Cross-Domain Authentication and Authorisation in Pervasive Computing Environments

University of Auckland, 2014

Anthony Wood

Abstract

This paper is an extension of the landmark paper, Trust-Based Security in Pervasive Computing Environments (Kagal, Finin, & Joshi, 2001). We attempt to improve their solution by including an additional Global Security Agent that has established trust relationships with external authentication servers. This allows for the authentication of untrusted users, and allows for a more precise level of accountability.

1. Introduction

Due to recent developments in wireless technology, pervasive computing environments have evolved significantly, with workplaces and public places providing access to a far greater number of users. In addition, mobile technology has become more powerful, and network administrators have less control over the types of devices that are used to access services through their network. Increasingly, network administrators are expected to provide computing environments that enable ubiquitous, seamless access to services and files, whilst upholding the basic principles of security.

There are four major user need categories for computer security when developing a security policy for a computer system (Lampson, 2004). Firstly, Secrecy refers to the ability of a computer system to protect resources and information from unauthorised access to information (Lampson, 2004). Secondly, Integrity, which refers to the ability of a computer system to control the use and consumption of computer resources (Lampson, 2004). Thirdly, Availability, referring to the responsiveness and accessibility of information in a computing environment, is of significant importance in pervasive computing environments (Lampson, 2004). Finally, Accountability, which refers to the ability to track and manage the use of computing system resources (Lampson, 2004).

Each of these four principles becomes very complex to uphold in a pervasive computing environment because we cannot make assumptions about the processing capacity of devices and services on our networks (Kagal, Finin, & Joshi, 2001). In addition, a subset of our user base may be unknown or untrusted for particular domains. For example, users that are in different divisions/locations may have access to a different set of services on the domain. In addition, it may be possible that the user is completely unknown to the network, for example a contractor joining the company for a temporary period of time (Kagal, Finin, & Joshi, 2001).

The evolution of pervasive computing environments has meant that accountability is of greater importance. This paper proposes a federated security agent infrastructure and explores the benefits and limitations of this model when compared.

This report outlines some of the related work for security in pervasive computing environments and delegation more generally. We then take a high-level look at the solution proposed by Kagal et al. (2001) and then demonstrate our proposed security architecture.

, and provides an overview of the proposed security architecture. The report then explains the benefits and limitations of the proposed model in comparison to the original architecture proposed by Kagal, Finin & Joshi (2001). Benefits and limitations are explained in the context of trust and delegation, revocation, and accountability as a security principle.

2. Related Work

In the landmark paper, Trust-Based Security in Pervasive Computing Environments, a solution for ensuring security in pervasive computing environments was introduced based on distributed trust (Kagal, Finin, & Joshi, 2001). Distributed trust is an essential part of the paper that allows users that are trusted in a domain to delegate their trustworthiness to users that are unknown or untrusted in the security domain. The purpose of the delegation is to meet the flexibility requirements in pervasive computing environments, and ensuring the efficiency and effectiveness of accessibility to services on the domain. The delegation allows previously untrusted users to access services on a domain, without requiring the security administration team to add the user to list of trusted users. While the decentralised security architecture works well to provide the high level of flexibility required in pervasive computing environments, it limits the ability of the network to monitor usage of services across domain, and ensure accountability.

