UNIVERSITY OF SOUTHAMPTON

Information Security Risk Management for Ubiquitous Computing

by

Mohammed Zia Hayat

A thesis submitted in partial fulfillment for the
degree of Doctor of Engineering

in the

Faculty of Engineering, Science and Mathematics
School of Electronics and Computer Science

June, 2007

Blank Page

Publications

Z. Hayat, J. Reeve and C. Boutle, "Domain Based Security: Improving Practices", *In Proceedings CD of 1st Conference on DBSy*, Malvern, UK, July 2005.

Z. Hayat, J. Reeve and C. Boutle, "Prioritisation of Network Security Services", *In Proceedings of IEE Information Security*, Vol. 153, No.2, pp.43-50, 2006.

Z. Hayat, J. Reeve and C. Boutle, "Electronic Security Implications of NEC: Tactical Battlefield Scenario", *RUSI Defence Systems Journal*, Vol. 9, No.2, pp.110-113, 2006.

Z. Hayat, J. Reeve and C. Boutle, "Dynamic Threat Assessment for Prioritising Computer Network Security", *In Proceedings of 5th European Conference on Information Warfare and Security*, pp.61-70, Helsinki, Finland, 2006.

Z. Hayat, J. Reeve, C. Boutle and M. Field, "Information Security Implications of Autonomous Systems", *In Proceedings CD of 25th IEEE MILCOM Conference*, Washington DC, USA, 2006.

Z. Hayat, J. Reeve and C. Boutle, "Threat Analysis of Vulnerable Networked Services", *In Proceedings CD of NATO Symposium on Dynamic Communications Management*, Budapest, Hungary, 2006.

Z. Hayat, J. Reeve and C. Boutle, "Securing Autonomous Systems", *In Proceedings of 1st SEAS DTC Conference*, pp.B7, Edinburgh, UK, 2006.

Z. Hayat, J. Reeve and C. Boutle, "Threat Assessment in Heterogeneous IT Networks", *The ISSA Journal*, pp.8-11, February 2007.

Z. Hayat, J. Reeve and C. Boutle, "Ubiquitous Security for Ubiquitous Computing", *To appear in ISTR Journal*, Elsevier, 2007.

Z. Hayat, J. Reeve, C. Boutle, M. Field and P. Tuson, "Distributed Security for Decentralized Information Sharing", *Submitted to IEEE Systems, Man, and Cybernetics Part C*, 2007.

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Engineering

by Mohammed Zia Hayat

The potential for rapid and diverse interconnectivity between devices utilising heterogeneous communications interfaces has enabled a truly ubiquitous computing environment. However this has resulted in equally ubiquitous security risks due principally to the number and complexity of services being run over such networks. As technology advances towards the realisation of a ubiquitous computing environment, what impact does this have on the need to preserve the key information security requirements of: confidentiality, integrity and availability? And how does this influence, future information security solutions, particularly in light of "always-on" business processes which require real-time information sharing? This thesis describes research conducted into answering these questions from a risk management perspective, using key industrial projects as case studies.

Contents

List of Figures

List of Tables

Blank Page

Abbreviations

AA	Agent Assignment
AC	Attribute Certificate
ACL	Access Control List
ADS	Accreditation Document Set
ALADDIN	Autonomous Learning Agents for Decentralised Data and Information Networks
AS	Authorisation State
BDA	Battle Damage Assessment
C2	Command and Control
CD	Compact Disc
CIM	Common Information Model
CL	Capability List
CONSEQUENCE	Context-Aware Data-Centric Information Sharing
CONSERT	Configurable Systems Engineering Research Tool
COTS	Commercial Off The Shelf
CPU	Central Processor Unit
CVF	Carrier Vessel Future
DAC	Discretionary Access Control
DBSy	Domain Based Security
DCOM	Distributed Component Object Module
DDF	Decentralised Data Fusion
DFS	Depth First Search
DoD	Department of Defense
DoS	Denial of Service
DPA	Data Protection Act
DRBAC	Dynamic Context-Aware RBAC
DTC	Defence Technology Centre
DTE	Deliver Target Effect
DTI	Department of Trade and Industry
EPSRC	Engineering and Physical Sciences Research Council
ETS	Emit Target Stimulus
F2T2EA	Find Fix Target Track Engage Attack
FC BISA	Fire Control Battlefield Information System Application

GPRS	General Packet Radio Service
HLS	Home-Land Security
HMG	Her Majesty's Government
HP	Hewlett Packard
HTTPS	Hypertext Transfer Protocol Secure
IFPA	Indirect Fire Precision Attack
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IRM	Information Requirement Management
ISO	International Standards Organisation
ISTAR	Intelligence Surveillance Target Acquisition Reconnaissance
IT	Information Technology
JSP	Java Server Pages
JVM	Java Virtual Machine
KTN	Knowledge Transfer Network
M2M	Machine-to-Machine
MAC	Message Authentication Code
MANET	Mobile Ad-Hoc Network
MAS	Multi-Agent System
MBA	Master of Business Administration
MD5	Message Digest 5
MIT	Massachusetts Institute of Technology
MLS	Multi-Level Security
MoD	Ministry of Defence
MOSQUITO	Mobile Workers' Secure Business Applications in Ubiquitous Environments
MVCE	Mobile Virtual Center of Excellence
NAC	Network Access Control
NAP	Network Access Protection
NATO	North Atlantic Treaty Organisation
NCO	Network Centric Operations
NEC	Network Enabled Capability
NIST	National Institute of Standards and Technology
NRUC	National Road User Charging System
OCTAVE	Operationally Critical Threat Asset and Vulnerability Evaluation
OWASP	Open Web Application Security Project
P2P	Peer-to-Peer
PA	Privilege Assignment
PCIDSS	Payment Card Industry Data Security Standard
PDA	Personal Digital Assistant
PKI	Public Key Infrastructure
RAID	Random Array of Inexpensive Disks

