear to be well suited to establish trust in open systems. Credentials can also authenticate other entities, their properties and relationships. Ting Yu et al. offer another definition [6], which states that digital credentials are verifiable, unforgeable digitally signed assertions. They are signed by a credential issuer about the properties of the parties mentioned in the credential. They can also contain a public key of one or more of the parties they mention, so these parties can prove that the credentials describe them.

Policies are defined in policy languages and they are essential for trust negotiation, because they control access to credentials and protect the security and privacy of entities. In many cases policy languages are implementation dependent and there are no standard metrics to analyse and evaluate the effectiveness of these languages or to compare them [11]. Further attempts to classify policy languages have been made. Kasem and Meier [10] present an overview of languages that are suitable for security and privacy. They classify their attributes into four main categories, such as *type*, *intention of use*, *scope* and *design and implementation details*. The work in [11] classifies policy languages into the following categories: *sophisticated access control languages*, *web privacy policy languages*, *enterprise privacy policy languages* and *context sensitive languages*. Seamons et al. [12] present a classification that is aimed at trust negotiation. They propose many criteria that can be useful for trust negotiation, such as *credential combinations*, *sensitive policies*, *transitive closure* and so on. This work is similar to ours. The difference is that we chose different criteria for the language comparisons and we include more languages.

The first trust-based negotiation-model was TrustBuilder and TrustBuilder2 is its enhanced version [8, 9]. Another trust-based negotiation-model is PROTUNE and it is rule-based [25]. Trust negotiation is classified as a subclass of the trust-based decision-model concept [7]. The models can use various negotiating strategies. PROTUNE uses a cooperative default strategy, where all the relevant and releasable information are disclosed at each step. TrustBuilder allows the entities to choose the most suitable strategy for their needs. They can choose for example the eager or the parsimonious strategy [4].

This paper aims to identify trust negotiation criteria and to classify policy languages against them. Kasem and Meier classified PPL, A-PPL, P2U, PRML, SecPAL4P, XPref and XACML [14, 16, 17, 27, 24, 21, 13] as security or privacy languages, which makes them good candidates for the analysis. Seamons et al. chose for their classification of trust negotiation languages such as PSPL, TPL, X-Sec and the language of the KeyNote trust management system [28, 29, 30, 31], which makes them also good candidates. The other languages that seem to be valid for trust negotiation, are PlexC, the language of the Cassandra trust management system, X-TNL, ASL and HiPoLDS [18, 19, 20, 22, 26]. In the following, the language of Cassandra and the language of KeyNote will be referred as Cassandra and KeyNote, respectively. All these languages have been selected, because they have proven useful for keeping privacy of entities, the appropriate access control handling or for other models of credentials exchange.

3 General Concepts of Trust Negotiation

Trust negotiation belongs to the trust-based decision-model concept that uses defined rules and policies to control access to credentials and resources [7]. It can be defined as a process of incrementally establishing trust by exchanging credentials between two entities, while they may be complete strangers to each other [4, 5]. The entities share their credentials iteratively one by one. The first entity discloses one credential to the other one and thus builds a basic trust in the second entity and then the second entity discloses one credential to the first one achieving the same effect. The exchange process continues until the required trust level is reached.

The basic concept of the trust negotiation is depicted in Fig. 1. It shows two entities trying to establish trust in each other. Entity (1) requests a credential from another entity (2). If entity (2) is willing to do so, it discloses a credential

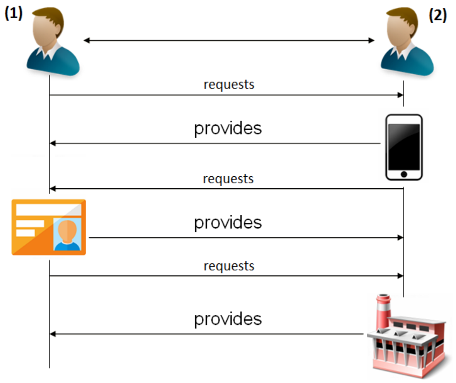


Figure 1: Trust negotiation basic concept

to entity (1) and by this action entity (1) might be willing to disclose another credential to entity (2). This way the entities exchange their credentials and build up trust in each other. The negotiation process continues until the desired level of trust is reached and the entities are willing to disclose new credentials. Once sufficient trust has been built, the trust negotiation successfully terminates. If an entity is not willing to disclose more credentials, depending on the negotiation strategy, it may be asked to provide alternative credentials or the whole negotiation process is terminated without being successful.