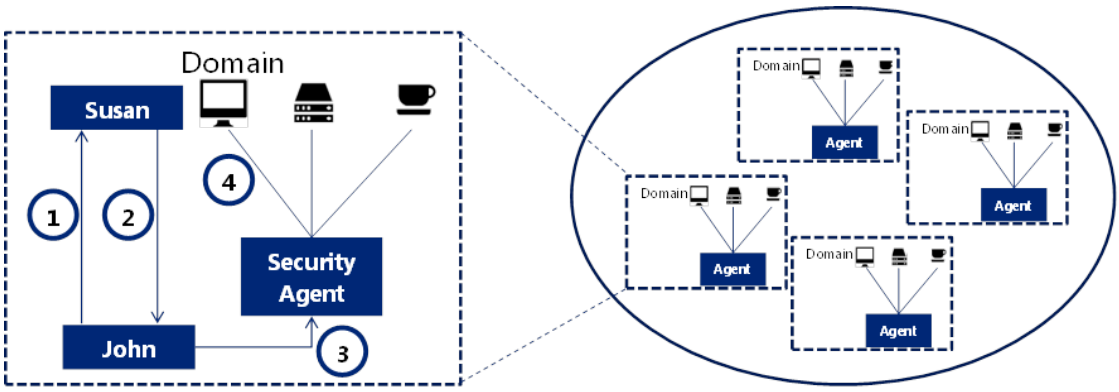
Role-Based Delegation was proposed as an approach to allow users within an organisation to selectively share information without compromising on the likelihood of unauthorised access to that information (Ahn & Mohan, 2004). To do this, the authors proposed a delegation relation, consisting of three elements; the set of original user assignments, a set of delegated user assignments and a set of

constraints e.g. duration of the delegation (Ahn & Mohan, 2004). The concept of role delegation improves on the solution of Kagal, Finin and Joshi (2001) as it allows all rights associated with a role to be delegated, rather than individual rights, making delegation much more user friendly. However, this may lead to the risk of users obtaining access rights which the original delegator did not intend to delegate.

3. Kagal's Solution

In this section of the paper, we take a high-level look at the solution provided by Kagal et al. (2001), and highlight some of the limitations of the architecture. The purpose of this section is to outline what the original solution was, and outline some of the limitations of this model.

This paper implicitly assumes that the services available on the domain do not require a high level of security, as limited damage could be caused by security breaches on the services described, such as coffee machines, projectors etc. However, their solution is able to be extended to include these sorts of services.



1. John sends request for delegation certificate and an ID certificate
2. Susan verifies John's ID certificate, and returns a valid delegation
3. John sends his ID and delegation certificates to the Security Agent
4. The security agent verifies Johns credentials against security policy and grants or denies access to the services

3.1 Benefits

The distributed trust model outlined above allows users that are untrusted in a domain to obtain access to services on that domain through the delegation of rights from trusted users. This achieves the level of flexibility required in pervasive computing environments, as it provides on-demand access to services, without the need for getting the IT Security Administration team involved.

In addition, the decentralised nature of the security agents is important due to the potentially limited processing power of the devices connected on the domain. The decentralised security agents allow for local policy to be enforced on the security agent, ensuring a greater level of flexibility (Kagal, Finin, & Joshi, 2001).

3.2 Limitations

It is possible for an organisation or a network to have a large number of domains, and therefore managing a large number of security agents could become quite difficult.

The decentralised architecture of the security agents means that delegations and revocations cannot be made across domains. This could potentially inhibit both the usability and security of the system. For example, consider the scenario where a role is delegated from a trusted user to an untrusted user in a domain. Only the security agent on that domain is aware of the delegation, and therefore if the same role exists across domains, then the untrusted user will have to request delegations from multiple users, or request multiple delegations from the same user.

