The Second International Conference on Emerging Security Information, Systems and Technologies

Towards Decentralised Security Policies for e-Health Collaborations

Oluwafemi Ajayi, Richard Sinnott and Anthony Stell
National e-Science Centre
University of Glasgow
G12 8QQ, United Kingdom
{o.ajayi, r.sinnott, a.stell}@nesc.gla.ac.uk

Abstract

Security in decentralised collaborative environments present huge challenges where many entities from different autonomous security domains want to access and share resources. This is largely due to cross-boundary issues where security credentials and policies are heterogeneous, and where yielding control to a centralised authority is not an option. Numerous cross-boundary approaches exist today and trust negotiation remains a promising solution that is rapidly evolving. In this paper we present dynamic trust negotiation, an approach that folds remote security credentials into local security credentials through trust contracts, thereby bridging the gap making decentralised security policies for multi-domain collaboration difficult. We show how trust can be realised between strangers through trusted intermediaries where direct trust negotiation between these strangers is otherwise unacceptable.

1 Introduction

In today's internet age where information resources and services are widely spread and available, the need to share and collaborate cannot be more emphasised. In open environments various applications and tools have been developed (and are still undergoing development) that encourage sharing of resources and information to the advantage of everyone involved. This allows everyone to freely access and use information since no single point of access exist. Collaboration by definition implies decentralisation as no leadership is required, rather co-ordination in working towards common goal. Collaboration in specialised areas such as e-health research can lead to greater access to resources and recognition and reward, than when individual work with limited knowledge and resources in isolation.

For such specialised areas, the need to protect and control resources is essential. However, security in decentralised collaborative environments present many challenges where many entities from different autonomous security domains want to access and share resources. This is largely due to cross-boundary issues where security credentials and policies are heterogeneous, and where yielding control to a centralised authority is not an option.

Currently, numerous organisations and bodies are coming together to establish standards to tackle this problem, e.g. through development of security languages and protocols for credentials and policies. However, these standards can only satisfy one part of the problem. A key issue is related to meaning. Attributes encapsulated in credentials may have different meanings between organisations as permissions and objects (resources) are different between organisations. These permissions are mapped to attributes and are encoded in policies to control access to resources. Thus, there is the need to fold or map remote attributes to local attributes and vice-versa. Otherwise organisations would have to incur an overhead in managing policies that are aware of remote attributes. This is not feasible or scalable for many systems.

Every organisation seeks to maintain control over credentials and resources. However, some cross-boundary solutions use centralised access control lists where collaborating partners have negotiated and agreed on privileges and resources amongst other things. Other solutions involve delegating some responsibility of access right management to trusted remote individuals in assigning privileges to their (remote) users [14, 15]. These solutions require negotiations and delegations but are often constrained by a number of factors such as changing requirements and difficulty in reaching or maintaining agreements [5].

Solutions such as these bring to the forefront the issue of trust. Specifically, trust realisation between organisations, individuals, entities or systems that are present in multi-domain authorities and multi-policy enforcement points. One approach that promises trust realisation is trust negotiation [18]. Trust Negotiation provides a means of establishing trust between strangers (non trusted entities) through an iterative but cautious disclosure of digital cre-

978-0-7695-3329-2/08 $25.00 © 2008 Crown Copyright
DOI 10.1109/SECURWARE.2008.15

dentials [8, 16].

In this paper we present dynamic trust negotiation (DTN) [4], an approach that folds remote security credentials into local security credentials through trust contracts, so bridging the gap that makes decentralised security policies for multi-domain collaboration difficult. We show how trust can be realised between strangers through trusted intermediaries where direct trust negotiation between strangers is otherwise unacceptable. In Section 2 we review trust negotiation and DTN. Section 3 describes *circle of trust* and *trust contracts* while section 4 discusses DTN policies. Section 5 presents the DTN architecture, implementation and performance. We conclude in section 6.

2 Background

Trust is the underlying phenomenon of any security system. Most security systems are designed using security policies, which define and describe what is trusted, how it is trusted and where it is trusted. Trust is built on the concept of limiting expected behaviour [7]. It is associated with an assurance measurement. That is, the level of confidence in limiting behaviour within a security policy determines the level of assurance.

