Trust Architecture for a Personal Distributed Environment

Swee Keow Goo, James M. Irvine, Robert C. Atkinson and John Dunlop
Mobile Communications Group, University of Strathclyde
Royal College Building, 204 George Square, Glasgow G1 1XW, UK
Email: {sweegoo,j.irvine,r.atkinson,j.dunlop}@eee.strath.ac.uk
Tel: +44-(0)141-5482061, Fax: +44-(0)141-5524968

***Abstract*— With an increasing number of wireless devices and access technologies available, users will be able to access their Personal Distributed Environment of services and data conveniently in a wide variety of ways. Unfortunately, this flexibility comes at a cost - higher security risks and vulnerabilities. The traditional association with a network provider may not exist, replaced by a far more nebulous association with a number of unknown entities, network nodes and service providers. These ad hoc relationships require a notion of trust, which presents great difficulties in a dynamic wireless environment. This paper presents a formal trust architecture to address these issues, with the focus on aspects of trust policy formation and its evolution.**

I. Introduction

The view of the next generation has evolved from the single multi-mode 'super-terminal', to one where users access services through a wide variety of different terminals optimised for their application. These terminals, services and data that the user will access, form the user's "Personal Distributed Environment" (PDE) [1]. The PDE is a dynamic entity, changing not only with the services, but also with the location and access technology. Locally within a PDE, the different terminals that a user has available, such as cell phone, laptop, media player, etc., are likely to communicate by means of one or a combination of long and/or short range wireless technologies. Although the PDE concept generates business opportunities for both the service providers and the network operators, it has also instigated trust issues between these parties and the PDE users. With distributed access to data, perhaps using shared terminals such as displays in an Internet café, the risks of unauthorised access to data or spoofing of the user are greatly increased.

Over the years, several trust management systems have been introduced. Some are developed to solve the trust issues with specific focus on general authorisation [2], [3] while others have concentrated on authentication [4], [5] and particular applications [6], [7]. However, comparison between these approaches is difficult due to the breadth of these system specifications and the trust languages employed. The lack of precision inevitably introduces further doubts of their suitability to specify and express the security needs both effectively and intelligently to a dynamically changing environment, with devices entering and leaving the PDE.

The ability to specify trust in a commonly understood format across domains is essential, as without this, users will not be able to trust that services offered from the 3rd parties are safe. The frequent need to physically split and merge several different PDE sub-networks will also make the trust problem more complex, as each different sub-network will have its own security mechanisms (based, for example, on the access network), and its own identity server process.

This paper considers the trust requirements and issues and how several policies can be generated and managed via a suitable specification language for a PDE scenario. The remainder of this paper is organised as follows. In Section II, a formal definition of trust notion is provided. In Section III, a representation of trust relationship between entities is introduced. Section IV details how the trust policies can be derived using an example. Finally, Section V concludes the paper.

II. Definition of Trust

In this article, we define PDE trust as:

The belief or willingness to believe an entity based on its competence (e.g. goodness, strength, ability) and behaviour within a specific context at a given time.

The competences refer to the performance capabilities such as interpreting the user requirements correctly or executing the policy rules properly while behaviours refer to the possibilities of colluding and lying.

III. Representation of Trust Relationships

Fig.1 presents an overall fundamental structure for explicit expressions of trust relations between the entities and how the various trust policies can be created in the PDE context, with the intention of sustaining trust for:

- Entities that wish to join the PDE,
- Entities that want to establish a PDE-internal or/and PDE-external relationship(s) with other entities, and
- Entities that want to be assured of a device's performance and the performance of the PDE's execution system.