4 Trust Negotiation Criteria

The criteria are a set of requirements that will guide the whole process of trust negotiation. We identified, analysed and collected the generally accepted requirements for trust negotiation from the literature. We were looking for the important criteria that are needed to accomplish or that can simplify the process of establishing trust between two entities. The requirements are defined quite generally as they should cover wide areas that can be further divided and can be mapped to diverse existing policy languages. Entities possess credentials and various resources that may contain private and sensitive data. For that reason, the requirements must ensure that the resources are protected and the conditions under which they can be accessed must be clearly defined. A set of requirements may form a policy that is a statement of intent to guide decisions and achieve rational outcomes. Entities in trust negotiation must be able to specify their own policies defining access to their private resources. For example, some negotiation strategies, such as the eager strategy, use a concept of locked and unlocked credentials, where only the unlocked credentials can be disclosed and the locked credentials may become unlocked after receiving new credentials from the other entity [4]. The following requirements can be identified for trust negotiation:

- ***Privacy of resources.*** This is an important requirement of trust negotiation. An entity must be confident that access to its private data will not be abused. It should not be possible to obtain the protected resources through a swindle, e.g. by providing forged credentials. The private data must not be intentionally modified by a third party. The access policy itself can lead to facts about obtaining access to credentials, so it should be protected.

- ***Access control to resources.*** This requirement partially covers the need for privacy, which is natural as it comes from a proximity of this criterion to the previous one. Entities need to manage access to their private data, such as credentials. The access control should define the conditions under which resources can be accessed. The access control management should be simple but efficient enough and it should be transparent to the entity, so the entity can be sure that its confidentiality will not be compromised.

- ***Usage control of resources.*** This requirement is referred as usage control, however it is understood more broadly, like a general compliance. It partially covers the need for privacy too, which comes from its nature. As entities exchange their credentials, they establish trust. However, an entity may discover that the other one is not playing according to the rules, e.g. it is providing false information or forged credentials. In this case the entity may decide to stop the process of building trust and terminate the trust negotiation. The entity may mark the other one as a cheater and may refuse to cooperate with it in the future.

- ***Exchange of resources.*** This criterion is very important for trust negotiation and also for a policy language in order to allow definitions of policies suitable for the exchange process and to support trust negotiation efficiently. The principle of trust negotiation means to exchange credentials that are the private and confidential resources of their owning entities. An entity must be provided with a secure and straightforward way of passing its credentials to another entity. An entity may use credentials and certificates exchange with an authority to verify their authenticity.

- ***Authority.*** Trust negotiation may require access to an authority for the validation of credentials. In case of doubts during a trust negotiation, entities can independently communicate with a local or global authority where they can verify credentials provided from the other entity involved. The rejection of credentials has implications. The use of authorities is not mandatory for all scenarios in trust negotiation, however sometimes it can be very helpful.

- ***Information granularity.*** Credentials exchanged during trust negotiations may comprise various information levels. The information contained in a credential may be too detailed and for the actual needs it would be enough to disclose only a part or a more general, less detailed version. For example, rather than disclosing the exact location information by providing the GPS coordinates, the region or the city name would be provided instead. It may be useful to support information granularity for trust negotiation, because the credential confidentiality levels can be controlled during their disclosure and thus protect the security and privacy of their owner.

- ***Context sensitivity.*** Trust negotiation should take place in the context of an intended goal. Entities have diverse capabilities and they possess various skills in different fields. For this reason, the trust built is context-sensitive and it should only be used for the defined goal or for goal-related purposes.

- ***Roles.*** Roles are related to context sensitivity. Entities play various roles that may determine attributes of the entities and the purpose of the entity in trust negotiation. Roles can also determine the access control to the resources of an entity.

