Trust Context Spaces: An Infrastructure for Pervasive Security in Context-Aware Environments

Philip Robinson and Michael Beigl
Telecooperation Office, Institut für Telematik, Üniversität Karlsruhe
{Philip, Michael}@teco.edu

Abstract. The issue we have focused on in the broad area of security for Pervasive Computing is maintaining trust in an interactive environment. Our solution is based on the premise that computers and implicit interaction mechanisms must function in accordance with the explicit parameters of physical human-human interaction. Otherwise, this results in imbalances between the physical and virtual worlds, which leads to "windows of vulnerability". Our solution presented requires an infrastructure of pervasive and context sensing technology, to provide entity mapping, policy and trust management services. We also investigate generating cryptographic keys using the context available The underlying technology is based on the Smart-Its context sensing, computation and communications platform.

1. Introduction

A primary driving force behind ubiquitous and pervasive computing research is the focus on bridging the divide between real-world-oriented tasks and the interfacing of facilitating technology [23, 32]. Want et al state in their 1999 paper [31], "There has long been a discontinuity between the rich interactions with objects in our physical world and impoverished interactions with electronic material." The narrowing of this discontinuity is primarily attributed to developments in Wearable and Smart-Appliance computing technologies [10]. They transform our computing experience into a more personal yet more personalize-able one. They allow a measure of freedom from location with respect to availability of services, yet the rendering of these services may be dependent on the location. Our identities are represented or impersonated by the devices we carry [9], such that we, our physical documents, and other physical artefacts often possess what is known as a "virtual counterpart" [20].

We believe that many security issues in Pervasive Computing stem from the difficulty of coordinating the symbiosis of physical and virtual entities or counterparts. In section 2, we elucidate this premise of "Virtual-Physical Imbalances and Security Vulnerability" by way of scenarios. Section 3 evaluates other existing work relevant to this theory, while section 4 presents our proposed solution called "Trust Context Spaces", including the goals, components and technology. We conclude with our expectations, intended contributions and scope for future work.

2. Virtual-Physical Imbalances: Opening Security Vulnerability Windows

Security starts with an analysis of risks. It is then the means by which we seek to limit access to items or knowledge that have some value to us. The extremities of these limitations range from oneself to a group of trusted peers. The benefits of electronic media including data organization, querying and exchange, facilitate more efficient collaboration and generation of business decisions and notifications. In pervasive computing, this may however include interaction with devices and services without the same owners, and no prior knowledge of the character or background of each other's impersonated identity [15]. Brown et al in [5] define six categories for pervasive context-aware applications, namely, proactive triggering of events, streamlining interaction, memory for past events, reminders for future events, optimising patterns of behaviour, and sharing experience. These sorts of applications enhanced with implicitly communicated context information may cause unintended leakage of information, even if there was an explicit physical effort to avoid this. The two scenarios below are characterized by more than one of the pervasive categories given above.

2.1. Imbalances within Personal Areas

When I choose to pack my personal items away in my brief case, this is an indication that no one is to see them. If this physical action is not equally and thoroughly supported by the virtual world, then this is a potential point of vulnerability that may be exploited by an outsider. We refer to such situation as "implicit sharing of resources". That is, while we use implicit interactions to gain some benefit of services, it often comes with the sacrifice of implicitly forfeiting my right to control the access to the piece of information distributed [27]. However, the result of completely forgoing implicit interaction leads to an overload of management tasks and frequent interrupts. Consider if I have to be worried about maintaining the secure environment of the physical documents in my desks, the physical keys that I carry around in my pocket, coupled with the numerous passwords and PINs that I carry around in my head [2]. This is clearly an administrative overload from the perspective of security, and something that we need to consider in our research. In any event, it is still in a person's best interest and within their rights to be confident in both the physical and virtual provisions of security.

Security is not only about protecting oneself (privacy), but also about groups collectively protecting their resources and knowledge produced during meetings and other form of interaction.

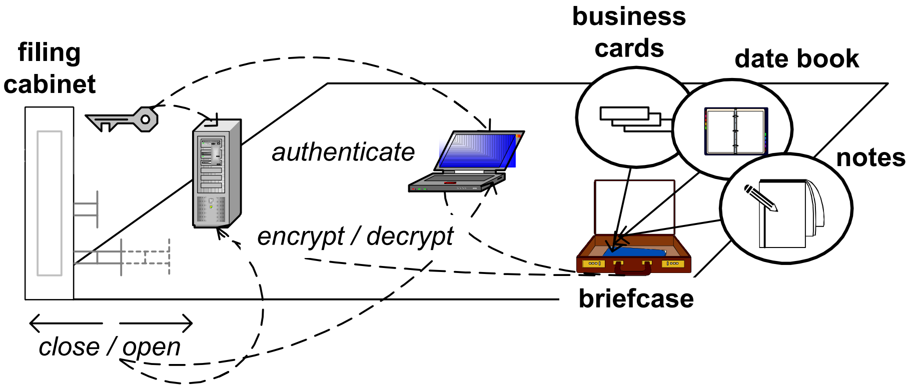

Fig. 1. Example of how physical artefacts may be mapped to virtual/ electronic applications and data, depicting a metaphor for explicit indication of need-for-privacy or personal security.

2.2. Imbalances within Shared Environments

We considered a further situation where information is explicitly shared within an assumed controlled and trusted setting and environment. However, the interaction that leads to distribution of electronic meeting articles may be implicit from the perspective of the pervasive computer systems at work [7, 9, 26]. Such is the case when a presentation is shared with a defined group of attendees, through graphics, text and voice. These media elements may also be captured in electronic form for storage, reuse, or recording of what occurred in the presentation. The participants also have digital identities as their personal devices may impersonate them, or there was some issuance of group IDs or personal tokens during a registration phase, prior to the meeting.