From Fig.1, three essential domains are identified:

- *PDE Domain*: a zone that consists of devices and entities owned and trusted by the PDE user.
- *Service Domain*: a zone whereby only trusted computing environment, users, devices, applications, agents, data sources are permitted to access when sufficient security procedures/ mechanisms are performed.
- *Other domain*: an untrusted zone as perceived by a PDE user. It consists the PDE networks of other users, 3rd

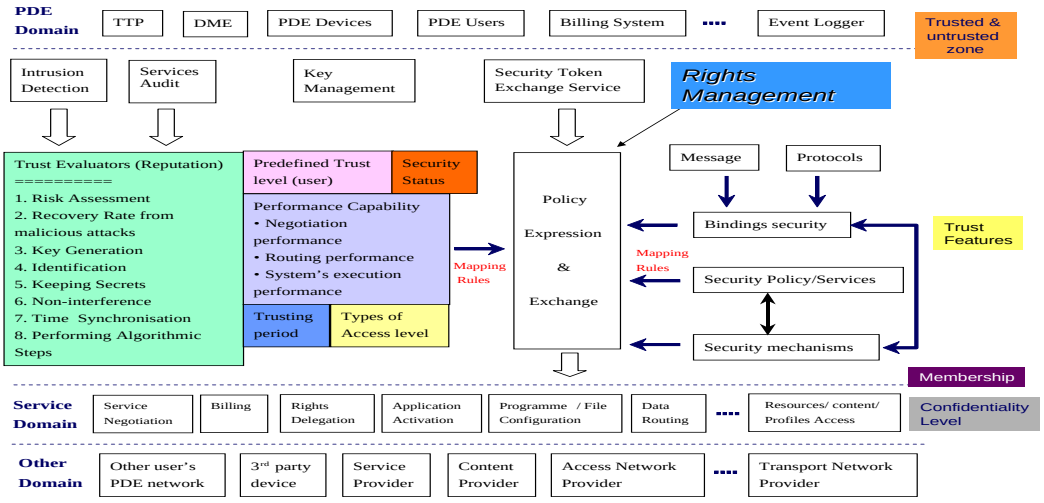

Fig. 1. Trust architecture for PDE

party devices, service providers, content providers, access providers and transport providers.

The advantages of having these separate domains are that:

- It allows issues to be addressed in each zone independently.
- The security functions can be executed more aptly and less ambiguously as only the specified entity formats in each domain can have the access rights to enquire for a security procedure. The recommended access formats are agent, execution environment, application, content (e.g.video), data and message.
- It supports secure group communication and enables interoperability that is not restricted by local security policies.

The crucial element in the framework is the *"trust engine"* which falls in between the *PDE* and the *Service* domains. Different relevant types of trust requirements can be identified and classified in this region. The trust engine depicts a formalism for expressing requirements for trust relations and a contemplation for identification of several security constraints in developing a trust policy. The trust information provided in the trust engine is time dependent and, in general, it is also varying rapidly in order to give a reliable state of information/condition.

Six key criteria, that are considered for expressing the trust needs as input to the procedures for handling requests in the policy expression and exchange are:

- The *Trust Reputation* [8], [9] anticipates that trust establishment can no longer rely solely in the outcome of the executed security mechanisms/techniques, which include cryptographic algorithms and access control methods. This is because these mechanisms/techniques do not say much about the broader notion of an entity's trustworthiness. For example, a signed certificate does not advise you if the owner is a co-conspirator or a spy. Depending on which particular trustworthiness or creditability that an entity wants to establish about its corresponding entity, eight different trust evaluators such as *Risk Assessment*, *Recovery Rate from Malicious Attacks*, *Key Generation*, *Identification*, *Keeping Secrets*, *Non-interference*, *Time Synchronisation and Performing Algorithmic Steps* are proposed. If direct social cues are not available, the reputation (or trust value) can still be based on indirect observations or evidence from the auditing service and the intrusion detection system. The other reason for having these trust evaluators is to guard against any indirect risk contact that could be inherited, especially from a presumably honest and authenticated entity. For instance, a PDE user's mobile phone may have downloaded content that is infected with virus and passed it on to a laptop without knowing it, or their mobile phone may have been accidentally corrupted by the messages when used by a Bluetooth-enabled PDA seeking its aid to send messages to a remote controller via infra-red. Hence, the trust evaluator like *Risk Assessment* which serves to assess the amount of risk that an entity will incur to its corresponding entities, is useful. However, these eight trust evaluators will not totally remove all the anticipated risks as their roles are to assist the users to decide on an option with the least level of perceived risk.
- The *Pre-assigned Trust Level from user*, whereby flexibility is required in both the trust allocation and mapping.
- The *Performance Capability* refers to the ability to deliver the promised services or tasks. For example, what a PDE user requires from his PDE system is whether his PDE system is faithfully executing the instructions that are given to it.
- The *Trusting Period* which is difficult to define quanti-