One of the most important requirements is the exchange of resources, which is implicit in the nature of trust negotiation, where credentials are exchanged between entities. Other important requirements are privacy of resources and access control to resources that handle privacy and protection of sensitive data. Context sensitivity is important too, if the intended trust relationship is considered to be established for a specific purpose. The rest of the requirements may enrich and simplify trust negotiation in specific scenarios.

5 Analysis of Policy Languages

The analysis of potentially suitable policy languages for trust negotiation is presented here. The languages were chosen if their attributes seemed to be helpful, e.g. they supported security or privacy. The languages will be analysed to classify their features and then these features will be checked against the trust negotiation criteria identified in Section 4. If a match is found with a particular criterion, the language is marked supporting trust negotiation.

5.1 Privacy of Resources

The maximum privacy preservation would be to not disclose any credentials or resources to anyone. This approach is not desirable as for trust negotiation information exchange is essential. Privacy preservation contradicts the requirement of exchanging resources. Therefore, it is important to find a trade-off between the two. To preserve privacy, it is important to disclose credentials sequentially and alternately, as they are received from the other negotiator. This approach is used by P2U [17], PlexC [18], Cassandra [19], X-TNL [20], HiPoLDS [26] and PSPL [28]. They can be configured to initially exchange fewer confidential credentials and as trust builds, more confidential ones are disclosed, thereby protecting privacy. However, we do not consider X-TNL, HiPoLDS and Cassandra to satisfy this privacy criterion as they neither explicitly provide any expressions of privacy policies nor include the preservation of privacy in their design. Other languages such as PPL [14], A-PPL [16], P2U, PlexC, XPref [21], SecPAL4P [24], PRML [27] and PSPL do support the definition of privacy policies. They can be used to define the accepted maximum confidentiality level and in that way control the exposure of credentials. Each language takes a different approach. Some languages, such as PPL, A-PPL, P2U and PRML use privacy specification elements, where policies express the privacy relationships among them. Other languages, such as XPref and SecPAL4P allow entities to specify their privacy preferences, which handle the way of treating their sensitive information by a service. PlexC ensures privacy by a minimisation of the over-exposure problem. When an entity reaches the desired acceptable exposure area, the credentials disclosure risk is minimised and its privacy is preserved. This provides the best approach for privacy, because it tries to keep the exposure within desired boundaries. This is also done through a given feedback, if the boundaries are exceeded. Other languages do not provide a feedback about privacy abuse. PSPL focuses on privacy preservation of clients and servers by avoiding unnecessary disclosures. The rest of the languages, such as XACML, ASL, TPL, X-Sec and KeyNote do not provide any privacy-preserving features.

5.2 Access Control to Resources

Access control is supported by all policy languages, because each one provides a data protection and an authorised access. This criterion is given by the nature and proposed operation of policy languages. Only credentials with a certain confidentiality level can be disclosed to preserve privacy and only to the authorised entity. The attribute-based access control (*ABAC*) is a basic one and is supported by XACML [13], PPL [14] and A-PPL [16]. These languages define access rights through attribute-combining policies with the use of Boolean logic and use triggers, which are events filtered by conditions, related to an obligation. This paradigm is suitable for defining disclosure policies with respect to negotiators. It combines entity attributes with Boolean logic to implement an exchange model. The values, such as the credential confidentiality level or a list/number of disclosed and received credentials should be defined. The role-based access control (*RBAC*) represents another approach to access control. Disclosure policies are defined based on the entities' roles. However, complete strangers may interact with each other and if they do not take on any roles, it is impossible to make control decisions based on them. The role-based access control is used by PRML, TPL and Cassandra, where entities are mapped into roles e.g. based on issued credentials by third parties. When an entity requests permission to perform an action, its role is checked against a policy in order to permit access. Yet another approach to access control are the privacy policies used by P2U, PlexC, XPref and SecPAL4P. Access to a sensitive resource is determined based on the entity's privacy preferences. They are highly suited to trust negotiation, because an entity can maintain its desired disclosure level to protect its privacy. KeyNote uses a compliance checker that controls the disclosure of credentials. The compliance checker is an appealing solution, because it handles the disclosure decisions on behalf of the entity. A similar approach is to use personalised access rights with respect to an authorised entity. PSPL defines a client-server access control, X-TNL uses disclosure and certificate policies, ASL defines policies and authorisations managing the access control decisions, HiPoLDS defines policy domains describing a global system architecture and X-Sec manages access control to web documents. The languages, such as PlexC, X-TNL, SecPAL4P, PSPL, TPL, X-Sec, Cassandra and KeyNote make use of authorities to handle access control decisions.