2.1 Automated Trust Negotiation

Automated trust negotiation (ATN) [18] is the process of establishing trust between strangers through the exchange of digital credentials. These credentials are sensitive information and are often protected through the use of disclosure policies. These disclosure policies inevitably require negotiation strategies as each entity tries to protect what credentials are released. However for a negotiation to succeed entities are expected to operate using the same family of disclosure strategies[20].

Different trust negotiations approaches have been proposed to support access control policies in open decentralised environments [17, 9, 19, 12, 10]. Some approaches are based on a trust negotiation framework in the context of a peer-to-peer environment. [19] introduces a *locally trusted third party* (LTTP) which acts like a cache and mediator between two entities for the purpose of successful trust negotiations in peer-to-peer systems. Similarly [9] introduces a *sequence prediction module* that caches and manages used credential sequences from previously trust negotiations. While [12] proposes a trust chain based negotiation strategy (TRANS), which dynamically constructs trust relationships using a *trust proxy* that can cache common credentials or partial trust chain information from previous negotiations.

Various ATN systems have been developed, they include Trust-X [9] and TrustBuilder [21]. Trust-X is a framework that provides an XML-based language that is used to encode policies and certificates for trust negotiations. It also provides a peer-to-peer architecture used for negotiation management. TrustBuilder is an architecture that focuses on negotiation strategies. The architecture verifies credentials and checks policy compliance. Other systems like Traust [11] have been developed to augment TrustBuilder to provide interaction between applications or systems that offer trust negotiation services.

2.2 Dynamic Trust Negotiation

The sensitive nature of clinical data makes security a high priority and any method of federating this data must adhere rigourously to the local security policies that protects this data. In addition, health providers like the NHS are only willing to interact with parties they have explicit contracts with [1]. This makes it especially difficult to support ATN as they are unlikely to deal directly with strangers owing to risks involved. However when an intermediary party is introduced that is known to both parties (strangers) the associated risks are reduced since credentials need not be disclosed to strangers.

Dynamic trust negotiation (DTN) formalised in [4], is the process of realising trust between strangers or non-trusting entities (e.g. domains), through locally trusted intermediary entities. Trust is realised when an entity delegates its digital credentials to trusted intermediary entities through which it can interact with non-trusted entities. These intermediary entities can in turn delegate to other intermediary entities resulting in what we call *n-tier* delegation hops. The trust negotiation process involves trust delegations through intermediary trusted entities on behalf of non-trusting entities, where *direct* trust negotiation with non-trusted entities is unacceptable. Any entity can serve as a negotiator for other entities provided it is trusted by the two non-trusting entities or by their intermediaries.

DTN explores how credentials can be negotiated as the basis to support collaborative research between autonomous, distributed resources. It addresses the heterogeneous and autonomous issues of trust management like credentials and policies in multi-domain environments. DTN negotiates credentials between trusted parties also known as a *circle of trust (COT)* [4], who act as mediators on behalf of strangers and thus bridge trust gaps. This bridge also reduces the risk associated with disclosing policies to strangers.

As an example of *circle of trust*, consider the following scenario. Jane from the Glasgow Royal Infirmary hospital - hereafter referred to as domain GRI - is an investigator on a cancer clinical trial. She wants to recruit patients onto specific trials and in doing so needs to query patient consented health records in Scotland. To achieve this, she logs in to a

trial portal and her credentials (privileges/attributes/roles...) are pulled from her domain, e.g. through Shibboleth pull or push from the portal service provider or GRI identity provider respectively. The trial portal initiates a credential negotiation request with all other domains that GRI trust such as Southern General Glasgow hospital (SGG). SGG returns patient records that satisfies GRI request based on Jane's credentials and delegated privileges at SGG. SGG also negotiates with other domains it trusts such as Royal Infirmary Edinburgh (RIE) using Jane's SGG delegated privileges. Similarly, RIE negotiates with other domains it trusts using SGG's RIE delegated privileges. Thus GRI, SGG, RIE are *trust pathways*. The request process continues with nodes joining the trust pathways until all possible trust paths are exploited. These negotiated credentials such as $RIE.investigator$ are forward to GRI, which then makes a query request with these credentials on behalf of Jane.