During this meeting, it may so happen that subgroups wish to exchange information within a separate context not included in the main stream of discussion. From a device and application perspective, creation of a new session and hence security context would be necessary. This would entail generation and distribution of new session keys for the subgroup. Group management (Join| Leave | Parse | Merge) is therefore an undertaking for both the physical and virtual systems at work, as participants possibly come and go [16]. Furthermore, consider unofficial people entering the room at untimely intervals, causing disruption to the proceedings and becoming privy to information perhaps not intended for their knowledge. Very likely scenarios for imbalance between virtual and physical worlds that may occur include an electronic presentation continuing to be displayed, even though the physical presentation has been put on pause, devices continue to interact in spite of the cessation of interactive activity among their real-world peers, or electronic files have been locked away (devices turned off or files logically encrypted), yet their physical information equivalents remain in the open.

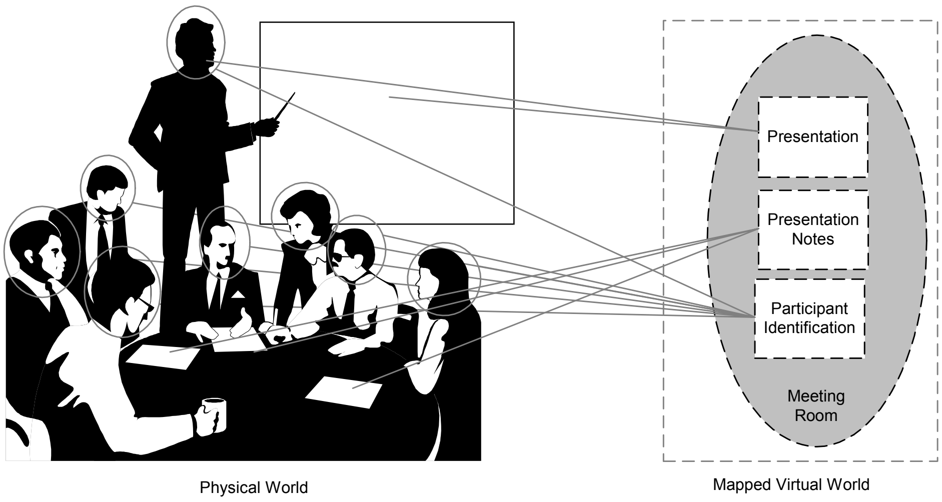

Fig. 2. People and artefacts in group meetings may also have physical/ virtual mappings

Sharing of interactive context and therefore the need to establish a Trust Context Space can be as a result of being in the *same place* and the *same time*, in *different places* but at the *same time*, in the *same place* but at *different times*, and finally, in *different places* and at *different times*. Nevertheless, as Brown et al state, "… a user should be able to retrieve [context] information about who was present at a meeting while they were present, but not to find out who came and went after they had left" [5].

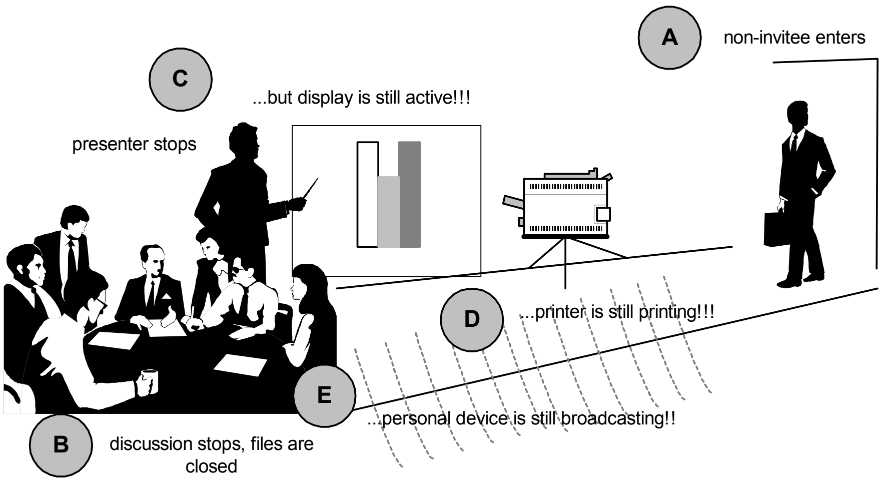
A
non-invitee enters
C
...but display is still active!!!
presenter stops
D
...printer is still printing!!!
E
...personal device is still broadcasting!!
B
discussion stops, files are closed

Fig. 3. Meeting scenario showing imbalance between virtual and physical worlds

2.3. Security Vulnerabilities

Arbaugh et al suggest two state models in [3], representing system vulnerability lifetime and system state with respect to vulnerability. Vulnerability is said to go through the phases of birth to death, with the possible intermediate states surrounding the discovery and correction of the causative flaws. Accordingly, a system is said to be "hardened" when its integrity, expected usage and behaviour remain consistent, "vulnerable" when potentially weakening flaws are discovered, and "compromised" when the discovered flaw is exploited at least once. The goal of systems design and management from the perspectives of safety and security are to first curtail the existence of vulnerabilities, minimise the time the system spends in the state of vulnerable, or to recover in a timely manner if the system has reached a state of compromised.