5.3 Usage Control of Resources

During trust negotiation, the entities involved exchange of credentials. The problem is that once credentials have been disclosed, their owner loses control over them. Access control policies define under which conditions credentials can be disclosed. Usage control policies specify how the disclosed credentials can be used by the recipient. Usage control policies are not mandatory for trust negotiation, however, they can improve an entity's privacy in the case that it obtains a credible feedback about its credentials usage. PlexC uses an *exposure control loop* that provides a periodical exposure feedback to entities about the conceded access paths to their resources and how they are being shared. This feedback is then used to adjust the entity policies over time in order to prevent over-exposure in the future. It is called exposure polymorphism [18]. Over-exposure is classed as when too many credentials are disclosed and their overall confidentiality is too high. PlexC may catch this situation and accordingly revise policies to prevent too many disclosures in a future trust negotiation. However, the exposure control is dynamic, non-trivial and some input values, such as the credential use after the disclosure, may be unknown. Therefore, PlexC tends to revise the policies continuously and to catch small changes in feedback. The rest of the languages neither control nor monitor credentials once they have been disclosed.

5.4 Exchange of Resources

Generally, policy languages supporting this criterion can be considered as usable for trust negotiation. The exchange of resources is a must for trust negotiation, because it is a given consequence of the process itself. Entities exchange credentials during trust negotiation and it is important that the exchange process is gradual and balanced to preserve privacy. Resources may be exchanged in large and distributed networks, such as the Internet. PlexC and Cassandra permit a trust negotiation scenario to be easily implemented in these networks. The languages, P2U, PlexC, Cassandra, X-TNL, HiPoLDS and PSPL are suitable for trust negotiation, because they allow rules for credentials exchange to be defined and controlled. Credentials are exchanged sequentially and alternately, which ensures a balanced privacy exposure of both parties. Each language can be used with a negotiation strategy that handles the exchange process, calculates the established trust from the credentials received and controls the disclosure of credentials based on their confidentiality. The strategy is not defined in the language itself, but rather in the system using the language. The Cassandra trust management system uses a strategy similar to the "Parsimonious Strategy" [4]. It handles the exchange process itself through agents thereby removing this responsibility from the user. PlexC and Cassandra are designed to process trust negotiations in large networks, which introduces security and privacy issues. PlexC reduces them by the over-exposure control and Cassandra follows well-defined conditions, defined by the local access control policies. The languages, P2U, HiPoLDS and PSPL use a different approach to exchange resources. P2U facilitates a user data sharing and negotiation over various applications, HiPoLDS uses reference monitors to control the exchange process among policy domains and PSPL exchanges credentials and declarations between clients and servers. The rest of the languages, XACML, PPL, A-PPL, XPref, ASL, SecPAL4P, PRML, TPL, X-Sec and KeyNote do not support this criterion and thus are not suitable for trust negotiation. However, TPL is extensible and provides other suitable criteria for trust negotiation, such as access control, authority and roles, so it could be extended to include this criterion.