RBAC	Role Based Access Control
RDD	Requirements Driven Development
RDP	Remote Desktop Protocol
RH	Role Hierarchy
RPC	Remote Procedure Call
SA	Situational Awareness
SAML	Security Assertion Markup Language
SEAS	Systems Engineering for Autonomous Systems
SETI	Search for Extra Terrestrial Intelligence
SHA-1	Secure Hash Algorithm
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SOA	Service Oriented Architecture
TCO	Total Cost of Ownership
TCP	Transmission Control Protocol
TST	Time Sensitive Target
UAV	Unmanned Air Vehicle
UK	United Kingdom
UML	Unified Modelling Language
UMTS	Universal Mobile Telecommunications System
USA	United States of America
USSR	Union of Soviet Socialist Republic
UWB	Ultra Wideband
VB	Visual Basic
VP	Vulnerability Period
VPN	Virtual Private Network
WiMAX	Wireless Broadband
XML	Extensible Markup Language

Blank Page

Acknowledgements

The author would like to thank his supervisors Dr Jeff Reeve and Chris Boutle, as well as the previous Engineering Doctorate secretary Anne Donahue. Jeff provided me with guidance whenever I required it, always ensuring that I understood the work in a wider context. Chris has been an inspirational manager always willing to listen and give the best possible advice he has facilitated many aspects of my work including project placements and providing necessary contacts. Anne introduced me to the Engineering Doctorate and helped me settle in the South of England, she is an incredibly dedicated lady.

In addition Chris Coles, Richard (Dick) Elder, Alex Graham, Paul Kimber and Tim Parsons have provided me with excellent support within BAE Systems.

I would also like to thank: Martin Field and Robert Johnston for their constant guidance and review of this work; Peter Tuson for developing the SEAS DTC soft model; Chris Wood for helping me with the SEAS DTC modelling scenario; Robert Mount for helping me with various graphics; David Green for introducing me to this fascinating subject during my undergraduate studies; Michael (Mo) Stevens, David Simpson, Stuart Miles, Larry Clark, David Charles and Joanne (Jo) Thoms for their help and advice during my time at BAE Systems.

The work described in this thesis was made possible by a sponsorship from BAE Systems and EPSRC, UK. Additional support was provided by the UK MoD for the work on autonomous systems.

To these people and institutions I am forever grateful.

Blank Page

To my parents, siblings and Henna.

Background

The research reported here has been driven by the requirements of the industrial sponsor, BAE Systems. BAE Systems is Europe's largest aerospace and defence company employing over 85,000 people worldwide. Since the end of the Cold War the defence industry has undergone vast change, where the emphasis particularly in the United Kingdom (UK) has gone from best-of-breed to best-fit. In simple terms this means reducing costs where possible. Therefore unlike in the past the military now actively seeks to leverage the best commercial technologies, due to rapid development cycles and economies of scale. The requirement for the use of commercial technologies for the military can be seen in the UK Ministry of Defence's (MoDs) quest to build a Network Enabled Capability (NEC) [1] using Commercial Off The Shelf (COTS) components wherever appropriate. NEC is an initiative known more widely in the North Atlantic Treaty Organisation (NATO) community as Network Centric Operations (NCOs). Its aim is to provide a ubiquitous computing capability for the military, interconnecting all information-based assets such as sensors and effectors to deliver more efficient and effective military operations.

With the military's need to leverage COTS components and the commercial world's drive towards improved information security, commercial and military information system security requirements can be seen to be converging. Therefore the aim of this research was to identify and improve information security best practices from both the commercial and military sectors, to deliver a secure future ubiquitous computing environment.

The Engineering Doctorate is a four year programme which comprises both technical and management studies. The first two years focus on identifying and scoping the research, through both taught and research technical studies. At the same time the business drivers of the research are identified through close collaboration with the sponsoring company and study of a significant portion of a Master of Business Administration (MBA) course. The final two years are spent undertaking research with the industrial sponsor.

Chapter 1

Introduction

Only the Paranoid Survive.

Andrew Grove, Former CEO of Intel Corp. [2].

The rapid evolution of Information Technology (IT) over the last few decades is leading to ubiquitous computing capabilities. This is evident in today's society, where there is an increasingly inherent reliance of governments, large corporations and the public at large on IT networks to carry out integral and critical tasks. Therefore the security of such key networks has arisen as a major business and political issue, it is the security of information which is considered in this thesis. As described in [3] the damage caused by attacks on IT networks is very significant: network down time can result in disruption to vital business processes; repairing compromised devices can take weeks and consume excessive amounts of people resource; and loss of reputation can be hard to quantify but can potentially be the most significant cost.

It is widely acknowledged that no mechanism alone or in combination can provide a *silver bullet* for security purposes. This is evident in the term *defence in depth* which is used in the information security industry, re-iterating the fact that no security solution can provide holistic protection. Therefore information security must be viewed as a form of risk management rather than risk prevention, incorporating both pro-active and reactive measures.

The International Standards Organisation (ISO) has defined ISO 7498-2 [4] as a security architecture reference model, which corresponds to the seven layer Open Systems Interconnection (OSI) model. The three key requirements of ISO 7498-2 are the need to preserve the confidentiality, integrity and availability of information. As technology advances towards the realisation of a ubiquitous computing environment, what impact does this have on the three key information security requirements of confidentiality, integrity and availability? And how does this influence future information security solutions,

particularly in light of always-on business processes which require real-time information sharing? This thesis argues that of the three key information security requirements availability has significantly evolved, increasing the need for pro-active and real-time information security solutions. To achieve such requirements research has been conducted in three principal areas, with an emphasis on the management of information security risks:

- Pro-active threat assessment for prioritising network security services.
- Novel access control and authentication mechanisms for secure decentralised information sharing.
- XML-based data validation for connecting sensitive information services.