What is special about security vulnerabilities in pervasive computing? While this is a very application-dependent question, there are some broad characteristics that do have immediate implications for security. Firstly, pervasive computing facilitates a greater overlap of virtual application and physical context, resulting in a larger interactive context [1, 4, 26]. The interactive context includes communications, processing, storage and I/O, which all influence the way that security is realized. Secondly, and consequently, the scope for implicitly shared information has become wider as context-awareness has become a more prominent feature of pervasive systems. Figure 4 depicts these anomalies.

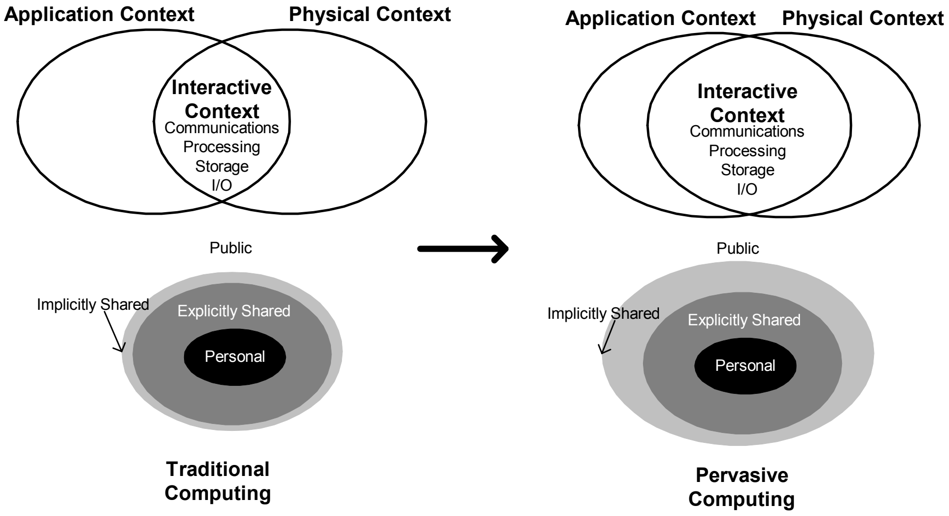

Fig. 4. Change in security considerations, moving from traditional to pervasive computing. These two concepts go together to define a Trust Context Space –

2.4. Pervasive Security Vulnerability Windows

Based on this look into the state-of-affairs, we describe security vulnerability windows in pervasive computing as the instances where information is implicitly shared with other peers through an inconsistency in the interactive context. This may be resultant from the sequencing of events in the physical world, including actions with explicit security implications, not being interpreted by the virtual world correctly or timely. Consider two people that have just discovered each other and are now at the stage of making decisions pertaining to trust; however, their personal devices are already interacting and exchanging pieces of information. This may be applied to the scenarios described earlier, where the people in the physical environment have made a decision not to share presentation information with an outsider, yet this was not the total reflected response of the virtual elements. Figure 5 gives a graphical representation of this.

In order for at least two peers to form an interactive relationship, or form a communications channel, they typically go through the phases of peer discovery, which is followed by some form of authentication, evaluation of trust, and channel assembly based on the policy for the communication. The final stage is of course a termination of communications. The issues in pervasive computing for spontaneous authentication, authorisation and trust are being addressed in the ad hoc networking and nomadic computing communities [33, 18, 11]. However, what we want to deal with a bit more is the effects of implicitly ordered invocation of these actions by the virtual counterpart, without proper guidance and sanctioning by the physical counterpart. This leads to weak forms of authentication and weak bases for determination of trust, and consequent weak constraining of any communications channels.

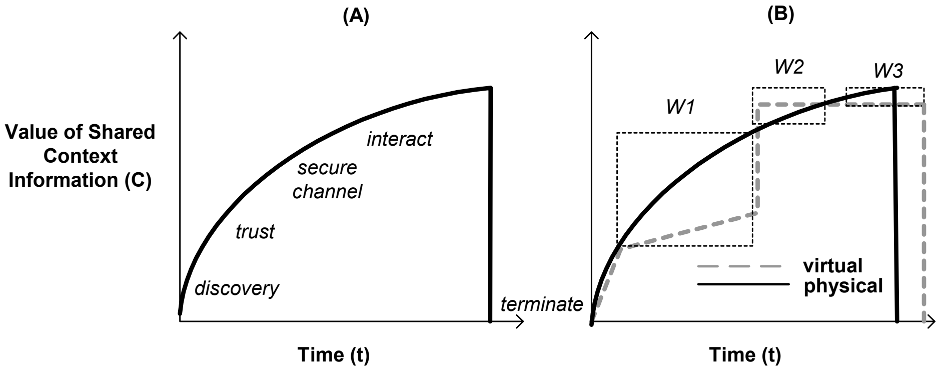

Fig. 5. Example State charts for Collaborative Context Transitions. Plot (B) is a superimposed plot of virtual over physical entity, highlighting the instances where particular transitional imbalances are cited, and depicted as Security Imbalance Windows (W1, W2, W3).

Having deliberated over this issue, we have two research agendas within the area of pervasive security; firstly, to *minimize the occurrence of security vulnerability windows*, and secondly, to *constrain the ill effects of implicit sharing of information*. Section 3 goes on to present our foundation themes that we considered.

3. Principles and Related Work

Within our lab, we have had practical experience with developing and using applications with multiple sensors for validation of situations and association of entities and events [4, 10]. Our first experiment with using simple sensor reading to associate items was the Smart-Its friends prototype [10], which allows a simple everyday activity, such as shaking, to create a logical association between items shook in the same instance, pattern and speed. This led us into investigations of combining more context variables (or sensor readings) into deriving association patterns, and coined this work the "generation of a *Context Key*". The use of this concept is outlined further in section 4, along with the required technology and experimental observations. Nevertheless, there is a wealth of research that we have found related to our concepts, and have evaluated their contributions to a solution for the issues we have highlighted.