$$
\begin{aligned}
GRI.investigator &\leftarrow Jane \\
GRI.circleOfTrust &\leftarrow SGG \cup SGH \cup GRH \\
SGG.circleOfTrust &\leftarrow RIE \cup IRH \\
GRI.investigator &\leftarrow SGG.delegatedInvestigator \\
&\quad \cap RIE.investigator
\end{aligned}
$$

where Southern General Hospital is referred to as SGH; Gartnavel Royal Hospital as GRH; and Inverclyde Royal Hospital as IRH.

DTN differs from ATN in that (1) negotiation occurs between trusted parties and not between strangers even though the goal is to realise trust between strangers; (2) it introduces multiple hops and delegation into the trust negotiation, which resolves some heterogeneity issues, and (3) it limits disclosure of access control policies, which reduces the need for disclosure strategies. However ATN can be used between trusted parties to negotiate for new trust contracts.

3 Dynamic Trust Negotiation Model

In this section we describe two features of our trust negotiation model: *cirlce of trust* and *trust contract*.

3.1 Circle of Trust

In the formalised model of DTN, the concept of *circle of trust (COT)* [4, 6] for trust negotiation is introduced. Figure 1 describes a *COT*, which is a network of locally trusted intermediary peers that a peer (or entity) trusts and collaborates with through one or more trust-contracts between each peer. A trust contract *TC* is an agreement that exists between two entities. This network of trusted peers enable interactions between peered and non-peered domains.

Through overlapping *COT*s a trust-pathway (chain of trust) can be discovered. Consider two peers P_1 and P_5, where P_1 is a requester and P_5 is a resource provider in another domain. P_1 and P_2 has $\{P_3, P_4, P_6, P_7\}$ and $\{P_3, P_4, P_5\}$ in their *COT* respectively. For P_1 to access P_5 resources, they will need to be trusted by P_2. In addition, P_2 will need to understand and trust credentials from P_1. Since P_1 has trust relationships with $\{P_3, P_4\}$, which are also in trust relationship with P_2, P_1 will initiate a trust negotiation with P_2 through $\{P_3, P_4\}$. Similarly, P_2 will initiate a trust negotiation with P_5. Thus $\{P_3, P_2\}$, $\{P_4, P_2\}$ are trust-pathways between P_1 and P_5. Hence trust is realised by exploring overlapping *COT*s between P_1 and P_5.

$$P_1 \leftarrow (P_3 \vee P_4) \leftarrow P_2 \leftarrow P_5$$

That is, trust is realised between P_i and P_j when:

$$
\begin{aligned}
&P_i \leftarrow P_j : \\
&\quad COT(P_i) \cap COT(P_{i+1}) ... \cap COT(P_j) \neq \{\}
\end{aligned}
$$

COT improves the likelihood of successful negotiations as peers can cache trust chains from previous negotiations, which will reduce the likelihood of future negotiations failing. The cache can also speed up future trust negotiations. However, this additional benefit of *COT* is yet to be explored in our current implementation.

The advantages of having *COT* are quickly overshadowed as the number of overlapping *COT* increases. This is because the more hops you have, the less likely peers will be delegating privileges in open decentralised collaborative systems.

Despite this limitation, COT provides an additional benefit. Overlapping *COT*s can help to abstract virtual organisations through which trust can be discovered and realised dynamically. In virtual organisations, relational hierarchies often exist, which can be modelled over the underlying *COT*.

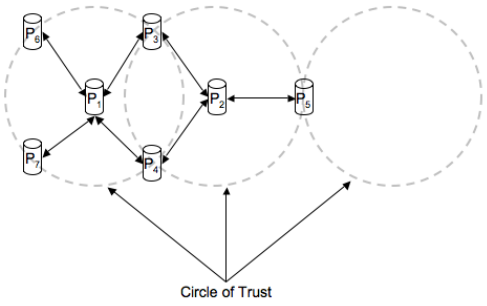


Figure 1. Circle of Trust

3.2 Trust Contract

The presence of multiple domain authorities and policy enforcement introduces a policy semantics divide between domains, i.e. knowing that $org1.investigator = org2.investigator$. Trust contracts *TC* [4] are static agreements between two trusting peers to map credentials between their domains. These agreements cover key management and identity management (authentication) issues. Trust contracts provide one mechanism to overcome the semantic issue of what a credential from one domain means (or should mean) in another domain. However trust contracts require that overlapping *COT*s exist.