Section 1.5 provides further details on the evolution of information security requirements and how the research undertaken here can help. Throughout this thesis the subject-privilege-object model is used to describe if an agent* (or subject) has sufficient privileges (read, write etc.) with respect to some information-based service (or object).

1.1 Confidentiality

Confidentiality requires the secrecy of information to be preserved, where only authorised agents can view the information in accordance with applicable confidentiality policies. The need to keep information secret is present in many walks of life, for example, sensitive enterprises such as government departments often restrict access to information. The first formal work in information systems security was motivated by the military's attempt to implement controls for the enforcement of the principles of *least-privilege* and *need-to-know*. Such principles also apply to businesses which keep proprietary designs secure from competitors. As a further example, all types of organisations keep personal (personnel and customer) information secret adhering to regulatory requirements such as the Data Protection Act (DPA) 1998 in the UK [5].

The seminal work by Bell-LaPadula [6] is used to define confidentiality of information in most modern information systems. In a traditional military sense agents are then assigned a *clearance level* and objects are assigned a *classification level*. Both clearances and classifications are said to be linear with specific levels of clearance authorising an agent to access objects up to a given classification. The system used in the UK government is summarised in table 2.1.

The Bell-LaPadula model states that in order to preserve confidentiality one must:

*An agent may be human or machine.

Object	Agent
Top Secret	Directed Vetting
Secret	Security Cleared
Confidential	Basic Check
Restricted	Counter Terrorism Check

TABLE 1.1: UK government information classifications.

- Prevent agents from accessing information at higher classifications than their authorisation permits (simple security property).
- Prevent unauthorised agents from declassifying information (*.property).

The Bell-LaPadula model can therefore be summarised as *no read up, no write down.*

1.2 Integrity

Integrity requires the state of information to be preserved, where only authorised agents can modify the information in accordance with applicable integrity policies. Several major integrity policies have been devised [7], [8] the foundation of which is the Biba [9] model. Historically in a commercial context the need to preserve accuracy of information stored or transported has taken precedence over confidentiality. This has recently started to change due to the uptake of e-business, where merchants such as online stores need to protect the privacy of sensitive personal data such as the address and credit card number of customers. The Payment Card Industry Data Security Standard (PCIDSS) [10] has been introduced by the major credit card companies to ensure that merchants who use their services preserve the confidentiality and integrity of consumer data. Therefore in such enterprises both Biba's integrity and Bell-LaPadula's confidentiality models must be preserved. Biba's model states that one must prevent an agent at a lower integrity level from modifying information at a higher integrity level, as it is less trusted information. Put simply this means only authorised agents should be able to change the state of information. This is a dual of the Bell-LaPadula model. The Biba model can therefore be summarised as *no write up, no read down.*

1.3 Availability

Availability refers to the ability of authorised agents to access information in accordance with applicable availability policies (or Service Level Agreements-SLA). Historically availability has been linked to robustness, which implies the need for fault-tolerance or graceful degradation in the event of failure or attack. The aspect of availability that

is relevant to security, is that someone may deliberately arrange to deny access to information by making it unavailable. This is also known as a Denial of Service (DoS) attack, which is most commonly performed against the initial handshaking phase (SYN) of the Transmission Control Protocol (TCP). Business continuity and disaster recovery plans are used to ensure the necessary availability of information systems. This includes regular data back up using techniques such as a Random Array of Inexpensive Disks (RAID).

1.4 Threats and Countermeasures

The following sections describe specific threats and countermeasures in terms of the key information security requirements for confidentiality, integrity and availability. Over time numerous mechanisms have been devised to counter threats and preserve the security of information, by far the most common and successful is the utilisation of a mixture of cryptography, hash functions, authentication and access control mechanisms as well as steganography.

1.4.1 Vulnerabilities

For the purposes of this thesis a vulnerability is defined as a potential flaw in a system including software or hardware faults, procedural weakness and mis-configurations. Such vulnerabilities may be exploited to violate the security of information. The violation need not actually occur for there to be a vulnerability. The fact that the violation might occur means that those actions that could cause it to occur must be guarded against or prepared for, such actions are called exploitations or attacks. The three security services of confidentiality, integrity and availability described previously can be seen to counter vulnerabilities, if preserved. The most common software vulnerabilities which are exploited by malicious agents are:

- *Buffer overrun* - An unchecked buffer in a program that can overwrite the program code with new data. If the program code is overwritten with new executable code, the effect is to change the program's operation as dictated by the attacker.
- *Privilege escalation* - Allows threat agents (or malware) to attain higher privileges in certain circumstances.
- *Validation flaw* - Allows malformed data to have unintended consequences.

There are numerous types of malware which can be used to maliciously exploit vulnerabilities, some of which include:

- *Virus* is an intrusive program that infects files in a single computer (unless manually transferred by a human) by inserting copies of self-replicating code. This can delete critical files, makes system modifications, or perform some other action to cause harm to data on a computing device. A virus attaches itself to a host program.

- *Worm* is a program which differs from a virus in that it can automatically (without manual human intervention) *self-replicate between computers*, they are often malicious, and can spread from computer to computer without infecting files first.

- *Trojan horse* is a piece of software or even an e-mail that professes to be useful and benign, but which actually performs some destructive purpose or provides access to malware.

- *Mail bomb* is a malicious e-mail sent to an unsuspecting recipient. When the recipient opens the e-mail or runs the program, the mail bomb performs some malicious action on their computing device.