3.1. Security through Physical Artefacts

Smart Cards represent a well-secured and limited interface device for transporting a person's virtual identification [8]. Therefore, when a person uses a Smart Card to open a door, it may trigger simultaneous virtual events associated with that person physically and explicitly announcing their authenticated presence. Furthermore, their well-known use in GSM mobile equipment to contain a user's credentials allows a network operator to react if the person's virtual identity has been stolen or involved in irresponsible use of resources. The field of Biometrics can also be considered as an even more intimate use of physical artefacts (our own bodies) in creating a link to our virtual identities. Fingerprints, retina scans and even more obscure things like footsteps [24] are used or are being further researched as ways of uniquely identifying a personality. The pervasiveness of fingerprint or retina scanners is a bit questionable, but having more loosely coupled sensors is more relevant for our work.

3.2. Secure Group Management (Secure Multicast)

The scenario described in section 2.2 indicated to us that some of the resources for tackling virtual-physical imbalances were to be found in the secure multicast community. There are two major techniques used to manage and exchange security context information amongst groups. One technique uses public key concepts, such as Diffie-Hellmann, to organize peers in a tree structure with the root being the overall group key [16]. However, peers remain autonomous with respect to generation of their private keys. Other techniques are based on symmetric cryptography, where n peers share a secret, such that at most two peers can reconstruct the entire group secret [29]. In any event, there are some criteria for good secure multicast key management techniques, which we endeavour to incorporate in our overall goals [16]. In summary they suggest that old members should not be capable of deriving the new group secrets (the context key in our case) and new members should not be able to derive those used before.

3.3. Virtual Identity Management

Perhaps if we have adequate means of coordinating our virtual identities then this is the solution to virtual-physical imbalances. The researchers at Freiburg University present a prototype that shows how a user-interface can change based on which identity the user assumes as a result of his context, primarily location [13]. Other research seeks to manage virtual identities through *pseudonyms* and *anonymity* [25]. That is, by providing a virtual counterpart identity that is not linkable to my legal and physical identity, one can be assured of a measure of privacy. An even more active concept of a virtual identity is the use of *mobile agents*, which can be described as nomadic executable content. They go from site to site and locally carry out transactions on behalf of their originating physical identity. The issues for security here are the risks of the mobile agent maliciously abusing the resources of visited hosts, and vice versa, the host altering or wrongfully depriving the agent's functionality. The streams of rebuttal research are either *passive preventative* – an agent is restricted to work in a trusted domain, *active preventative* – mathematical functions for obfuscating the functionality of the agent, while the host platform is either a tamper resistant, sandboxed operating system, or *detective* - agent activity is audited and mutual malicious behaviour is reprimanded [28, 30]. We have comparable goals for identity management, trust and security through interrogation of the interactive context.

3.4. Privacy and Policy Management

Privacy has been one of the earlier security themes addressed within the area of pervasive and ubiquitous computing. We equate the concerns for privacy with what we have described as implicit sharing. Langheinrich states in [19], "…we must realize that our real-world presence cannot be completely hidden, or perfectly anonymized." His principles for maintaining privacy centre on notice, choice and consent, proximity and locality, and access and recourse. Having a feedback mechanism with regards to use of resources is also incorporated in our research agenda. Other recent work in privacy-enabling systems by the group from Berkley led them to deriving the theory of "Approximate Information Flows"[14]. From this they seek to detect when there are imbalances in the flow of information between communicating parties, or hoarding of private information, and aim for minimum information asymmetry as a matter of communications policy. Our work is complementary, in that we consider information asymmetry with respect to physical and virtual counterparts.

3.5. Context Information in Security

Looking into other context-based security research was a matter of fact. Noble and Corner describe "Transient Authentication" as a means of authenticating users with devices through a small, short-ranged wireless communications token that they wear. This therefore demonstrates the use of location (proximity) as authentication based on context [22]. Following authentication, Covington defines Environment Roles as a methodology for capturing context in authorization and access control models [6]. Kindberg et al go a step further and suggests that the context information itself should be authenticated [17]. Their work on context constrained channels suggests that communications channels are "send" or "received" constrained, based on some context predicate. As will be seen in our architecture, authentication and authorisation are not our primary uses of context information, but they are emergent properties of using a context key.

4. The Trust Context Space Architecture

A Trust Context Space represents the degree and state of trust within a certain interactive context (figure 4). It is therefore a mapping of physical and virtual application contexts, based on the nature of communications, processing, storage and I/O within the area of interaction. The operation of the architecture is therefore dependent on our quantification of an interaction context and criteria for determining trust. Our target platform for pervasive and ubiquitous computing involves wireless short-, medium- and long-ranged communications (therefore power and protocol-constrained broadcast), possibility for spontaneous networking protocols (bluetooth, IrDA, WaveLAN), Networked Sensors (location, temperature, light, speed, etc), personal and shared devices and workspaces. When we say "environment", we typically refer to offices or other rooms that people share and interact using security-relevant data.

At this point we give a bit more details about the core technology used to facilitate context-awareness in the architecture, namely, the Smart-Its [10]. These are products of an ongoing project that supplies a hardware and software platform for integrating computing, sensing and communications capabilities into everyday objects.