5.5 Authority

Trust negotiation may require the presence of one or more authorities. A trusted certification authority serves to issue and verify credentials, which, in turn makes the negotiators more confident about the credentials' authenticity. A general authority is referred as it may stand for different types of authorities, however, the analysed languages do not specify the exact type of a certification authority. It is supported by a few languages, such as X-TNL, SecPAL4P, PSPL, TPL and X-Sec. This is the basic use of authority and it can be helpful for entities involved in trust negotiation to verify the validity and the genuineness of credentials. X-TNL, TPL and X-Sec allow authorities to be organised into categories. In X-TNL and X-Sec, credentials and declarations form certificates that are collected into X-Profiles. TPL organises certificates into certification profiles and is able to automatically collect missing certificates from peer servers. Some languages, such as PlexC, SecPAL4P, Cassandra and KeyNote support a delegation of trusted decisions or actions, where the trustee can act on behalf of the trustor. These languages are designed for large decentralised networks, such as the Internet. Delegation of authority is useful here, because the requester and the authoriser may not have established a trust relationship. A disclosure decision of a trustor may be inspired by a disclosure decision of a trustee. The delegation of authority, trusted decisions or actions is also suitable for trust negotiation, because it allows entities to delegate disclosure decisions to trusted parties, such as security agents. The Cassandra trust management system uses trusted agents to control the credentials exchange process. These agents take responsibility for actions and decisions that are delegated to them by trustors. The rest of the policy languages, XACML, PPL, A-PPL, P2U, XPref, ASL, HiPoLDS and PRML do not support this criterion.

5.6 Information Granularity

Credentials possess various confidentiality levels depending on, for example, their importance. When a credential is disclosed, its owner's privacy is automatically compromised, as far as the confidentiality allows. PlexC allows information about an entity to be disclosed with different accuracy levels. The degree of information provided can be defined by policies. For example, an entity can provide its precise position with GPS coordinates, or less accurately by disclosing only a region or a city name. Additionally, rules can be defined based on the current time or location. For example, the access to resources can be permitted, only when the entity is located in a certain region or for a certain period of time. Data can be shared with more or less precision so as to preserve privacy. PlexC may revise the disclosure policies to provide less precise and therefore less confidential information, when the entity during trust negotiation reaches the over-exposure area. The revision of policies is controlled by PlexC automatically. The rest of the languages do not explicitly support any form of different accuracy levels. They understand credentials disclosure as a binary operation, so a credential is either disclosed completely or not at all. If an entity wants to provide less confidential information, it must do so for itself e.g. by dividing the credential into parts.

5.7 Context Sensitivity

Entities that want to establish trust with each other, do so to accomplish a common goal, for a specific purpose. Each entity possesses different knowledge, abilities, skills and resources, for which the entity can be trusted to successfully participate in reaching the goal. The entity might not be trusted for another purpose, because different attributes would be required. Therefore, trust negotiation examines important attributes for the intended purpose, for which the entity can be trusted. One approach is to define purpose-dependant conditions, under which credentials can be shared or disclosed. This is the case of PPL, A-PPL, P2U, PlexC and XPref. On the other hand, PRML defines such conditions in order to control data operations, i.e. what operation on which data can be performed. This approach is more general, because it allows an entity to define its own purpose-dependant data operations. All these languages allow the purpose for the credentials exchange to be defined during the trust negotiation. If a credential is demanded for a different purpose then this is specified in the disclosure policy and the access may be denied. Concretely, PPL and A-PPL define authorisation types that can use resources only for a particular set of purposes, P2U uses the *purpose-relevance-sharing* principle, where only the relevant resources to the specific purpose and context of use are shared, PlexC defines context-dependant policies that influence the access traces to private data and XPref and PRML specify a purpose object for the same reasons. Unlike the others, PPL and A-PPL allow a hierarchy of the purpose elements to be created, so that Boolean logic can be applied to them, which improves and simplifies the purpose-relevant access decisions. For example, the parent-purpose element may specify that a credential may be disclosed only for establishing trust and its child-purpose element may specify in which context. The credential will be disclosed if both of the conditions are satisfied. Some languages, such as A-PPL, P2U and XPref, allow a retention value to be defined, which specifies the duration of access to a resource. This feature improves security and privacy of the resource owner. After the defined time period, the resource will be deleted. The rest of the languages are not context sensitive and do not allow purpose-relevant conditions to be specified.