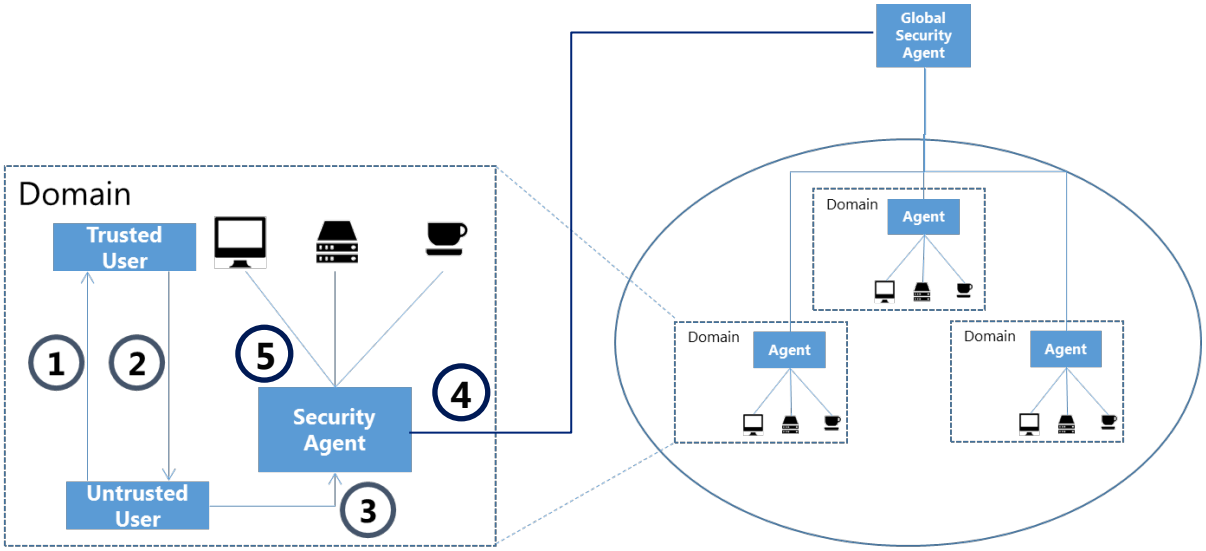
A similar issue exists for revocations. Consider for example a contractor has finished his contract and has access to files and/or services that you no longer want them to have access to and these files are stored across domains. A user will have to revoke the delegations a number of times across multiple times, and security concerns could arise if any of these are missed out. Furthermore, if a user seriously breaches a security policy you may want to automatically revoke all delegations associated with that right. Under the model proposed by Kagal et al. this is difficult to achieve because the security agents operate autnomously.

4. Proposed Solution

This section provides an overview of the proposed security architecture, and outlines how untrusted users are authorised to consume or use services in the domain. We introduce the domain-level security agent, and the global security agent, and their respective roles in the security architecture that we have proposed.

4.1 Assumptions

This paper explores the possibility of a solution that builds on the work done by Kagal, Finin and Joshi (2001) to be more relevant in a society with far greater technology. However, we do not implement the solution or provide, any detailed analysis other than to compare the benefits and limitations of this solution with respect to the architecture. In this paper, we assume that security is of some importance in the network, and that services are not limited to devices that would not typically implement their own security. For example we extend our solution to encompass other services such as File Systems and internal systems.



All untrusted users are able to register on the network using third party authentication providers that are trusted by the organisation. For example, a contractor for an organisation may use his own corporate credentials to authenticate, provided that the external contracting organisation's authentication server is trusted by the organisation's Global Security Agent. In a public setting, a public network may trust web based identity providers such as Open ID or Facebook, and require users to authenticate with those providers and trust.

4.2 The process of authorising an Untrusted User

In our solution, an untrusted user should always be known to the Global Security Agent. However, to access services on a domain, they will need to be given rights from a user with the ability to delegate rights. This happens as follows:

1. An untrusted user may ask a trusted user to delegate a certain role or rights to perform a set of actions. The untrusted user will send his ID certificate, and the right that they would like to obtain.
2. The Trusted User will verify that the Untrusted user is in fact who they say they are by prompting the user to make sure that the ID certificate received in fact belongs to the person sender said they were. Once complete, a delegation certificate is sent from the Trusted user to the untrusted user. The delegation certificate contains information about the rights and/or roles that have been delegated to the Untrusted user.
3. The untrusted user then sends his ID certificate, the request for access to services and the delegation certificate to the Domain-level security agent.
4. The domain level security agent then evaluates all of the security policy rules to ensure that the untrusted user is in fact allowed access to the security agent's managed services. Once this is verified, the domain-level security agent sends the ID certificate and the delegation certificate to the global security agent. The Global Security Agent then checks if the user record exists on the network, and advises the domain-level security agent whether the user should be allowed to

access the services or not based on the security policy. At this point, the global security agent, logs the ID Certificate, Domain-Level Security Agent, Delegation Certificate and the request.

5. Once the domain level security agent is satisfied that the untrusted user should be allowed to access the services on the domain, access to these services are provided.