Everyday objects include chairs, tables, doors, shelves and more. The Smart-Its contain their own operating system, processor (PIC 18F452 at 20 MHz, 5 MIPS), memory (32 program, 1.5 kB RAM, 8 kB FRAM), RF communications (868,35 ISM band, bandwidth 125 kbits/s) and a variety of integrated sensors, and they are programmable through an API. The available onboard sensors include audio (high-linear microphone and amplifier from 50 Hz – 20 kHz), light intensity (TSL 250 light sensor at 880 nm wavelength), temperature (Dallas DS1621 >>99% accurate between 0 – 40^{o}C), and humidity (Hygrote SHS A3, 99% precision). As we have already built many different scenarios based on this technology, we leverage the platform for constructing the architecture specified here.

4.1. Technical Goals of the Architecture

The first goal of the architecture was to determine a well-reasoned means of evaluating trust. One analogy we used was the inherent willingness of people to trust acquaintances or even strangers, based on the fact that they have access to the same physical property. That is, based on the established guarantee of security by a well-enforced building, people tend to be more at ease with other peers they choose to interact with on the premises. This can be compared to establishing an intermediary trust reference, such as a CA (certificate authority) in PKIs (public key infrastructure). Therefore, our first architectural goal is to allow the trust of an entity to be derived from the overall sense of trust of the physical and virtual environment. Once the environmental sensors and systems detect or are alerted of a security imbalance, the overall trust changes and entities will then be notified of how to collectively redeem the overall environmental trust. Our practical implementation heavily depends on the precision of the sensors, the precision of the Analog/Digital interfaces and the computing power of the processors.

Nevertheless, in order for infringement of trust to be detected there must be some concrete rules or policies in place for stating conditions that must be maintained, and the evidence that must be produced to verify these conditions. Policies are defined by an administrator for access to the resources of the environment, while each entity is also responsible for managing their local policies. We therefore need a scalable methodology for specifying and negotiating policies, as the smart items and user devices are not as powerful as the workstations hosting the environment's services.

With reference to figures 1, 2 and 3, we also require a way of transferring real-world security primitives into the computational domain. That is, a means for the virtual counterparts to recognize explicit cues from the physical world, intended to counter disclosure of information. This is a clear requirement for incorporating the use of context awareness in a trust infrastructure. However, there are three things that we found must be considered when dealing with context information in security: firstly, security mechanisms are generally reliant on values that are not easily guessed and part of a wide field space, such as would be provided by finding the modulus of a large prime integer by a random number. Some elements of context have very little variability and hence may be easily guessed through basic knowledge and heuristics, which is not good if thinking to use them as the basis of a secret key. For example, temperatures in a building typically range between 19 and 25^{o}C and light intensity is normally between 50 and 60 Hz. Secondly, sensor values can be quite variable even in closely co-located spaces e.g. the temperature near a window where the sun is shining through is warmer than next to the AC unit. Therefore forming implicit symmetric keys is a challenge using sensor values with these properties, without being restricted to very particular circumstances of very close collocation over an extended period of time. Thirdly, some sensor values may remain consistent over a long period of time, e.g. humidity, such that if one is thinking about a protocol with frequently refreshed key values that possess weak backwards dependability, this is a constraint.

Finally, with so many policies, regions of trust, smart items and sensors, the management of these environments where implicit sharing is allowed leans towards an administrative overload. The goal of using the context key is to define security mechanisms that are more integrated in everyday activity and not severe distractions from productive tasks, unless critically required to do so.

4.2. Architecture Components

The goals of the architecture may be summarized as ensuring that trust services are deployed appropriately and in a timely fashion. The components we derived are therefore our opinion of the basic infrastructure for managing context-based security and trust services. Our use of the term “appropriate” suggests that we do not seek to provide “security overkill”, such that it becomes more of a task or hindrance than a service and support. With reference to figure 8, phases (A), (B) and (C) represent situations where differing expectations of security exists. Differing expectations therefore implies differing policies and mechanisms. The architecture diagram is presented below (figure 7); the role of each component becomes clear in the process description that follows.

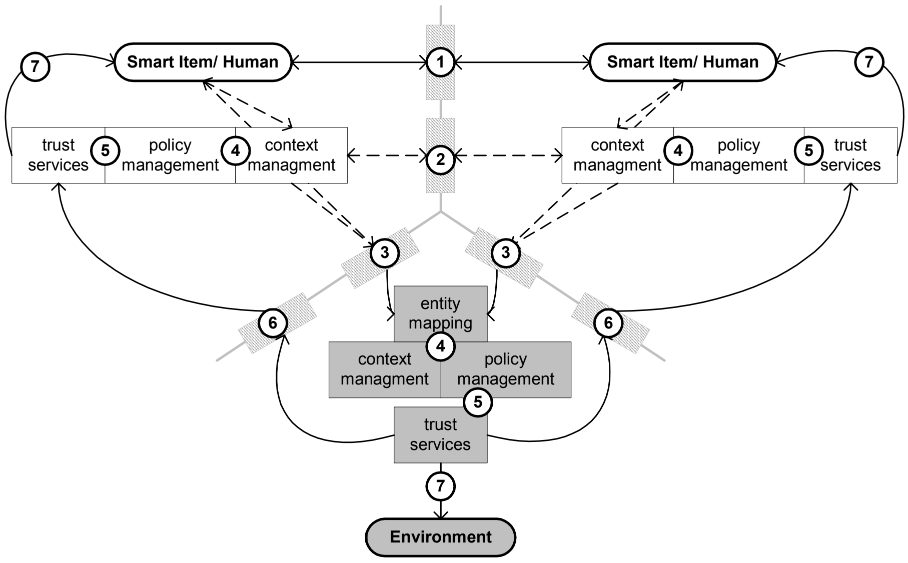

Fig. 6. Smart Environment where Trust Context Space Architecture manages security

(1) Communication

Explicit human - human (everyday interaction), human - smart item (I/O), and human initiated smart item – smart item (electronic communications) interactions are executed through established and defined interfaces. Objects in our environment use various communications channels. For example, the Smart-Its use a proprietary RF protocol called SPOT, which allows low power and low range communication, while PCs, laptops, and PDAs are either wired to the Ethernet, or connected via WaveLan. We therefore had to implement a Bridge between the Smart-Its communications protocols and the Ethernet/IP network to which the environment services are connected.