The aim of the aforementioned vulnerabilities and threat agents is to violate the security of information systems through the following generic threat classes:

- *Snooping* is the unauthorised passive interception of information which can be countered by confidentiality services.

- *Modification* is the unauthorised alteration of information which can be countered by integrity services.

- *Spoofing* is the impersonation of one agent by another, luring a victim into believing that the agent with which it is communicating is a different agent. Spoofing can be countered by authentication mechanisms.

- *Repudiation of origin* is a false denial that an agent sent information it can be countered by authentication mechanisms.

- *Denial of receipt* is a false denial that an agent received some information it can be countered by authentication mechanisms and availability services.

- *Delay* is a temporary interruption of a service this can be countered by availability services.

- *Denial of Service* is a long-term interruption of a service this can be countered by availability services.

1.4.2 Cryptography

Cryptography is the art and science of keeping a message secure from threats such as snooping described previously. If a sender wants to send a message (*plaintext*) securely to a receiver preserving its confidentiality, then the sender must disguise the plaintext using *encryption.* An encrypted message is known as *ciphertext.* The process of turning ciphertext back into plaintext by the receiver is known as *decryption*, the whole process is illustrated in figure 1.1.

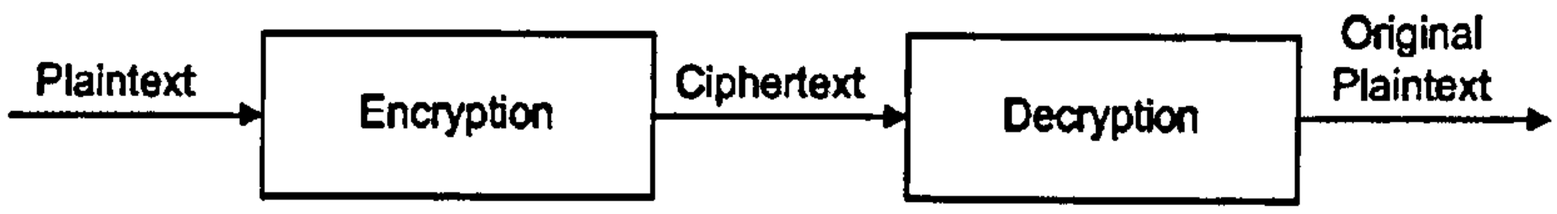

FIGURE 1.1: Process of encryption and decryption.

As well as preserving confidentiality cryptography can be used to preserve integrity and availability of information. For example Message Authentication Codes (MACs) can be used to check whether a received message is that which was intended by the sender, thus countering the threat from modification. Similarly SYN cookies [12] can be used to ensure DoS attacks on the TCP-SYN protocol are minimsed.

There are two main forms of modern cryptographic algorithms for encrypting and decrypting information, these are symmetric and asymmetric techniques[†]. Symmetric [12], [14] and asymmetric [15], [16] algorithms rely on the difficulty of predicting a key to securely transmit information. Symmetric algorithms use the same key for both encryption and decryption, whereas asymmetric algorithms utilise two distinct keys for the encryption and decryption processes. A generic symmetric cryptographic process is illustrated in figure 1.2, where it can be seen that in order to preserve the security of a message the *secret key* used to encrypt and decrypt the data must be known only to the sender and intended receiver(s). In figure 1.3 a generic asymmetric cryptographic process is illustrated where it can be seen that a *public key* is used to encrypt the message and a corresponding *private key*[‡] is used to decrypt the message.

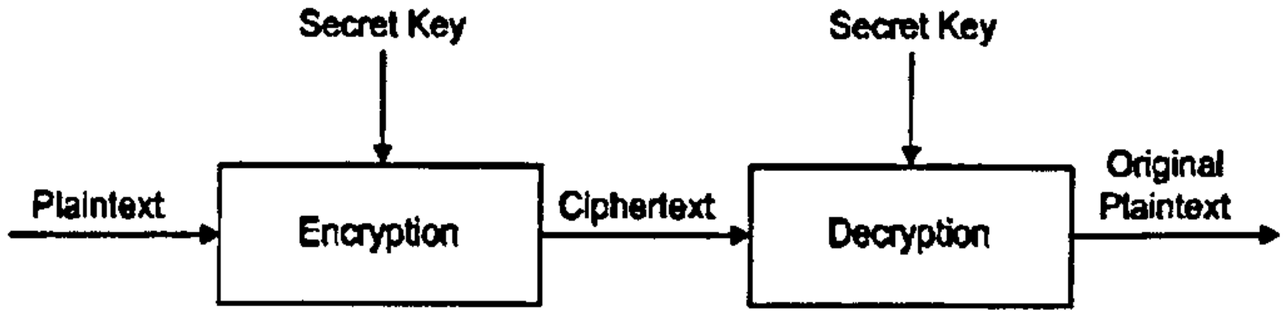

FIGURE 1.2: Generic symmetric encryption and decryption process.

[†]It has been shown how asymmetric algorithms can be simulated using symmetric techniques in conjunction with hash functions [11].

[‡]It is assumed, that the private key cannot be calculated from the public key in a reasonable amount of time, to compromise the security of any message(s) transmitted using such a method.

A major issue with symmetric cryptography algorithms is the *key distribution problem*, which relates to the way in which the secret key is distributed between transmitting and receiving systems. Whereas most asymmetric algorithms suffer from intense computational power requirements. One solution is to use the asymmetric encryption method to distribute secret keys between communicating end-points, which may then use the more efficient symmetric cryptography algorithms for subsequent secure communications. This assumes one can securely identify relevant public key(s), for example by using a trusted third party to act as a trust bootstrapping mechanism by signing and therefore certifying public keys or the public parameters in a Diffie-Hellman [17] key exchange.