4.3 Solution

Our solution recognises that the solution put forward by Kagal et al. (2001) has excellent features that recognise the level of flexibility required in pervasive computing environments. Our solution extends their solution, by adding an additional security agent that manages the authentication and authorisation for untrusted users. This ensures that the security architecture addresses the flexibility and security requirements for pervasive computing environments, but also adds additional security to ensure that the architecture can cater for information or services that require a higher level of accountability and security such as file system accesses.

5. Key Differences

The key changes to the solution proposed by Kagal, Finin and Joshi (2001) include the addition of a Global Security Agent. The global security agent is an additional layer of security that ensures that the organisation has a single view of all users that connect to the network and use the services on individual domains. The global security agent acts as an authentication server for unknown users, and registers users on the network based on the credentials that come from the Domain-level security agents. The global security agent then assigns a role based on the delegation that was originally made by the trusted user. This role is not necessarily static and can be coupled with other constraints such as duration.

The domain-level security agents under the Kagal et al. model operate autonomously and are unable to work together to reduce security risks. Our proposed solution introduces another security agent that all domain-level security agents connect to and communicate with. This security agent we call a Global Security Agent, and the purpose of this agent is to ensure that users can be tracked and managed across domains in a pervasive computing environment.

Global Security Agents play a large part in doing the initial authentication of users, through established trusts with external identity providers. This allows users to be created on the domain/network to be created as needed. Initially, a created user has limited rights, for example may be included in the “Guest” role, which, depending on the security policy, could have nothing except read only access to an otherwise public website. However, as the user requires, trusted users can delegate their rights to the untrusted user for a specified period of time. Delegations, revocations, and service consumption is tracked and managed against that user record.

6. Benefits and Limitations

The purpose of this section is to outline some of the core benefits and limitations of the proposed security architecture in comparison to the architecture proposed by Kagal et al. (2001).

6.1 Global Security Policy Implementation

The Global Security Agent can hold some rules, similar in structure to those stored at the Domain-Level Security Agents, to determine whether a user should be able to access services on a domain. The Domain-Level Security Agents send the request, Id Certificate and additional delegations (if any) to the Global Security Agent to ensure that the request and credentials comply with a global security policy that applies to all security agents in the network.

6.2 Trust and Delegation

Distributed Trust is achieved through the ability of trusted users to delegate their rights to other users in the domain. The solution we have proposed has a number of benefits with regards to the Trust/Delegation process especially where a user is known or trusted in one domain, but not another and where rights need to be delegated across domains.

Firstly, consider the case where a user is known to one domain, but not another. The new solution ensures that the user record is known across domains, and also ensures that users only have to register on the network once. The domain level security agent sends the delegation certificate and the ID certificate to the global security agent. The global security agent is then able to manage all delegations against that user record and disseminate this information across all domains. This is especially helpful for performing system audits as it can be done from a central server, rather than having to audit the domain-level security agents individually

Secondly, the proposed solution enables roles to be delegated across domains. This function could be useful if there are a number of smart spaces that are all on different domains, but a user that is able to use one of the smart spaces, should also be able to use the services in another smart space. The global security agent allows roles to be assigned across domains when a domain-level security agent prompts the global security agent to decide whether a specified user should be able to issue a request to a service on its domain

6.3 Revocation

Revocation of access rights is an important concept to uphold the security of network. By introducing the global security agent, we are able to manage revocation of access rights and roles across domains and also to ensure that any breaches of security policy can be actioned appropriately. Another aspect to consider is the case when one user has delegated a right, and another user has revoked an access right.

Similar to trust and delegation as discussed above, the proposed solution allows for the revocation of access rights, or revocation of role assignments across domains where appropriate. This reduces the amount of work a user needs to do in a situation where roles apply across domains, and also reduces the likelihood of overlooking the revocation of the access rights and/or roles in a domain.