(2) Context Management

The **context management** components, serving each entity, coordinate sensor samples, collate the sensor values and semantically determine the interactive context. Therefore, the explicit interactions in (1) also generate implicit sensor information, which is used to derive the interactive context between the peers. This may also include implicit interaction of the context management component peers, as they share the same conditions. The context management components are the core application of the Smart-Its, which may be connected to the user devices as a peripheral over the serial interface. It is therefore during this implicit exchange that the context key between peers is locally, simultaneously and symmetrically calculated. The time of key generation is implicitly determined based on when the peers initiate physical interaction. However, at our stage of testing we still agree on the time that entities start to generate shared context, as the samples of the sensor signals used must be identical. Consider our experimentation and results using sound below:

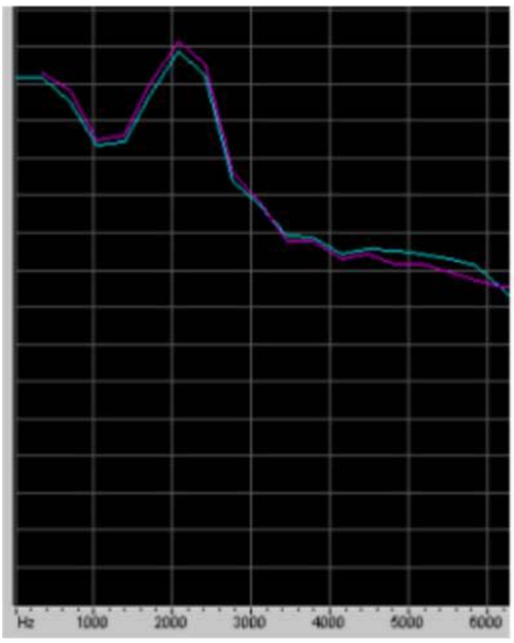

Fig. 7. A curve of the typical room noise in the Frequency domain recorded from two different sources inside the room.

The advantage of audio is its fast changing characteristic over time and the equal distribution over a closed area, e.g. a room. Furthermore audio signals do not spread beyond closed rooms and are therefore suitable for the scenarios aimed in this paper. Co-located audio signals are especially comparable from their frequency spectrum. In our approach we use an audio signal recorded over 3 seconds. Due to this long sampling time the shift of the signal resulting from the distance of the two microphones plays no role. Recording starts in all participating devices at exactly the same time (less than 10µs shift, guaranteed by the network protocol). Recorded signals are then transferred into frequency domain and normalised. The resulting distribution is divided into 11 500 Hz spectrums starting from 500 Hz. Each of the spectrums is rated a value between 0-7 and therefore contributes with 3 bits to the key (see figure 7). The overall length of the key is 33 bit. Although this key length seems to be inadequate for high security settings and also the 3 bit key-parts are not completely independent, we suggest that the key freshness is still an advantage. In our system we propose a change at least every 60 seconds. A change of the key also eliminates the seldom error that two co-located nodes do not compute the same key.

(3) Entity Mapping

The sensors of the environment share the interactive conditions with all the interacting entities. Therefore, an active or aware environment may also be considered as an entity in itself, yet it has a constant interactive relationship (implicit and explicit) with its occupants. In a controlled environment, it is therefore possible for the application services of the environment to be aware of the location and other context factors of each entity. This means that it can generate the context key of any occupants and must therefore be provably trusted – recall the attacks on agents discussed in section 3.3. The **entity mapping** service creates a thread of execution per entity, which results in the instantiation of their virtual counterpart from the perspective of the environment. Our platform for the entity mapping service is the *RAUM* system [12]. The *RAUM* orders related location information in a tree-structure, with the root containing the semantics and Cartesian coordinates of the overall area of interest. Entities in the environment may then be located by following a path through the tree.

(4) Policy Management

Policies are specified and enforced locally (autonomously) per entity as well as environmentally (hierarchical constraints). Sloman broadly categorizes policies as either *obligation* (what a subject must do) or *authorization* (what a target allows) [28]. The **policy management** component is registered with its corresponding context management component, to be informed of key states of interactive context. Currently, we conceive these being related to location (Known? Trusted?), communication (Range? Authenticated? Encrypted?), interacting peers (Known? Trusted?), group enrolment (Explicit? Trusted?), and time (Scheduling Conflicts?). We also include a policy called an *introspective policy*, which is a form of obligation policy that states conditions and controls for an entity's internal reconfiguration based on the context. Access controls or authorization policies then state conditions and controls for how interacting entities can use the resources of another.

(5, 6, 7) Trust Service Management – Conditions and Evidence

The recipient of policy decisions is the **trust services** manager, which controls the invocation of services for handling authentication, authorisation and confidentiality. It is this component's task to present the appropriate interface and request for evidence that policy-specified conditions are met. Evidence may include passwords, certificates, public keys, and in the circumstances where it is warranted, the context key. The environmental services also go through this process of introspective policy enforcement followed by definition of access controls. In its role as overall "interactive context monitor", the environment's trust services also provide feedback to the entities regarding the trust state of the overall interactive context. Upon receiving obligation policies from respective trust services, each entity makes autonomous decisions or negotiates on how it will provide the evidence required to continue interactions in a manner that is acceptable by the overall environment.