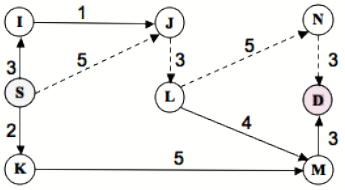


Figure 2. A network of collaborating health organisation

Figure 2 shows an abstract network of a collaborative environment. The network, which is a non-negative, acyclic graph is denoted as G(V,E). The Node set V represents autonomous organisations. A node refers to an end point in a communication chain and consists of security credentials. The Edge set E represents the direction of trust, which consists of policies and constraints. Edges have weights, which represents the cardinality of Trust-Contracts (TC) sets between two Nodes. A tc, $tc \in TC$ is an agreement between two nodes (u, v) that states the mapping/relationship between two credentials (c^u, c^v). TC exists between two nodes when one or more credential mappings are agreed between them, that is:

$$TC = (\{c^{u_0}, c^{v_0}\}, \{c^{u_1}, c^{v_1}\}, ..., \{c^{u_k}, c^{v_k}\})$$

Relationships between credentials are based on *credential equivalence rules*. A tc stems from these rules, which are modelled by function tc:
Let c^u and c^v be the set of credential in domain u and v respectively.

$$[c^u, c^v]$$

$$\frac{tc : c^u \nrightarrow c^v}{\exists\, x : c^u;\ y : c^v \bullet tc(x) = y}$$

Trust contracts provide one solution to credential equivalence problems that exist between autonomous organisations by using equivalence rules. Credential equivalence rules define the relation that exist between credentials. These relations are used in the folding of one credential to another between different organisations [13]. Some of the credential equivalence rules modelled by trust contracts are as follows:

1. Transitive membership rule:

   $R \leftarrow R_1$ and $R_1 \leftarrow R_2 \Rightarrow R \leftarrow R_2$

   This rule means that R_1 is a member of R and R_2 is a member of R_1, then R_2 is a member of R. As an example,

   $org_1.investigator \leftarrow org_2.healthpractitioner$
   $org_2.healthpractitioner \leftarrow org_3.specialist$
   $\Rightarrow org_1.investigator \leftarrow org_3.specialist$

2. Linking delegation rule: $R \leftarrow R_1 \cdot R_2$

   This rule means an entity that has R_2 can act as R if the entity is contained in R_1. Requires at least two dependent roles. As an example,

   $Org_1.CancerTrial \leftarrow Org_1.GP.Investigator$

   $Org_1.GP \leftarrow Org_2.GP$

   $Org_2.GP \leftarrow VOTES$

   $VOTES.Investigator \leftarrow Bob$

   $\Rightarrow Org_1.CancerTrial \leftarrow Bob$

3. Intersection rule: $R \leftarrow R_1 \cap \cdots \cap R_k$ implies an entity that has $R_1, R_2, ...,$ and R_k is delegated R. For example, $Org_1.BrainIT \leftarrow Org_1.Neurologist \cap Org_1.Consultant$, means an entity who is a Consultant Neurologist can participate in BrainIT study.

4 Policies

Four types of security policies are used in DTN. They are local policies, trust-contract policies, access policies and release policies. These are triple-based policies consisting of {subjects, objects, actions} and {obligations} are optional. Each type of policy relates to different stages of a negotiation and for different resources. We consider two types of resources: *credentials* we regard as resources and *database objects and files* we regard as general resources. We express these policies as XACML [3] policy data structures. Each type of policy consists of multiple rules combined by a higher level PolicySet.

4.1 Local Policies

These are policies put in place by service providers to make access decisions for services or resources they provide. The type of resources mostly protected by these policies are general resources like data files or database objects

like tables, data views, stored procedures, etc. These policies are stored in a policy repository and are available to the policy decision point (PDP) for request decisions on data resources. In a decentralised environment, service providers delegate authority to various Attribute Authorities (AA) for privilege management allowing creation and management of security policies which are decentralised and supporting fine-grained access control.