tatively, but can have defined *Pre-Trust, Mid-Trust and Post-Trust* periods. The latter requires an entity to also satisfy the security rules set by both the *Pre* and *Mid-Trust* periods before it can be granted the rights to access and release confidential resources and information such as user's profile, location details and monetary information.
- The current *Security Status* shows the security level that an entity has obtained. This component places an effect on the amount of access rights (i.e. trusting rights) which will be granted to an entity.
- The *Types of Access* which ties with the types of required membership.

Though the proposed criteria above will influence the trust deployment in a relationship in every respect, it has nevertheless facilitated a consistent view of trust to be deployed in the PDE context.

As for setting up a dynamic trust policy, the required criteria are:

- To be effective in exchanging information on which trust decisions may be based, agreed *Protocols and Message* are also necessary.
- The *Security Policy/Services* refers to the existing policies/services such as privacy policy and authorisation function that can be re-used and integrated as part of the new trust policy.
- The *Security Mechanisms* such as digital signature and encryption assure security functions (e.g. integrity & confidentiality) have the capability of enforcing various service qualities between the end-users. The amount of security mechanisms to be employed in a service also rely on how important this sevice is and its implication.
- The *Bindings Security* is to tie the security characteristics from the *Security Mechanisms* to the agreed *Protocols and Message*.
- The *Security Token and Exchange Service* provides a set of rules to the trust engine to create and exchange an entity's characteristics such as name, group and capability.
- *Policy expression & exchange* is where an ideal policy language is identified and is used to express the capabilities or any strong constraints of the PDE security. It also facilitates service requestors and providers to exchange dynamically security (among other) policy information in order to establish a negotiated security context between them.

Unlike the OSGI's Web Services (WS)-Trust Specifications [10], the PDE's trust architecture anticipates that a trusting relationship should not be established by just using trusted proxies. Evaluation of trust and interrelationships between the outcome of the security execution and the access rights are also vital to building a trusting relationship. If not, issues may arise when a relationship is built with no clear understanding on the referring or requiring trust component. In addition, it appears that WS-Trust Specifications may apply only to the WS security standard for securing web services at the message layer.

IV. A PDE Scenario

A PDE user, Bob, wants to make use of his own PDA to update a program file in his home computer network while he is travelling to work. Through his PDA's search, two public members have responded. To invite one of the public members, Bob has to consider a few issues with the assumption that both the service cost and the transmission speed are not critical factors.

A. Consideration Issues

To Bob, his prime trust concerns when selecting a foreign device comprise of:

1) How can I trust the foreign device or the offered service?:

- Does this foreign device have sufficient of security protection and features?
- Will it deliver the promised service at the end of the day?
- Did this foreign device hack any of my devices before?
- Did this foreign device pass any computing virus to my device?

The listed concerns are relevant if Bob has any past experiences (either general or specific ones) with the foreign device and if there is any operation records of the foreign device from trusted sources. If this foreign device is new to Bob's PDE, the concerns addressed in Part 2) will be more appropriate and applicable.