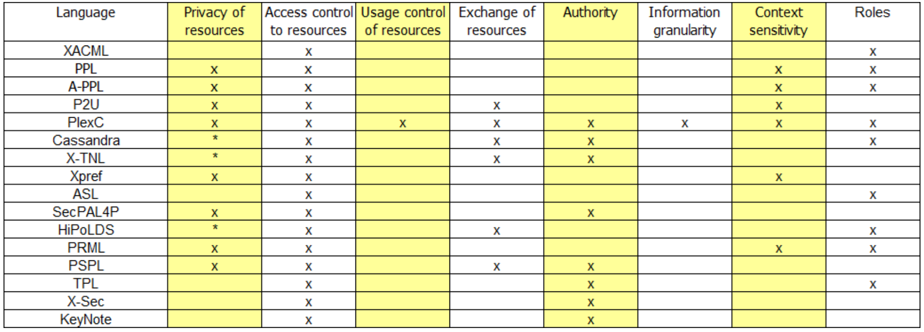
5.8 Roles

As mentioned earlier, entities take part in trust negotiations to reach a common goal. Only those entities with suitable attributes become trusted to accomplish it. Entities may take on roles that are associated with their attributes, e.g. a certain role requires that an entity possesses certain features. These roles can be specified by some languages, such as XACML, PPL, A-PPL, PlexC, ASL, HiPoLDS, PRML, TPL and Cassandra. Some of them, such as XACML, PPL and A-PPL already contain some specific, predefined roles that entities can have and that control access to their resources. However, all these languages allow entities to define a new role and to perform a particular action based on it. In PlexC, roles can be assigned to a group with given permissions, which simplifies the permission management and is useful for large networks. Cassandra supports auxiliary roles that can express some attributes of their owner and can be used without an active role. HiPoLDS defines roles by assigning policy domain attributes to policy domains representing entities. PlexC, TPL and Cassandra can use an authority to issue and verify certificates about the assigned roles. The languages, ASL, PRML and Cassandra can form a hierarchy of roles. This enables a role combination to be easily defined in order to perform an action or disclose a credential. As occurs for the context sensitivity, Boolean logic is applied when forming the hierarchy of the roles. PRML allows a role to extend over multiple other roles and to inherit their permissions. Unlike the others, Cassandra supports a role retention, which means that a role validity period can be defined. After its expiration the role is no longer valid and in consequence, obtaining credentials during trust negotiation can be refused. This feature improves the security and privacy of negotiators.

5.9 Analysis Summary.

After the policy languages analysis is performed in Section 5 we can summarise our findings in Table 1. A partially supported criterion is defined as a criterion that was found as a secondary effect of other criteria supported by the language, but was not explicitly mentioned in the literature nor its presence was intended or designed by the authors of the language.

Table 1: Supported trust negotiation criteria



Language	Privacy of resources	Access control to resources	Usage control of resources	Exchange of resources	Authority	Information granularity	Context sensitivity	Roles
XACML		x						x
PPL	x	x					x	x
A-PPL	x	x					x	x
P2U	x	x		x			x	
PlexC	x	x	x	x	x	x	x	x
Cassandra	*	x		x	x			x
X-TNL	*	x		x	x			
Xpref	x	x					x	
ASL		x						x
SecPAL4P	x	x			x			
HiPoLDS	*	x		x				x
PRML	x	x					x	x
PSPL	x	x		x	x			
TPL		x			x			x
X-Sec		x			x			
KeyNote		x			x			

Legend: ×: supported
*: partially supported

Each of the languages was originally designed to solve another type of problem. Some of them already included some form of trust negotiation in their design, such as P2U, PlexC, X-TNL, HiPoLDS, PSPL and Cassandra. The best one for trust negotiation seems to be PlexC, as it is the only one to support all of the identified criteria in Section 4. PlexC takes an interesting approach. It introduces the exposure control problem and claims that there is an area of acceptable exposure. Entities try to eliminate the over-exposure of their data and tend to transform it into the most acceptable exposure that does not expose them to a major risk. For trust negotiation this means that only a minimal set of credentials will be disclosed to reach the required level of trust. The other highly recommended languages are P2U and Cassandra. P2U focuses on the purpose of data sharing, so the context is important. It simply defines the data provider, the data consumer and the relevant policies for the data exchange. Cassandra acts as a local service and it is completely decentralised. It supports roles and actions that are performed over these roles. In this way each entity has total control over its resources and policies in trust negotiation.