A tabular representation of some local policies is shown in Table 1. This shows various subjects (entities), resources that can be accessed and actions that can be performed. It should be noticed that the subject field contains user names or roles or both. Thus a request may contain a user name and role as subject attributes.

Local policies in XACML are stored as PolicySets with each PolicySet containing more than one policy. The results of each policy are combined using policy-combining algorithms in other to reach a final authorisation decision. Similarly, use of a rule-combing algorithm allows the effects of all rules in a policy to be used for an authorisation decision for that policy. Based on the final authorisation decision reached, where applicable, obligations with a matching "fulfilOn" effect are also included in the PDP response.

4.2 Trust-contract Policies

Trust-contract policies are policies similar to local policies described above. However they are designed to validate and enforce *trust contracts* (or agreements) with respect to existing local policies. Trust-contract policies allow both trusted and third-party trusted entities to be included as subjects in a policy. They are used when a remote entity (trusted or third-party trusted entity) requests a service or resource (data files or database objects like tables, data views, stored procedures). Trust-contract policies provide restrictive access to resources and services in that it limits privileges on roles.

Trust-contract policies include obligations for negotiation requests received from remote entities. A tabular representation of some trust-contract policies is shown in Table 2. Each row represents a trust contract. It should be noticed that the subject (entity) field contains attributes such as third-party-node-identity, remote roles or trusted entity roles, and local roles (based on existing trust contracts). Policies could be written to accommodate "Any" third-party-node-identity and/or trusted entity roles for policy scalability reasons. Policy obligations ensure that authorisation decisions are logged and managed as needed.

Third-party-node-identity enables a PDP to identify and make decisions when a trusted third party is making request for itself or for a third party. This is necessary for scenarios where a service provider restricts a trusted party (trust contract) from negotiating on-behalf of other parties or to restrict third-party access.

4.3 Access Policies

Access policies capture the rules that govern access to local attributes. That is who can invoke and assume local attributes and from where. The subject field of these policies include attributes for third-party-node-identities and roles from trusted parties (roles agreed in a trust contract). However, since third-party-node-identities are unknown, their subject attributes are generic. Local attributes (credentials) are resources to be protected which can only be invoked through satisfying trust contract that have been agreed between trusted parties.

Table 3 shows exemplar tabular representations of information described by access policies. It should be noted from the table that a subject attribute called *D-links* is given and represents the interpretation of linking delegation rule (discussed in section 3.2). This is dynamically concluded based on satisfied trust contracts. For example, in the table shown, when a trusted party satisfies trust contracts for GP and $VOTES$, shown as the first two rows in the table, it is implied that GP is a role in the $VOTES$ trial, that is $VOTES$ is the linking attribute for the GP role. In view of table 2 and 3, it implies that if an entity (user) can perform select, insert, update on PatientMaster and if the same entity can perform select on PatientDrug, then he can perform update on PatientDrug.

4.4 Release Policies

Release policies determine if a negotiated local attributes can be released to a trusted party for purpose of trust negotiation whether for itself or for other parties. It controls what can be released and who it can be released to. Local attributes (credentials) are regarded as sensitive resources, hence are protected by policies. Since release policies are mainly geared towards negotiations with trusted parties, the subject attributes of these policies can only contain trusted-node identities. Table 4 shows a tabular view of some release policies.

5 DTN Architecture

Two systems make up our DTN architecture, discovery system and negotiation system. Figure 3 shows the data flow of the negotiation system. In this section we describe the main components that make up the architecture, which is based upon Security Assertion Markup Language (SAML) [2] as the underlying framework.

