Department of Computer Science
MSc Data Communications, Networks & Distributed Systems

# ZION

# A Lightweight SEINIT based Security Framework Implementation for Pervasive Computing

## Executive Summary

## September 2004

**Team X-Matrix**
Kumardev Chatterjee
Philip Ho
Wasif Mehdi
Fahd Shariff
Muhammad Solangi

**Supervisor**
Steve Hailes

This document introduces and summarises the goals, related work, design and implementation and conclusions for the project, Zion.

# Introduction

The ZION framework effort was a research and development project to develop as the title mentions, a lightweight security framework implementation for pervasive computing that was compatible with and indeed a pilot Proof of Concept demonstrator for SEINIT. SEINIT is a Security Experts Initiative to create, deploy and standardise security architectures, artefacts and frameworks for computing in ambient network environments.

# Problem Domain

Simply stated, the project envisions a dynamic working environment where an user/ application needs security at all times although his location/network conditions/environment, i.e. his ‘context’, changes frequently.

The questions arise; how does a system/framework understand a user/application’s ‘context’? How does it then use that understanding to create custom security for the user/application based on pre-defined security policies? Can something be devised that allows all of this to happen on a *single device* as well as possibly a networked way? How can all of this be done without making the user/application and the security system/framework tightly integrated; in-fact can they be relatively transparent to each other i.e. highly de-coupled?

# Summary of Goals

The goals of the project encompassed finding a solution to all of the above. A way of representing the ‘context’; evaluating it via a set of ‘rules’ and consequently tagging it in ‘some’ way; finding a way to write/design/implement High Level Security Policies which deal with such context tagging; a system that maps and understands such policies and context and can use them to perform ‘some’ security actions; a ‘sub-system’ that can be used to ‘discover and use’ artefacts that perform the required security actions and of-course designing a robust, standardised way to let applications and the system interact with each other while not being aware of each other’s code, configuration or even location.

# Summary of Related work

Given the bleeding-edge nature of the project goals, very few parallel examples, in the world of pervasive computing or even beyond, could be located. MIT’s ‘Oxygen’ and the project ‘Aura’ are perhaps the closest examples, although it must be stressed that though they do implement rule

based pervasive environments, they do not have much work in terms of providing a security framework implementation for such environments.

# Overview of Architecture and System Workings

The core of the system, Zion, is a Security Manager, which initiates/performs/monitors all actions in the system. It starts up first and initialises other components as and when necessary. A key component is the Application Manager, which once initialised, listens to all well-known SEINIT ports and optionally the SEINIT Multicast IP (explained below). It gets the latest of these by reading in a text file with the information. Once the Application Manager senses data on these ports it calls the Security Manager. The Security Manager detects the context of the device via the Context Manager. It then validates the context via the Business Rules Engine and uses the Ponder Manager to locate the Security Policy Level for this particular context. The Ponder Manager does that and lets the Security Manager know what kind of Security actions need to be taken (such as 32 bit encryption). The Security Manager then uses SATIN (a component based middleware developed by a UCL PhD student, Stephanos Zachariadis) to perform these actions via component loading and execution of low level security components. Once the actions are completed, the secure data is passed back to the originator application via the Application Manager.

The project envisages a proposed set of well-known Port numbers for applications that want to talk to SEINIT frameworks. It also proposes a SEINIT Multicast IP, SEINIT recognised Contexts, SEINIT business rules and the SEINIT Security Policy Levels; all proposed in detail and implemented convincingly. The design and implementation of these entities/artefacts done for this project, is hoped will be adopted/merged by SEINIT into published standards.

# Overview of Implementation

The project implementation included multiple green field technology implementations for the first time. These were the seamless successful integration of Ponder, SATIN, XML and JAVA (particularly Java Crypto classes). The architecture being component based, most development was done in silo with component owners doing their bits. Integration was done collaboratively. The entire system was built using J2SE (mostly J2ME compliant) and XML is the lingua franca of the system.

# Summary of Conclusions

The team successfully delivered a system which provides context based, High Level Security Policy driven secure data communication on a single device

and is multicast network capable as well. The system developed was tested and found to work fine for laptops and desktops with easy extensibility onto PDA’s and handhelds. A chat application was successfully built on top of the system. A comprehensive Audit Tool with an live, real-time illustrative State Transition Diagram display was also built. These two helped the team demonstrate the workings of the system and greatly helped with the testing and debugging. All system components were built to be scalable and easy to extend.

The team is upbeat that it managed to achieve all defined goals including all of the client’s later stage high priority expectations Finally, this robust implementation and demonstration of Proof of Concept opens up new vistas in the world of security framework architectures for pervasive environments.