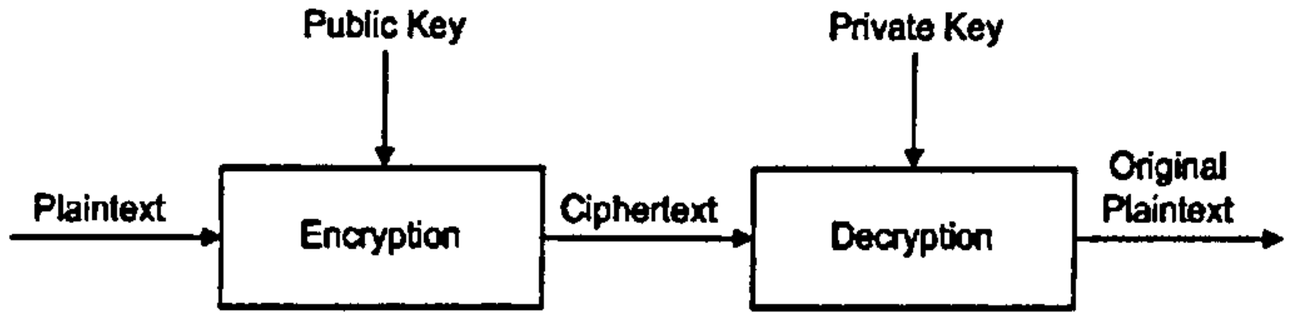

FIGURE 1.3: Generic asymmetric encryption and decryption process.

Asymmetric cryptography algorithms also provide the ability to counter the denial of origin threat through digital signatures. As only the authorised agent has knowledge of the private key corresponding to its published public key, if a message is signed with an agents private key this provides a high level of assurance[§] that the message originated from that agent.

1.4.3 Hash Functions

A hash function is a function mathematical or otherwise which converts a variable-length message to a fixed length value. Central to the concept of cryptographic hash functions is that they are said to be only *one-way*. That is given a pre-image a hash value can be calculated, however given a hash value it should be infeasible (within a reasonable time) to calculate the pre-image used to derive this hash value. A good cryptographic hash function is also required to be *collision-free*, meaning it should be hard to generate two pre-images which hash to the same value. Numerous functions exhibiting such behaviour exist, the most prominent of which are the MD5 [18] and SHA-1 [19] hash functions. As described previously hash functions can be used to preserve the *integrity* of information (pre-image), which if it is sent securely with its hash function cannot be accidently modified due to the one-way and collision free properties of hash functions. It must be noted that a MAC is a hash function which requires knowledge of a secret key, this can be used to authenticate the transmitter as well as preserve the integrity of the information.

[§]Providing the private key has not been compromised.

1.4.4 Authentication Mechanisms

Authentication involves the binding of an identity to a subject. Such a binding requires the subject to provide necessary information to confirm its identity to the information system. This information comes from one (or more) of the following, authentication factors:

- What the subject knows (such as passwords or other secret information).
- What the subject has (such as a badge or card).
- What the subject is (such as a fingerprint or other unique characteristic).
- Where the subject is (such as in a particular location).

The authentication process consists of obtaining the authentication factor information from the subject, analysing the data, and determining if it is associated with that subject. This means the system must store and manage authentication information for each subject.

1.4.5 Access Control Mechanisms

In modern information systems there are two main forms of access control mechanism:

- Access Control Lists (ACLs)
- Capability Lists (CLs)

ACLs maintain a list of all objects with associated rights[¶] for each agent. Whereas CLs maintain a list of agents with associated rights for each object. It must be noted that ACLs are used to a greater extent [20] in modern information systems, due to their efficiency in computing: *which agents can access an object & how, given the object*, this is required most frequently for traditional access control. As opposed to the question of: *which objects can an agent access, given the agent.*

1.4.6 Steganography

Steganography is used to hide secret messages in other messages, such that the very existence of a secret is concealed. Generally the sender writes an innocuous message and

[¶]Rights may be in the from of read, write, execute etc.

then conceals a secret message in the same message. Historically this was achieved using invisible inks, minute differences between handwritten characters etc. More recently, computer graphics have been used to conceal secret messages. For example replacing the least significant bit of each byte of an image with bits of a message. Due to gradations, which are un-noticeable to the human eye in most graphical standards the image will not change appreciably for one to notice. In [21] it is shown how such traditional forms of computing steganography can be used to conceal the existence of communications.

Steganographic techniques may also be used for authentication. Instead of the traditional challenge response where a user authenticates to a system through a username-password combination; a user may perform a certain sequence of events such as clicking on certain parts of a screen within time constraints. This would provide an alternative to the traditional username-password challenge mechanisms required for user authentication and would even conceal the very existence for the ability of one to utilise a system. Such work builds on previous efforts [22]-[26] into using visual techniques for login. It is felt that with the advances in artificial intelligence, machine learning algorithms can be tuned to identify the authenticity of a users actions, by analysing characteristics such as speed and movement of a mouse when carrying out such stegonagraphic authentication.

1.5 Evolving Information Security Requirements

From an enterprise perspective ubiquitous computing is facilitating always-on (24-7) processes, where users can access any information-based service from anywhere at any time. Therefore ever more critical decision making is based upon information delivered over IT networks, where information security requires the confidentiality, integrity and availability of information to be preserved. From an information security point-of-view the requirements for preserving the confidentiality and integrity of information described in sections 1.1 and 1.2 remain the same in modern networks. As discussed in section 1.3 availability is the third key requirement of information security and has traditionally related to robustness. However in an "always-on" business environment it is argued that availability necessitates timeliness on a par with robustness, where timeliness implies that all relevant information is delivered to authorised agents at the correct time. Timeliness is essential in enabling operations which require real-time processing and therefore information sharing.