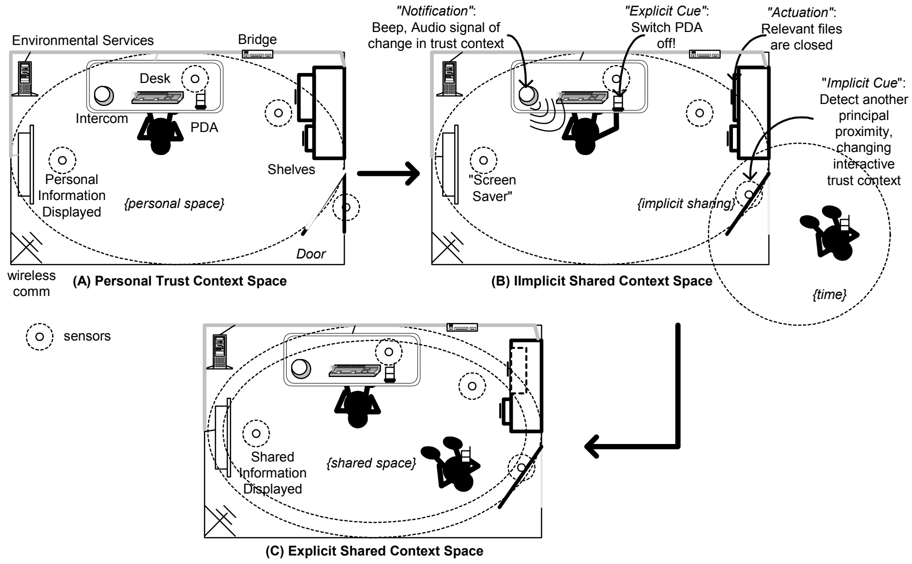
"Notification":
Beep, Audio signal of change in trust context
"Explicit Cue":
Switch PDA off!
"Actuation":
Relevant files are closed
Environmental Services
Bridge
Desk
Intercom
PDA
Shelves
Personal Information Displayed
{personal space}
Door
wireless comm
(A) Personal Trust Context Space
"Screen Saver"
{implicit sharing}
"Implicit Cue":
Detect another principal proximity, changing interactive trust context
{time}
(B) IImplicit Shared Context Space
sensors
Shared Information Displayed
{shared space}
(C) Explicit Shared Context Space

Fig. 8. Visualization of transitioning trust states, and reorganization facilitated by the Trust Context Spaces architecture

An environment is most trusted by an occupant after a period of adjustment and familiarization, or if they have made explicit efforts to reinforce the areas of apparent insecurity. With reference to figure 8 (A), the room's principal user has obtained his trusted environment by closing the door. This explicit cue triggers the generation of a context key between the communicating devices (task of context management component), and sets the trust state to "personal space" (registered by policy management and trust services). This is the formation of a context group within the room environment. Each entity in the group has a locally determined responsibility to maintaining the integrity of the group. For example, in (B), the door is the logical component to monitor proximity of other people, while the public display parses data based on the parameters of disclosure stated by the access control policy. Therefore, when the door entity notices another person entity arrive, it signals to the context group that the trust state has changed to implicit shared, and each entity goes through reconfiguration – the intercom alerts the user and the public display temporarily hides its display with a screen saver. Another interesting procedure that we will also seek to implement is the use of actuation (physically changing the context by affecting the environment). In (B) the shelf is also part of the context group as the files stored are physical counterparts of the data being currently displayed on the public screen and PDA. The user's explicit movement to turn off the PDA is then a confirmation that everything is to be locked until given further notice. Such is the case in (C) when the user establishes a relationship with the new peer to the environment, and, through the use of policies and issuing of conditions to the peers by the trust services, the trust state is resolved to shared, and new context keys are generated. The ability to reorganize, resolve and accommodate are therefore the summarized administrative goals of the Trust Context Space.

6. Conclusion

In this paper we have introduced another viewpoint of the issues in security for pervasive computing, where we have shown examples of the loss in synchronization of interactive context between virtual and physical counterparts resulting in security vulnerabilities. As a direction towards solving these issues, we have also presented our experience with the use of context information in generating symmetric keys between peers, bypassing the need to implement a complicated distribution mechanism. To manage the generation of these keys and the formation of groups, the Trust Context Space architecture was derived.

We currently use our technology as a test-bed for the use of context in security, starting in a very controlled or active environment. We see this as an incubator for virtual identities, and thus a more practical environment to study their complexities. In terms of scalability, we envision complex Trust Context Space architectures being constructed by hierarchically grouping active environments. This stems from our designation of an active environment as an entity in itself. However, we are also thinking about how the use of context in security is suited to an uncontrolled environment, such as in infrastructure-less public spaces with no sensing architecture or presence of an online trust reference.

Furthermore, there are still some weaknesses of this approach and other functionalities that we do not wish to overlook. Firstly, there is the question of sensor integrity and detection of tampering. We refer to these situations as “perceptive attacks”, where an adversary maliciously alters the sensor readings to report false context data. Secondly, we have only managed to generate a 33-bit key with sound, within an environment of possibly small entropy field space. By today’s standards this is not a challenge to hack, but we have also stated that the situation in which we use these keys makes a difference. Researchers at Bell Labs and Carnegie Mellon are also working along a related research path, as presented in [21], where they state that the use of simple pass-phrases by users does not provide sufficient entropy to form the basis for a cryptographic key (subject to dictionary attack). Therefore, their approach has been to analyse the variance in how the pass-phrase is spoken as opposed to the content of the pass-phrase alone. We also believe that functionally combining more than one aspect of the context can lead to better-sized keys within larger field spaces.