Negotiation Service This service is the point of interaction between domains. It provides a secure interface

Subject	Resource	Action	Obligation
Femi, GP	PatientMaster	Select, Insert, Update	
Richard, Investigator	PatientMaster	Select	Anonymise(PatientId, CHI)
Nurse	PatientMaster	Insert	
Clinician	PatientMaster	Select	
Specialist	PatientMaster	Select, Insert	
GP/Investigator	PatientDrug	Select	Anonymise(PatientId, CHI)
VOTES	PatientDrug	Select, Insert	Anonymise(PatientId, CHI)
VOTES/GP	PatientDrug	Update	

Table 1. Tabular View of Local Policies

GP/Investigator is a role label just as a Clinician label. It indicates Investigator is a **subset** of GP.

Subject	Resource	Action	Obligation
Ox, Ox.GP, GP	PatientMaster	Select, Insert, Update	Anonymise(PatientId, CHI), logging
Any, Ox.GP, GP	PatientMaster	Select, Insert	Anonymise(PatientId, CHI), logging
Ox, Ox.(Nurse ∩ Clinician), GP	PatientMaster	Select, Insert	Anonymise(PatientId, CHI), logging
Ox, Ox.Investigator, GP	PatientMaster	Select	Anonymise(PatientId, CHI), logging
Any, Ox.Investigator, GP	PatientMaster	Select	Anonymise(PatientId, CHI), logging
Ox, Ox.VOTES, VOTES	PatientDrug	Select, Insert	Anonymise(PatientId, CHI), logging
Ox, Ox.GP/Investigator, GP/Investigator	PatientDrug	Select	Anonymise(PatientId, CHI), logging
Ox, D-links, VOTES/GP	PatientDrug	Update	Anonymise(PatientId, CHI), logging
Any, Any, GP	PatientMaster	Select	Anonymise(PatientId, CHI), logging

Table 2. Tabular View of Trust-Contract Policies

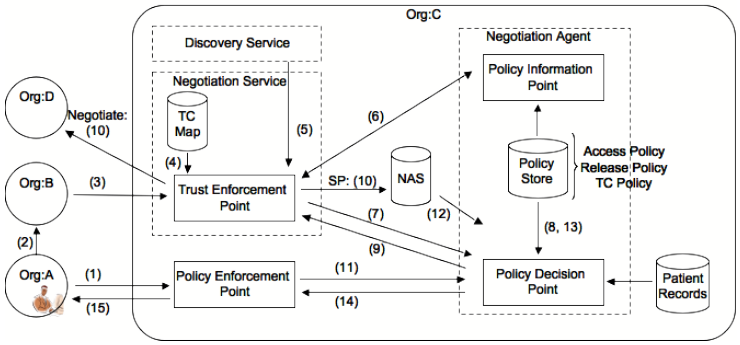
Ox is an abbreviation for Oxford. Ox.GP is combination of two labels that implies Ox issued GP role or means Ox's GP. Ox.(Nurse ∩ Clinician) implies an entity that can function as a Nurse and a Clinician, both issued by Ox. While D-links is an abbreviation for dynamic links discussed in section 4.3.

Subject	Resource	Action	Obligation
Ox, Ox.GP	GP	Invoke contract	logging
Ox, Ox.VOTES	VOTES	Invoke contract	logging
Ox, D-links	VOTES/GP	Invoke contract	logging
Any, Ox.GP	GP	Invoke contract	logging

Table 3. Tabular View of Access Policies

Subject	Resource	Action	Obligation
Nottingham	GP	Negotiate	logging
Imperial	VOTES/GP	Negotiate	logging
Oxford	Nurse	Negotiate	logging

Table 4. Tabular View of Release Policies



1:Demographic data request; 2:Negotiate credentials; 3:Same as (2); 4:Get TC; 5:Get next-hops; 6:XACML; 7:XACML request; 8:Access Policy [& Release Policy]; 9:XACML response; 10a:Negotiated attributes or 10b:Negotiate with attributes; 11:SAML request; 12:Get attributes; 13:TC Policy; 14:SAML response; 15:Data access

Figure 3. Negotiation Data Flow

through which domains can request and exchange attribute assertions. The service encapsulates the negotiation protocol that is used for interpreting messages and carrying out the process of checking *trust contracts*. The service interacts with the negotiation agent and enforces the decision of the agent. Similarly, the service interacts with the discovery service in order to identify other trusted domains (next-hops), when it is acting as an intermediary domain.