Furthermore, it is possible that breaches of security policy are serious enough that we may want to revise their access rights across multiple domains. Under the proposed solution, the global security agent can enforce a number of rules to ensure that serious security breaches result in the appropriate access rights to be revoked across domains. This would be dependent on the security policy of the organisation, but could be useful for situations where smart spaces have similar services, and one of these services have their security policy breached. In this case, the global security agent revokes the rights to use any service similar to the service for which security policy was breached, and this could potentially reach across domains.

The final case for revocation is where one user has revoked a right from a previously "untrusted user", and another user has delegated a right. The Global Security Agent allows business rules to be implemented across all security agents to ensure that these conflicts are managed appropriately. For example, if a revocation of a right is a result of a breach in security policy, all subsequent delegations are invalid, unless a specific action (such as waiving the breach) is performed.

6.4 Accountability

Because all users are required to be authenticated in some form, we can more precisely ensure that users are held to account for any security breaches. For example, in Kagal et al.'s solution, all users in a delegation chain below the user that breached a security policy lose that particular right(the paper actually states all users in the delegation chain lose the right, but we are assuming that this is an error). However, with the additional security agent, we are able to effectively ensure that there are

This also allows us to more accurately check how resources in a network have been accessed and used, and allows us to hold a user to account for accessing or modifying information. For example, if a user accesses information in a network, this can be recorded in the Domain-Level Security Agent and then used in Network-Wide audits of the system.

6.5 Privacy

The major limitation of our proposed solution is privacy. Because we are exposing our identity to the network, it is fundamental that the user knows that privacy will be upheld within the network. In certain situations this may not be the case, and therefore this is the major weakness in the solution.

7. Conclusion

Our solution, in theory addresses some of the major drawbacks of the solution proposed by Kagal et al. However, one thing that Kagal's solution does very well is ensure the privacy of the users that are accessing the services on the domain is maintained. In situations where it is feasible that privacy is less of a concern than ensuring secure access to network resources, our proposed solution provides additional controls over the way in which Trust and Delegations are issued, how Revocations of rights and/or roles are managed and how accountability is managed for untrusted users.

8. Further Work

The purpose of this section is to outline a number of further pieces of work that could be undertaken in relation to the security of pervasive, or ubiquitous computing environments. There is more work to be done to ensure the architecture is effective and also some further work required, tailoring the architecture to meet the privacy requirements of Mobile Ad-Hoc Networks.

Firstly, because this is just a hypothesised architecture, and hasn't actually been implemented, additional work would be required to determine whether it was appropriate. As part of this further work, detailed analysis should be undertaken to ensure that there are no security risks and to clearly articulate the trust relationships between identity providers and the global security agents

Technology is advancing extremely quickly, and pervasive computing environments are evolving into Mobile Ad-Hoc Networks. Mobile Ad-hoc networks have a number of challenges for the security design, including an open, peer-to-peer network architecture, shared wireless infrastructure, variable resource constraints and a dynamic network topology (Yang, Haiyun, Ye, Lu, & Zhang, 2004). In such situations, users of a network may not know who is providing the network connectivity and therefore it may be risky to share their ID certificate and/or public key with other devices on the network.

One potentially major limitation of this solution in Mobile Ad-Hoc Networks is that users are known to the network, and therefore a robust approach to ensuring that the privacy of users is maintained, whilst ensuring the security of information contained on the network.

Bibliography

Ahn, G.-J., & Mohan, B. (2004). Secure Information Sharing Using Role Based Delegation. *Proceedings of the International Conference on Information Technology: Coding and Computing Vol.2* (pp. 810-815). IEEE.

Kagal, L., Finin, T., & Joshi, A. (2001, December). Trust-Based Security in Pervasive Computing Environments. *Computer , vol. 34 , no. 12*, pp. 154-157.

Lampson, B. W. (2004, June 21). Computer Security in the Real World. *Computer, Vol. 37, No. 6*, pp. 37-46.

Yang, H., Haiyun, L., Ye, F., Lu, S., & Zhang, L. (2004, February). Security in Mobile Ad-Hoc Networks: Challenges and Solutions. *IEEE Wireless Communications*, pp. 38-47.