2) How do I specify how much trust I should place in this external device?:

- Are my devices (i.e. my PDA and computer network) more vulnerable than others?
- Can all my devices speak for me?
- Who are the owners of this external device and its service?
- If the foreign device has no record, how many security mechanisms and functions are required for my complete trust?

The aspects discussed above somehow rely on how much confidence Bob has on his PDA and his home network and whether the trust evaluator and the performance analyser can furnish an honourable assertion. On the other hand, if the public device holds a credential that is accredited by one of the PDE's Device Management Entities (DMEs) or Trusted Third Parties (TTPs), Bob may save some "worrying" times. Once Bob has decided whether he should completely or partially trust this foreign device, he can move on to determine the trust period and the amount of access rights that the foreign device can have with a bit of help below.

3) How much trust period and access rights are required in this transaction?:

- How important is this program file?
- Did the foreign device satisfy the minimum security access procedures?

At this stage, Bob is not excessively concerned about how the corresponding foreign parties can trust Bob's devices as the trust negotiation will be established when the two parties

exchange their security policy information. When Bob has fairly determined his trust criteria for this external device/ service, the trust engine can proceed to generate a trust policy.

B. Policy Implementation

1) Policy Specification Language: We select a structured and widely adopted language, eXtensible Markup Language (XML), as our policy representation and implementation [11]. The main advantage of XML is that it is increasingly used to integrate applications and communicate between systems in many environments [12], [13].

2) The Derived Trust relationships: Presumably, Bob has derived two different trust relationships to be executed:

- Situation 1 (simple) - The external device is new and has no operation record. Bob's PDA and his home computer network have no record of any illegal intrusion or virus infection. The program file requires minimum amount of security services as it's already highly encrypted with a security technique. Bob specified his trust requirements as:
  - Foreign device needs to be authenticated first and then satisfy the security access rules to *Mid-Trust Period*.
  - Only then, membership with "forward program file-name, "Program123" to home network Servername "KingKong" will be endorsed.
- Situation 2 (slightly complex) - Bob's PDE network changes because his authoriser (PDA) moves to his new office's PDE. Bob wishes to include the current policy with additional requirement such as:
  - Extends the current membership to access his office network if no bad reputation is detected.

3) Psuedo-code in Steps: Situation 1 can be expressed in the following algorithm steps.

```
Set external device = ED
if ED's access format = selections {agent, message}
{
   if ED's authentication & authorisation status = satisfy pre-trust security procedures
   { Membership id = 0011
      Expiry date = yyyy.mm.dd
      Issuer = Bob's LocalDME01
      Given access area = Bob's PDE subnetwork1
      Given rights = Talk/send message to all Bob's devices
      ...
      if Ed's authorisation = satisfy mid-trust security procedures
      { Extend expiry date = yyyy/mm/dd
         Extend rights = Forward program
         Given external file = Bob's updated Program123
         End username = KingKong
         Initiator = Bob's PDA
         else
            No access rights
            Remain at pre-trust membership
      } else
      Tag ED's status = "Illegal entity"
   }else
   Inform ED's recognisable access format = selections {agent, message} }
```

As for Situation 2, it can inherit situation 1's policy and extend to:

```
   if Bob's LocalDME01 detects PDA has moved away from Bob's PDE subnetwork1
to subnetwork2
{ Auto-search for policy_PDA
   if Membership id 0011 is not expired
   { Auto-authorisation request to access Bob's PDE subnetwork2
      if authorisation = satisfy cross-network security procedures
      {
         if reputation results from trust evaluators = PDA pre-assigned trust value
         { Extend appropriate rights & policy expiry dates = yyyy/mm/dd
            Issuer = Bob LocalDME02
         else
            Remain at pre-trust membership until it's expired
         } else
         Remain at pre-trust membership
      } else
      Check PDA if ED is still required
      if yes, check if ED is contactable & available
      if no, send disconnection message
      ...
   } else
   Tag ED's status = "expired entity"}
```