Trust Enforcement Point This component interacts with the negotiation agent and enforces the decision of the agent by communicating responses or by interacting with the SAML module. It communicates responses through the negotiation service to a requester or to an intermediary domain. Based on the negotiation agent decision, it initiates a negotiation request with other next-hop domains on behalf of the requester.

Negotiation Agent An agent must understand the protocol used for trust negotiation as in [11] and manage the negotiation session. An agent validates negotiation requests and checks that access and release policies are not violated. The attribute assertions received are validated against access policies and checked against trust contracts that exist between domains. Depending on the negotiation strategy in use, further requests can be made for more attribute assertions from the requesting domain. The agent checks the release policies upon validating the access policies. If any release policies are satisfied, attribute assertions are issued for further negotiations with other intermediary domains or for interaction with the SAML module.

SAML+ The SAML *plus* module are triggered when a domain is the targeted domain. It is called SAML *plus* because it extends SAML by using a Negotiated Attribute Store (NAS). The store is populated with attribute assertions that are issued based on the domain's release policy. These attributes are used to make authorisation decisions based on trust-contract policies for service (data) requested.

6 Performance

We tested our DTN implementation [6] across four *COTs*. The network (trust) topology used was arbitrarily selected and comprised of four COTs. The reason for this constraint was based on our simulated experiment, which indicated an exponential fall in performance as the number of *COTs* involved in trust negotiation increases [5].

We tested the performance of our implementation using several scenarios. Each of the scenarios were tested over several runs comprising the averages of 10 runs on similar network (trust) topologies, each with a total of 8 nodes. Each node was a 2.2Ghz Celeron with 512MB RAM running Linux. All nodes had Grid middleware installed hosting both the discovery and negotiation services. In all our scenarios we identified a node as a source node and another node as a target node. In the first scenario, a source node requested the discovery of a target node by sending route messages to nodes that exists in it's *COT*. This scenario executed on average in approximately 59 seconds. In the second scenario, a source node initiates attribute negotiation with nodes that serve as intermediaries for a target resource. The last negotiation feedback took approximately 15 seconds.

We are carrying out further tests incorporating varying network (trust) topology sizes, number of hops and the size of *trust contracts*.

7 Conclusion

Many cross-boundary security solutions exist today and trust negotiation remains a promising approach. In this paper we presented dynamic trust negotiation, an approach that folds remote security attributes into local security attributes through trust contracts within the e-Health domain. We discussed policies used by the DTN model to limit and control what is negotiated. Obligations were introduced to provide an additional layer of control for data release.

DTN uses intermediary trusted parties to satisfy government directives and to soften the challenges of decentralised security policies. We recognise that further work is needed in the area of trust realisation if we are to reduce to a minimum the security challenges of decentralised collaborations.

Acknowledgements

This work is supported by a grant from the Medical Research Council (MRC) UK and is undertaken as part of the Virtual Organisation for Trials of Epidemiological Studies (VOTES) project.

References

[1] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. http://ec.europa.eu/justice_home/fsj/privacy/law/index_en.htm.

[2] Organization for the Advancement of Structured Information Standards (OASIS). Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, 15 March 2005.

[3] eXtensible Access Control Markup Language (XACML) TC v2.0. Published online at http://www.oasis-open.org/specs/index.php#xacmlv2.0, 2005.

[4] O. Ajayi, R. Sinnott, and A. Stell. Formalising Dynamic Trust Negotiations in Decentralised Collaborative e-Health Systems. In *Proceedings of the 2nd International Conference on Availability, Reliability and Security, (ARES07), Vienna, Austria*. IEEE Computer Society, Apr. 2007.

[5] O. Ajayi, R. Sinnott, and A. Stell. Trust Realisation in Multi-domain Collaborative Environments. *Proceedings of 6th IEEE International Conference on Computer and Information Science, ICIS'07*, July 2007.

[6] O. Ajayi, R. Sinnott, and A. Stell. Dynamic Trust Negotiation for Flexible e-Health Collaborations. *To Appear in Proceedings of 15th Mardi Gras Conference, Baton Rouge, USA*, Feb. 2008.