The languages, X-TNL, HiPoLDS, PSPL and TPL support the criteria only partially. They can be used for a special or simple case of trust negotiation, but they lack its general support. All the languages except TPL support the exchange of resources, which is essential. TPL supports access control, authority and roles, but can be further extended to the exchange of resources. X-TNL although originally designed for trust negotiation, lacks some criteria, such as roles, context sensitivity and partial privacy of resources. HiPoLDS and PSPL were designed for different purposes, but their capabilities could also be used for trust negotiation. PSPL is an expressive and extendible language. The remainder of the languages, XACML, PPL, A-PPL, XPref, ASL, SecPAL4P, PRML, X-Sec and KeyNote serve specific purposes that are not concerned with trust negotiation. XACML is an attribute-based access control system, PPL and A-PPL are extensions over XACML with data handling and protection capabilities, XPref and SecPAL4P serve for users to define their privacy preferences, PRML merges the corporate privacy policies and the data handling policies, ASL serves for expressing authorisations, X-Sec protects web documents and KeyNote handles authorisations in decentralised environments.

6 Conclusion

In this paper we have analysed policy languages to check their suitability for trust negotiation. In order to do so, we have first identified the following criteria: privacy of resources, access control to resources, usage control of resources, exchange of resources, authority, information granularity, context sensitivity and roles. We believe them to be quite a complete list of general criteria for trust negotiation regardless that they can be refined in the future for specific purposes.

Then, the policy languages have been analysed against them. From this analysis, it has emerged that only PlexC is fully suited for trust negotiation. PlexC was found to be the only one from all the languages analysed that supports all of the identified criteria for trust negotiation. Due to its completeness and flexibility, PlexC is a good candidate to be used in the Internet of Things (IoT) trust negotiation scenarios.

A subset of the chosen languages were suitable in part, because they generally support the exchange of resources, but lack the other possible criteria demanded by trust negotiation. The rest of the languages are not suitable, because they lack the essential criterion for trust negotiation, which is the exchange of resources and other possibly important requirements too.

In the future work, the identified criteria will be divided into more fine-grained criteria, if needed for specific purposes, that could match the analysed policy languages more precisely. The current criteria are quite broad, so it is a good idea to make another identification of more specialised criteria important or useful for trust negotiation. They will form a subclass of the currently identified criteria. In addition, other criteria although not directly related to trust negotiation could be taken into consideration, such as the languages syntax and user-friendliness.

Acknowledgements

This research has been supported by the European project “European Network for Cyber-security (NECS)” - the European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 675320 and the Spanish Ministry of Economy and FEDER through the project PRECISE (TIN2014-54427-JIN).

References

[1] D. Gambetta, Can We Trust Trust?, D. Gambetta (Ed.), Trust: Making and Breaking Cooperative Relations, B. Blackwell, Oxford, 1990, pp. 213 238.

[2] A. Jøsang, R. Ismail and C. Boyd, A Survey of Trust and Reputation Systems for Online Service Provision, Decision Support Systems, 2007, vol. 43 pages 618-644

[3] T. Grandison and M. Sloman, A Survey of Trust in Internet Applications, Commun. Surveys Tuts., 2000

[4] W. H. Winsborough, K. E. Seamons and V. E. Jones, Automated Trust Negotiation, DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings, 2000, 1, 88-102 vol.1.

[5] W. H. Winsborough and N. Li, Towards Practical Automated Trust Negotiation, Proceedings Third International Workshop on Policies for Distributed Systems and Networks, 2002, 92-103.

[6] T. Yu and M. Winslett, A Unified Scheme for Resource Protection in Automated Trust Negotiation, 2003 Symposium on Security and Privacy, 110-122, 2003.

[7] F. Moyano, Trust Engineering Framework for Software Services, PhD thesis, Lenguajes y Ciencias de la Computacin, Universidad de Mlaga, 2015.

[8] M. Winslett, T. Yu, K. E. Seamons, A. Hess, J. Jacobson, R. Jarvis, B. Smith and L. Yu, Negotiating Trust in the Web, Internet Computing, IEEE, vol. 6, no. 6, pp. 30-37, 2002.