However, before these expressions are converted into proper source codes, policy statements are required to capture information such as the elements of the trust reputation and the permission assignment to the local DMEs, PDA and PDE network. These include how these involved parties are related in terms of their roles and hierarchy arrangement. The examples are shown below:

```
<!– Policy statement for trust requirements–>
<?xml version="1.0" encoding="UTF-8"?>
<Policy_ID> Policy_PDA </Policy_ID>
<rule><Membership> 0011 </Membership>
<Actions> <Action> <TrustAction_Match>
<Security_Status>
<SignatureValue>cRCKrtwPS6vd...VNCcY5rHaFPYW
</SignatureValue></Security_Status>
<PREDEFINED_TRUST> User_Value = "moderate" </PREDEFINED_TRUST>
<REPUTATION_STATUS>TrustTitle="NA"</REPUTATION_STATUS>
<PERFORMANCE_STATUS>PerformanceTitle="NA"</PERFORMANCE_STATUS>
<RequiredAccess_Level>Mid-Trust Period </RequiredAccess_Level>
...
<Other_credential></Other_credential>
</TrustActionMatch>< Security_procedures>
<RequiredPermission> PID="P5" REMOVE OPERATION =
"CREATE"</RequiredPermission>
</Security_procedures >
</Action></Actions>
</rule>
```

```
<!– An example of what a trust reputation can be based on. The trust level is
reconfigurable but trust synchronisation is required if no common trust definitions are
used across different domains/parties–>
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT TRUST_REPUTATION (TrustValue)>
<!ELEMENT TrustValue (RiskAss,RecoveryRate,KeyGen,Identification,Secrets,
NonInterf,ClockSyn,AlgoriStep)>
<!ELEMENT RiskAss (#PCDATA)>
...
<!ELEMENT AlgoriStep (#PCDATA)>
<!ATTLIST RiskAss (-1|0|1|2) #REQUIRED>
...
<!ATTLIST AlgoriStep (-1|0|1|2) #REQUIRED>
<REPUTATION_STATUS>
<REPUTATION TrustTitle = "Most trusted"><TRUST_REPUTATION> TrustValue
= "16" </TRUST_REPUTATION>
<REPUTATION TrustTitle = "Moderate trusted"><TRUST_REPUTATION>
TrustValue = "greaterthanequal 8"</TRUST_REPUTATION>
<REPUTATION TrustTitle = "Least trusted"><TRUST_REPUTATION> TrustValue
= "lessthan 8" & "greaterthan 0" </TRUST_REPUTATION>
<REPUTATION TrustTitle = "No trust"><TRUST_REPUTATION> TrustValue =
"lessthan 1" </TRUST_REPUTATION>
</REPUTATION_STATUS> ...
```

```
<!– A DTD file is created to capture the architectural elements of a PDE network –>
<?xml version="1.0" encoding="UTF-8"?>
<pde>
<assign_date = "2004/04/04"></assign_date><time>14:05</time>
<!ELEMENT PDE_hierarchy (RootDME,LocalDME,Device)>
<!ELEMENT RootDME (ISP_givenID,Root_VID,Root_other)>
<!ELEMENT ISP_givenID (#PCDATA)>
```

```
<!ELEMENT Root_VID (#PCDATA)>
<!ELEMENT Root_other (#PCDATA)>
<!ELEMENT LocalDME (user_DMEgivenID,Local_VID,Local_other)>
<!ELEMENT user_DMEgivenID (#PCDATA)>
<!ELEMENT Local_VID (#PCDATA)>
<!ELEMENT Local_other (#PCDATA)>
<!ELEMENT Device (user_DevicegivenID,Device_VID,actualfunctioning_name,