There are numerous examples where the need to improve the timeliness of current information security processes and procedures is evident, three such cases have been studied as part of this research:

- Prioritising network security services through pro-active threat assessment.

- Secure decentralised information sharing through novel access control and authentication mechanisms.
- Connecting sensitive information services through XML-based data validation.

1.5.1 Network Security Services

The importance of cryptography and access control mechanisms was outlined in section's 1.4.2 and 1.4.5. However these techniques alone do not provide holistic protection, because the software which performs security as well as other functions may become exploitable due to previously unknown design and/or implementation vulnerabilities. Therefore vendors such as Microsoft periodically release software updates (patches) to overcome vulnerabilities. For end users the process of integrating patches into the system is known patch management. Patch management involves identifying affected devices, undertaking configuration testing (to ensure updates don't adversely affect processes) and updating all affected devices. Patch management is an instance of a network security service, which forms part of an overall defence-in-depth security strategy aiming to efficiently and effectively manage information security risks. Other forms of network security service also exist such as event log analysis. Event log analysis is performed by systems administrators to ensure all critical processes are running as expected, and potential security breaches such as the malicious probing of firewall's is identified and countered. Clearly such activities can be very time consuming and costly, particularly when many hundreds of heterogeneous devices need to managed by a relatively small team of network administrators. The research described in chapter 2 builds upon work undertaken as part of the Carrier Vessel Future (CVF) project, in particular it details:

- Why delays in performing critical network security services such as patch management, are likely to introduce unacceptable levels of risk.
- A potential solution for reducing such risks according to business needs, this is based upon a pro-active threat assessment technique.

1.5.2 Decentralised Information Sharing

In many business processes fine-grained authentication and access control decisions are taken by a central security-aware agent. For example if a user subscribes for access to the nearest eating places of interest with their mobile phone operator, this information will be pushed to their mobile handset, this is known as location-based services. In such scenario's a dedicated and centralised back end system is sufficient to control the dissemination of relevant information to individual subscriber devices. However in the near future ad-hoc networking capabilities are likely to enable business operations which

require decentralised interactions between agents. For example in law enforcement a patrol officer may require real-time access to sensitive information such as a reconnaissance video stream directly from an unmanned sensor. In light of this need for timely as well as secure information exchange through decentralised principles, chapter 3 describes research conducted as part of the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre (DTC), with an emphasis on:

- The limitations of traditional access-control and authentication techniques.
- A potential context-based access control and authentication solution.

1.5.3 Connecting Sensitive Information Services

Many sensitive information-based services are currently physically partitioned from other services. In such circumstances human operators are often employed to transfer information from one service to another through manual mechanisms such as re-typing data seen on one screen to another. This is done to minimise the potential for security breaches, by controlling the flow of information in and out of such services, therefore preserving the confidentiality and integrity of information. The use of a human operator to control the flow of information between machine agents such as unmanned sensors and effectors is likely to introduce significant bottlenecks in future time-sensitive operations. Chapter 4 describes research conducted as part of the Indirect Fire Precision Attack (IFPA) project, it argues that the security role currently performed by human operators can be fulfilled in a more efficient and effective manner using electronic processes, additionally it discusses:

- The limitations of current physically partitioned infrastructures when considering future military operations.
- XML-based techniques as a potential off-the-shelf data validation solution.

Blank Page

Chapter 2

Prioritisation of Network Security Services

On large networks security administration tasks such as patch management and event log analysis can take many hours and even days to successfully complete even with automated solutions. Currently it is left to the systems administrators' discretion to choose in which order to protect individual devices. In light of the rapidly decreasing time between vulnerabilities being discovered and maliciously exploited by malware, such an arbitrary method introduces an unacceptable level of risk to the security of those devices, which are critical to business processes.

An information risk management approach needs to be adopted to ensure the protection of the network with a high likelihood of success; this can be achieved through the prioritisation of critical devices. In this chapter a generic prioritisation technique for individual devices in a network is described offering a methodical alternative to the current ambiguity of a systems administrators operations. The technique is based upon compromise path analysis, which identifies critical paths in a network from a security viewpoint and is relevant in a wide range of operations from the application of security services to analysing their results. The concept of a Vulnerability Period (VP) metric is introduced, as a mechanism to control the risk exposure to individual devices through prioritisation.

This chapter is organised as follows, section 2.1 provides background information on the drivers for a prioritisation strategy. Section 2.2 describes a simple prioritisation strategy and highlights its limitations, section 2.3 details how connectivity in modern IT networks has evolved and how this has impacted the security risks, resulting in the investigation into compromise path analysis to quantify such risks, this is followed by an overview of how networks are modelled in this work. In section 2.4 the algorithm developed to prioritise the order in which individual devices should receive security servicing

is specified and explained through an example. An overview of preliminary results from testing carried out on a software implementation of the algorithm is provided in section 2.6, with a summary and potential future work detailed in section 2.7.

2.1 Rapid Exploitation of Vulnerabilities

It is argued that if the current trend of reducing time scales in exploitation and increasing number of software vulnerabilities continues as reported in [27] then, critical business processes will be exposed to unacceptably high levels of risk, even in the presence of automated security services such as path management as advocated in [28]. Therefore the efficient and effective utilisation of available security resources is required in order to further exploit their potential. This is highlighted in [29] where it is suggested that many organisations fail to gain real benefit from their investments in IT systems. Prioritisation of devices to receive security servicing is an approach, which may be used to enable this, reducing the time frame within which critical assets may be compromised. This time frame was introduced in [30] as VP. The VP for each susceptible device is the time between a vulnerability first being reported to the time at which that device is made secure from such a vulnerability. The VP is defined as follows,

$$VP = P \cdot \tau + K \tag{2.1}$$

where P is the priority (integer value starting from '1', which is the highest priority) assigned to a given device, τ is the average* time taken for a service to be successfully performed per device; τ is also variable due to differences in: individual services, network latencies and dynamic characteristics of individual devices; K is the time taken for the developers to provide a solution to fix the vulnerability from the time of its discovery.