[7] M. Benantar. *Access Control Systems: Security, Identity Management and Trust Models*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[8] E. Bertino, E. Ferrari, and A. Squicciarini. Trust Negotiations: Concepts, Systems, and Languages. *Computing in Science and Engineering*, 06(4):27–34, 2004.

[9] E. Bertino, E. Ferrari, and A. C. Squicciarini. Trust-X: A Peer-to-Peer Framework for Trust Establishment. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):827–842, 2004.

[10] V. Bharadwaj and J. Baras. Towards Automated Negotiation of Access Control Policies. In *Proceedings of the Fourth International Workshop on Policies for Distributed Systems and Networks (Policy 2003)*. IEEE Computer Society Press, 2003.

[11] A. J. Lee, M. Winslett, J. Basney, and V. Welch. Traust: A Trust Negotiation-based Authorization Service for Open Systems. In *SACMAT '06: Proceedings of the eleventh ACM symposium on Access control models and technologies*, pages 39–48, New York, NY, USA, 2006. ACM.

[12] J. Li, J. Huai, J. Xu, Y. Zhu, and W. Xue. TOWER: Practical Trust Negotiation Framework for Grids. *2nd IEEE International Conference on e-Science and Grid Computing*, Dec. 2006.

[13] L. Ninghui, J. Mitchell, and W. Winsborough. Design of a Role-based Trust-management Framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, 2002.

[14] L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke. A Community Authorization Service for Group Collaboration. In *POLICY '02: Proceedings of the 3rd International Workshop on Policies forDistributed Systems and Networks (POLICY'02)*, page 50, Washington, DC, USA, 2002. IEEE Computer Society.

[15] R. Sinnott, J. Watt, J. Koetsier, D. Chadwick, O. Otenko, and T. Nguyen. Supporting decentralized, security focused dynamic virtual organizations across the grid. In *Proceedings of 2nd IEEE International Conference on e-Science and Grid Computing, Amsterdam, December 2006*, 2006.

[16] W. Winsborough and J. Jacobs. Automated Trust Negotiation Technology with Attribute-based Access Control. In *In Proceedings of DARPA Information Survivability Conference and Exposition, 2003*, volume 02, pages 60–62, 22-24, Apr. 2003.

[17] W. Winsborough and L. Ninghui. Safety in Automated Trust Negotiation. In *Proceedings of IEEE Symposium on Security and Privacy, 2004*, pages 147–160, 2004.

[18] W. H. Winsborough, K. E. Seamons, and V. E. Jones. Automated Trust Negotiation. *DARPA Information Survivability Conference and Exposition (DISCEX)*, 01:0088, 2000.

[19] S. Ye, F. Makedon, and J. Ford. Collaborative Automated Trust Negotiation in Peer-to-Peer Systems. In *P2P '04: Proceedings of the Fourth International Conference on Peer-to-Peer Computing (P2P'04)*, pages 108–115, Washington, DC, USA, 2004. IEEE Computer Society.

[20] T. Yu, M. Winslett, and K. E. Seamons. Interoperable Strategies in Automated Trust Negotiation. In *CCS '01: Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 146–155, New York, NY, USA, 2001. ACM Press.

[21] T. Yu, M. Winslett, and K. E. Seamons. Supporting Structured Credentials and Sensitive Policies through Interoperable Strategies for Automated Trust Negotiation. *ACM Trans. Inf. Syst. Secur.*, 6(1):1–42, 2003.

Author/s:
Ajayi, O; Sinnott, R; Stell, A

Title:
Towards decentralised security policies for e-health collaborations

Date:
2008-11-17

Citation:
Ajayi, O., Sinnott, R. & Stell, A. (2008). Towards decentralised security policies for e-health collaborations. Proceedings - 2nd Int. Conf. Emerging Security Inf., Systems and Technologies, SECURWARE 2008, Includes DEPEND 2008: 1st Int. Workshop on Dependability and Security in Complex and Critical Inf. Sys., pp.165-172. IEEE. https://doi.org/10.1109/SECURWARE.2008.15.

Publication Status:
Published

Persistent Link:
http://hdl.handle.net/11343/28856

File Description:
Towards decentralised security policies for e-health collaborations