Device_other)>
<!ELEMENT user_DevicegivenID (#PCDATA)>
<!ELEMENT Device_VID (#PCDATA)>
<!ELEMENT actualfunctioning_name (#PCDATA)>
<!ELEMENT Device_other (#PCDATA)>
</pde>
```

```
<!– An example of a xml document which captures the hierarchical structure of Bob's
PDE. –>
<?xml version="1.0" encoding="UTF-8"?>
<assign_date = "2004/04/04"></assign_date><time>14:05</time>
<!DOCTYPE PDE SYSTEM "pde.dtd">
<pde hierarchy_arrangement>
   <parent_PDE_ID>
      <ISP_givenID> 123.45.12.10 </ISP_givenID>
      <Root_VID> user10 <Root_VID>
      <child_PDE_ID>
         <user_DMEgivenID> LocalDME01 </user_DMEgivenID>
         <Local_VID> p22 </Local_VID>
      </child_PDE_ID>
      <child_PDE_ID>
         <user_DMEgivenID> LocalDME02 </user_DMEgivenID>
         <Local_VID> p10 </Local_VID>
      </child_PDE_ID>
   </parent_PDE_ID>
</pde hierarchy_arrangement>
```

```
<!– Below shows the basic permission assignment for different role definitions. Only
the Root DME has all the access rights to all the resources while his Local DMEs could
only access the resources of its respective subnetworks –>
<?xml version="1.0" encoding="UTF-8"?>
<assign_date = "2004/04/04"></assign_date><time>14:05</time>
<ACCESS_MODEL TYPE_NAME="ROLEACCESS_POLICY">
<!–Role set definition–>
<ROLE TITLE="RootDME"></ROLE>
<ROLE TITLE="LocalDME"></ROLE>
<ROLE TITLE="Device"></ROLE>
...
<!–Role hierarchy and rights inheritance definition–>
<INHERITS FROM="Device" TO="LocalDME" ></INHERITS>
<INHERITS FROM="LocalDME" TO="RootDME"></INHERITS>
...
<!–Permission set definition–>
<PERMISSION PID="P1"OPERATION="READ,WRITE,CREATE,UPDATE,DELETE,
NAVIGATE,EXECUTE,DELEGATE" RESOURCE ="DATA01"></PERMISSION>
<PERMISSION PID="P2"OPERATION="READ,WRITE,CREATE,UPDATE,DELETE,
NAVIGATE,EXECUTE,DELEGATE" RESOURCE ="DATA02"></PERMISSION>
<PERMISSION PID="P3"OPERATION="READ" RESOURCE ="DATA02">
</PERMISSION>
<PERMISSION PID="P4"OPERATION="READ,CREATE,EXECUTE" RESOURCE
="DATA03"></PERMISSION>
<PERMISSION PID="P5"OPERATION="DELEGATE,CREATE" RESOURCE =
"DATA03"></PERMISSION>
<PERMISSION PID="P6"OPERATION="READ,WRITE,CREATE,EXECUTE"
RESOURCE ="DATA04"></PERMISSION>
...
<!–Resource definition–>
<RESOURCE DATA_ID="DATA01" ITEMS= "DATA02" DOMAIN= ALL
LOCALDMES></RESOURCE>
<RESOURCE DATA_ID="DATA02" ITEMS= "User_profile,Location_details,
Monetary_info,DATA03" DOMAIN= ONE LOCALDME></RESOURCE>
<RESOURCE DATA_ID="DATA03" ITEMS= "Service_profile,data_file,DATA04"
DOMAIN= ONE SUBNETWORK></RESOURCE>
<RESOURCE DATA_ID="DATA04" ITEMS= "Contact_profile,Local_Appfiles"
DOMAIN= ONE SUBNETWORK></RESOURCE>
...
<!–Basic permission assignment–>
<PERMISSION_ASSIGNMENT ROLE="RootDME" PERMISSIONS= "P1">
</PERMISSION_ASSIGNMENT>
<PERMISSION_ASSIGNMENT ROLE="LocalDME" PERMISSIONS= "P2">
</PERMISSION_ASSIGNMENT>
...
</ACCESS_MODEL>
```