[9] A. J. Lee, M. Winslett and K. J. Perano, TrustBuilder2: A Reconfigurable Framework for Trust Negotiation. No. SAND2007-1928C. Sandia National Laboratories (SNL-CA), Livermore, CA (United States), 2007.

[10] S. Kasem-Madani and M. Meier, Security and Privacy Policy Languages: A Survey, Categorization and Gap Identification, arXiv preprint arXiv:1512.00201, 2015.

[11] P. Kumaraguru, et al., A Survey of Privacy Policy Languages, Workshop on Usable IT Security Management (USM 07): Proceedings of the 3rd Symposium on Usable Privacy and Security, ACM. 2007.

[12] K. E. Seamons, et al., Requirements for Policy Languages for Trust Negotiation, 2002 IEEE Proceedings of the Third International Workshop on Policies for Distributed Systems and Networks, 2002.

[13] B. Parducci and H. Lockart: eXtensible Access Control Markup Language (XACML) 3.0, Committee Specification 01, 10 August 2010.

[14] C. A. Ardagna, et al., Primelife Policy Language, W3C Workshop on Access Control Application Scenarios. W3C, 2009.

[15] S. Trabelsi, et al., PPL Engine: A Symmetric Architecture for Privacy Policy Handling, W3C Workshop on Privacy and Data Usage Control. Vol. 4. No. 5. 2010.

[16] M. Azraoui, et al., A-PPL: An Accountability Policy Language, Data Privacy Management, Autonomous Spontaneous Security And Security Assurance. Springer, Cham, 2015. 319-326.

[17] J. Iyilade and J. Vassileva, P2U: A Privacy Policy Specification Language for Secondary Data Sharing and Usage, 2014 IEEE Security and Privacy Workshops, 2014.

[18] Y. L. Gall, A. J. Lee and A. Kapadia, PlexC: A Policy Language for Exposure Control, Proceedings of the 17th ACM Symposium on Access Control Models and Technologies, 2012.

[19] M. Y. Becker and P. Sewell, Cassandra: Distributed Access Control Policies with Tunable Expressiveness, Policies for Distributed Systems and Networks, 2004. POLICY 2004. Proceedings. Fifth IEEE International Workshop on. IEEE, 2004.

[20] E. Bertino, E. Ferrari and A. Squicciarini, X-TNL: An XML-based Language for Trust Negotiations, Proceedings POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks, 2003.

[21] R. Agrawal, J. Kiernan, R. Srikant and Y. Xu, XPref: A Preference Language for P3P, Computer Networks, 2005.

[22] S. Jajodia, P. Samarati and V. S. Subrahmanian, A Logical Language for Expressing Authorizations, Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No.97CB36097), 1997.

[23] J. Clark and S. DeRose, XML Path Language (XPath) Version 1.0. W3C Recommendation, 1999.

[24] M. Y. Berker, A. Malkis and L. Bussard, A Framework for Privacy Preferences and Data-Handling Policies, Technical Report MSR-TR-2009-128, 2009.

[25] P. A. Bonatti, J. L. De Coi, D. Olmedilla and L. Sauro, A Rule-Based Trust Negotiation System, IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 11, 2010.

[26] M. Dell'Amico, et al., HiPoLDS: A Hierarchical Security Policy Language for Distributed Systems, Information Security Technical Report 17, 81-92, 2013.

[27] PRML: Privacy Rights Markup Language Specification Version 0.9, Zero-Knowledge Systems, 2001.

[28] P. Bonatti and P. Samarati, Regulating Service Access and Information Release on the Web, 7th ACM Conference on Computer and Communications Security, Athens, Greece, November 2000.

[29] A. Herzberg, Y. Mass, J. Mihaeli, D. Naor and Y. Ravid, Access Control Meets Public Key Infrastructure, Or: Assigning Roles to Strangers, Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000, Berkeley, CA, 2000, pp. 2-14.

[30] E. Bertino, S. Castano and E. Ferrari, On Specifying Security Policies for Web Documents with an XML-based Language, Sixth ACM SACMAT, Chantilly, Virginia, May 2001.

[31] M. Blaze, J. Feigenbaum, J. Ioannidis and A. Keromytis, The KeyNote Trust-Management System Version 2, RFC 2704, September 1999.