In [3] Brown et al. describe a pro-active malware susceptibility testing technique known as Active Countermeasures, which applies a vaccine (a virus with a NULL payload) to devices on a network. An automated response is sent back from the device under investigation indicating its susceptibility to the virus and if a device is found to be susceptible it is immediately made safe. The scanning of individual devices is based upon a SETI@home-style [31] setup. Using this technique networks are separated into clusters of devices with each cluster being assigned to a given scanner for inspection purposes. However this process provides the service to devices in an arbitrary sequence. From equation 2.1 it can be seen how prioritisation unlike an arbitrary technique can aid the owners of devices to control the risk exposure to a given device by reducing or increasing its VP accordingly. Without prioritisation the expected VP can be any time between

*It must be noted that other measures such as longest/shortest time could also be used to represent a more cautious/optimistic time frame.

one and n (number of devices under consideration) times the average time (τ) taken to service a single device.

The use of prioritisation to manage information security risks in IT networks is endorsed in [32] and [33]. In both of these papers the authors identify the requirement for determining the priority and therefore sequence in which individual devices receive security servicing. The question is how to develop a systematic technique for achieving such prioritisation? In [34] the authors introduce Operationally Critical Threat, Asset and Vulnerability Evaluation (OCTAVE) a high-level strategy designed to identify the relative criticality of information and infrastructure assets. Although this methodology provides a valuable context-driven systematic approach to managing information security risks, it does not directly address the issue of how one would technically achieve prioritisation and therefore control the risk exposure through the length of the VP for individual devices.

The application of security services such as automatic updating of anti-virus patching and signatures, security event log analysis as well as stateful (deep) packet analysis are a few examples of where a prioritisation capability could be employed as part of an overall quality of security service strategy, reducing the impact of potential compromises. For example in the case of deep packet analysis, the excessive overheads associated with scanning network communications as described in [35] means only a subset of the total traffic can be scanned. Therefore a prioritisation technique would prove useful in differentiating between network traffic for analysis purposes, based upon its potential impact if it was to compromise the end-point(s).

Security services may also be significantly degraded if the infrastructure used to facilitate them is compromised and brought offline. This is particularly true if the resources (CPU time and communications bandwidth) of 'ordinary' devices on the network are required to enable such services, which is the case for the SETI@home-style setup used in the Hewlett Packard (HP) Active Countermeasures technique as described previously. It is argued that if a network was to use a security service(s) based upon such an architecture then the service(s) would be severely affected if a worm such as Sasser [36] was to attack it, rapidly limiting available resources. In such a scenario a prioritisation technique would enable the scarce resources available to be focused at those devices perceived to be of greatest business value.

2.2 A Simple Prioritisation Strategy

In order to develop any sort of information security risk strategy, one must derive a security criticality classification system. This involves the identification and valuation

of assets following which prioritisation can be derived, based upon the risks to those assets. Traditionally a classification system consists of a number of criticality levels, where individual devices are assigned to one such level. There may be a different classification system for each of the three key information security requirements of confidentiality, integrity and availability however, for the purposes of the modelling described here only one classification system is used. This is analogous to that commonly used by many organisations for information security classifications. The criticality levels used in this system are (from high to low criticality): VH, H, M, L and S.

A simple prioritisation strategy for security purposes would be to order the application of security services based upon the criticality level to which a device is assigned, hence all devices with a criticality level of 'VH' would be assigned the highest priority for security service applications. However this may still result in a relatively arbitrary prioritisation of devices if there are a significant number of devices at the same criticality level in a network. This is illustrated in equation 2.2,

$$P_s = \frac{1}{\eta} \tag{2.2}$$

where it can be seen that as the total number of devices (η) at the same criticality level in a network increases, the probability of any one device (P_s) receiving the appropriate prioritisation decreases, as they could be serviced in any order. Therefore the probability of every device (P_a) receiving servicing in the correct order at any given criticality level can be defined as,

$$P_a = \frac{1}{\eta!} \tag{2.3}$$

Equation 2.3 further highlights the limitations of an arbitrary servicing strategy when considering all devices at the same criticality level.

From equations 2.2 and 2.3 it is clear that in order to successfully prioritise, one requires a differentiating factor(s) between a set of devices at the same criticality level, otherwise the VP for assets will remain undetermined, where large numbers of devices are assigned to the same criticality level. One way of achieving such differentiation is to identify and analyse dynamic risks to individual devices and then prioritise the application of security services based upon these perceived risks. Numerous techniques ranging from software development lifecycle best practice to attack trees [37]-[41] exist for the identification, analysis and ranking (prioritisation) of risks in a network scenario. However a major limitation with all of these techniques is that they assume an exhaustive search of the problem space, which means identifying individual risk(s) to device(s) and then quantifying these, based upon the attacker model. This search can be vast and

complex in the case of security vulnerabilities in modern IT networks.

In order to achieve the goal of pragmatic risk management it was decided to abstract from specific instances of threats, instead concentrating on perceived threats from other devices based upon their criticality level. The classification system used for the purposes of this work assumes that individual devices are assigned to a criticality level based upon the information and services they host. In order to reduce security risks to devices with increasing criticality a number of assumptions have been made for the purposes of this model. Devices of increasing criticality are said to be:

- Accessed by authorised users in whom one has higher degrees of confidence
- More frequently monitored for abnormalities
- Given higher prioritisation for anti-malware purposes

Based upon the aforementioned assumptions the security risk associated with devices of increasing criticality are likely to decrease.