Due to space restrictions, the attribute assertions and the identity expressions of the user and devices are not shown.

V. Conclusions

We have presented a new approach to managing security policies in a distributed, dynamically changing and ad-hoc wireless environment. By adopting a widely-used structured language such as XML, properties such as interoperability and manageability can be achieved across various specified domains. Generally, we believe that we could further develop a general security language for expressing any trust policy for a distributed and ad-hoc environment with investigation on the security specifications approaches from SAML [14] and XACML [15] and security issues of XML [16].

Acknowledgment

The work reported in this paper has formed part of the PDE area of the Core 3 Research Programme of the Virtual Centre of Excellence in Mobile & Personal Communications, Mobile VCE, www.mobilevce.com, whose funding support, including that of EPSRC, is gratefully acknowledged. Full detailed technical reports on this research are available to Industrial Members of Mobile VCE.

References

[1] Dunlop, J., Atkinson, R. C., Irvine, J., and Pearce D., "A Personal Distributed Environment for Future Mobile Systems", *IST Mobile & Wireless Communications Summit*, June 2003.

[2] Blaze, M., Feigenbaum, J., and Lacy, J.,"Decentralised Trust Management", *Proc. IEEE Symposium on Security and Privacy*, Los Alamitos, 1996, pp164-173.

[3] Blaze, M., Feigenbaum, J., Ioannidis, J., and Keromytis, A., "RFC 2704 - The KeyNote Trust-Management System Version 2", http://www.faqs.org/rfcs/rfc2704.html, 1999.

[4] Stubblebine, S. G., "Recent-Secure Authentication: Enforcing Revocation in Distributed Systems", *Proc. IEEE Symposium on Research in Security and Privacy*, 1995.

[5] Wobber, E., Abadi, M., Burrows, M., and Lampson, B., "Authentication in the Taos Operating System", *Proc. ACM Symposium on Operating System Principles*, 1994.

[6] Balfanz, D., Dean, D., and Spreitzer, M., "A security infrastructure for distributed Java applications", *Proc. IEEE Symposium on Security and Privacy*, 2000, pp15-26.

[7] Chu, Y.H., Feigenbaum, J., LaMacchia, B., Resnick, P., and Strauss, M., "REFEREE: Trust management for web applications", *Computer Networks and ISDN Systems*, 29(8-13), 1997, pp953-64.

[8] Beth, T., Borcherding, M., and Klein, B., "Valuation of Trust in Open Networks", *Proc. 3rd ESORICS*, 1994.

[9] Goo, S.K., Irvine, J.M., and Atkinson, R.C., "Personal Distributed Environment - Securing the Dynamic Service Platforms Beyond 3G", *Proc 3G2003*, London, UK, June 2003, pp18-22.

[10] Della-Libera, et al, "Specification: Web Services Trust Language (WS-Trust)", http://www.ibm.com/developerworks/library/ws-trust/index.html, 18 Dec 2002.

[11] Bray, T., et al, "Extensible Markup Language (XML) 1.1", W3C, http://www.w3.org/TR/2004/REC-xml11-20040204.html, 04 Feb 2004.

[12] Kudo, M. and Hada, S., " XML Document Security Based on Provisional Authorisation", *Proc. 7th ACM Conf. Computer and Communication Security*, 2000, pp.87-96.

[13] Damiani, E., et al, "A Fine-Grained Access Control System for XML Documents", *ACM Trans. on Information and System Security*, Vol.5(2),2002, pp.169-202.

[14] DeJesus, E., "SAML Brings Security to XML", http://www.fawcette.com/xmlmag/2002_02/magazine/columns/collaboration/edejesus.html.

[15] OASIS 2001. XACML language proposal, version 0.8.

[16] Blyth, A., et al, "Security analysis of XML usage and XML parsing", *Computers and Security*, Elsevier Science, 2003, pp.494-505