References

[1] Abowd, Dey, Brown, Davies,Smith and Steggles, “Towards a better understanding of context and context-awareness”, (panel statements), Handheld and Ubiquitous Computing, (H.-W. Gellersen, Ed.), Springer, Berlin, pp. 304-307, 1999.

[2] Anderson, "Security Engineering: A Guide to Building Dependable Distributed Systems", pp 38, Wiley 2001

[3] Arbaugh, Fithen, McHugh, "Windows of Vulnerability: A Case Study Analysis", IEEE Computer, December 2000, pp 52-59

[4] Beigl, Gellersen, Schmidt, “MediaCups: Experience with Design and Use of Computer Augmented Everyday Artefacts”, Computer Networks, Special Issue on Pervasive Computing, Vol. 35, No. 4, March 2001, Elsevier, p. 401-409

[5] Brown, Burleson, Lamming, Rahlff, Romano, Scholtz, Snowdon. "Context-awareness: some compelling applications", December 2001, Retrieved from http://www.dcs.ex.ac.uk/~pjbrown/papers/acm.html Feb 2003

[6] Covington, Long, Srinivasan, Dey, Ahamad, Abowd, "Securing Context-Aware Applications Using Environment Roles", SACMAT 2001, Copyright 2001 ACM

[7] Dourish, P. and Bellotti, V., Awareness and Coordination in Shared Workspaces, Proceedings of CSCW'92, 107-114.

[8] Hendry, "Smart Card Security and Applications". Artech House, 1997

[9] Holmquist, Falk, Wigstroem, "Supporting group collaboration with interpersonal awareness devices", Personal Technologies", Vol. 3, pp. 13 - 21, 1999

[10] Holmquist, Mattern, Schiele, Alahuhta, Beigl, Gellersen, "Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts", Proc. of UBICOMP 2001, Springer 2001

[11] HuBaux, Buttyan, Capkun. "The quest for security in mobile ad hoc networks". In Proc. ACM MOBICOM, Oct. 2001.

[12] Hupfeld, Beigl, "Spatially aware local communication in the RAUM system", Proceedings of the IDMS, Enschede, Niederlande, October 17-20, 2000, pp 285-296

[13] Jendricke, Kreutzer, Zugenmaier, "Pervasive Privacy with Identity Management", Workshop on Security in Ubiquitous Compiting , UBICOMP2002, September 2002

[14] Jiang, Hong, Landay, "Socially-Based Modeling of Privacy in Ubiquitous Computing", UbiComp 2002, Springer LNCS 2498, pp 176-193

[15] Kagal, Finin, Joshi, "Trust-Based Security in Pervasive Computing Environments". IEEE Computer, December 2001

[16] Kim, Perrig, Tsudik, "Communication-efficient group key agreement," in Proceedings of IFIP SEC 2001.

[17] Kindberg, Zhang. "Context authentication using constrained channels". HP Labs Tech. report HPL-2001-84. 2001.

[18] Kong, Zerfos, Luo, Lu, Zhang "Providing robust and ubiquitous security support for mobile ad-hoc networks". In Proc. IEEE ICNP, pages 251--260, 2001.

[19] Langheinrich, "A Privacy Awareness System for Ubiquitous Computing Environments", UbiComp 2002, Springer LNCS 2498, pp 237-245

[20] Langheinrich, Mattern, Romer, and Vogt. „First Steps Towards an Event--Based Infrastructure for Smart Things". In Ubiquitous Computing Workshop, PACT 2000.

[21] Monrose, Reiter, Li, Lopresti, Shih. "Toward Speech-Generated Cryptographic Keys on Resource Constrained Devices". in Proceedings of 11th USENIX Security Symposium, 2002

[22] Noble, Corner. "The case for transient authentication". Presented at the 10th ACM SIGOPS European Workshop, September 2002

[23] Norman, "The Invisible Computer", MIT Press, 1999

[24] Orr, Abowd, "The Smart Floor: A Mechanism for Natural User Identification and Tracking", Georgia Institute of Technology, 2000

[25] Reiter, Aviel D. Rubin. "Crowds: Anonymity for web transactions". DIMACS Technical Report, 97(15), April 1997. 22

[26] Schmidt, "Implicit Human-Computer Interaction through Context", Personal Technologies, June 2000, pp. 191-199

[27] Schmidt, Beigl, "New Challenges of Ubiquitous Computing and Augmented Reality", 5th CaberNet Radicals Workshop, 5-8 July 1998, Valadares, NR. Porto, Portugal

[28] Sloman, Lupu. "Policy Specification for Programmable Networks". Networks (IWAN'99), Springer Verlag Lecture Notes in Computer Science 1999

[29] Stadler. "Publicly verifiable secret sharing". In EUROCRYPT '96, vol. 1070 of LNCS, pp. 191--199. Springer Verlag, 1996.

[30] Strasser, Rothermel, "System Mechanisms for Partial Rollback of Mobile Agent Execution". In: Proceedings of the 20th International Conference on Distributed Computing Systems (ICDCS 2000), IEEE Computer Society, Los Alamitos, California, pp. 20-28

[31] Want, Kenneth, Fishkin, Gujar, Harrison, "Bridging Physical and Virtual Worlds with Electronic Tags", Proceedings of CHI'99, ACM Press, April, 1999

[32] Weiser, "Some Computer Science Issues for Ubiquitous Computing", PARC 1993

[33] Zhou, Haas, "Securing ad hoc networks", IEEE Network, vol 13 pp. 24 - 30, 1999