2.3 Evolving Risks in Evolving Networks

One of the major changes in computing networks over the last few years has been the ability to conveniently form agile business processes through paradigms based upon wireless ad-hoc communications, Service Oriented Architectures (SOAs) and in the longer term grid (or utility) computing. This has resulted in rapid and complex interconnections between devices previously unimaginable. Such complex interconnections result in open and dynamic connectivity across traditional networking boundaries not only at a personal or home and small office level, but increasingly in the constrained environment of the larger corporate and government network, where sensitive information and services must be protected.

The majority of enterprise users today utilise computing devices with constrained communications capabilities for a limited set of well-defined services. However in the near future these devices will be used to deliver much more varied and flexible services, using the most functionally and cost effective communications technique available. This can be seen in the desktop computer which is no longer only enabled for communications over the wired corporate backbone. Instead it is increasingly capable of communicating through a multiplicity of interfaces such as Infrared and Bluetooth. In addition to the desktop other devices such as laptops and Personal Digital Assistants (PDAs) are also being introduced enabling ubiquitous capabilities through mobile and heterogeneous[†]

[†]WiFi, General Packet Radio Service (GPRS), Universal Mobile Telecommunications System (UMTS), Ultra Wideband (UWB) and Wireless Broadband (WiMAX).

communications using techniques such Virtual Private Networks (VPN).

Another key driver of the complex interconnectivity between devices will be the enablement of an all Internet Protocol (IP) communications system through the concept of IP Multimedia Subsystem (IMS) [42]. IMS is being developed by the telecommunications industry, allowing individual devices such as mobile phones to directly communicate pushing and pulling content and services to and from one another in a distributed fashion unlike the principally client-server internetworking model of the past. An example of how the traditional internetworking environment is evolving is given in figures 2.1 and 2.2 respectively. From figure 2.2 it can be seen that a much more complex and meshed environment is developing, where every device has the potential to become a gateway to external networks. The complexity introduced by such extended connectivity adds to the threat vector in IT networks.

Although the business benefits of using a more a flexible approach to internetworking are clear the complexities introduced pose significant technical challenges, not least from a security point-of-view. One such challenge is to provide tailored security services according to the diverse needs of individual users and their devices as described by the Jericho forum [43] and their vision of de-perimeterised security solutions. De-perimeterisation can already be seen to be taking place, where distributed or embedded firewall's are utilised on individual devices, supplementing the static and skeletal security model of the monolithic perimeter firewall solution.

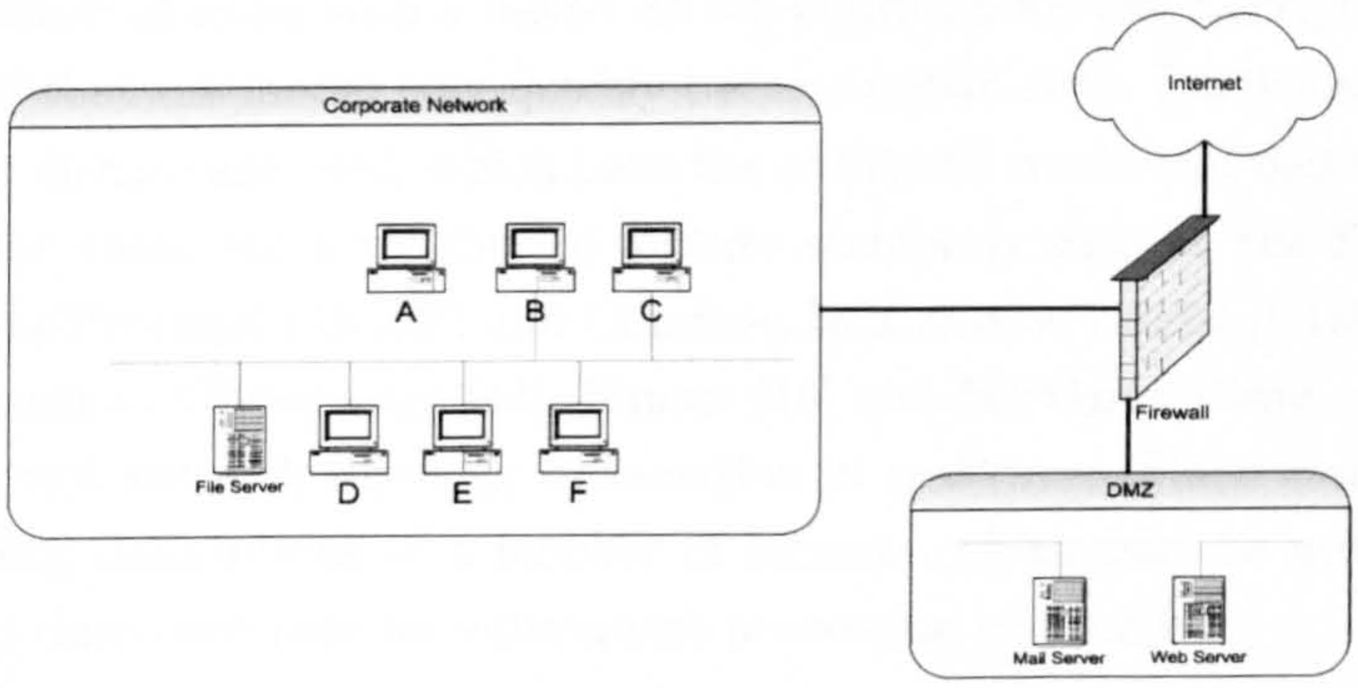

FIGURE 2.1: Traditional networking scenario.

2.3.1 Network Modelling

Throughout this chapter the network is represented as a graph $G=\langle D,C\rangle$, where D is the set of physical devices (d) and C is the set of inter device compromise connections (c). Semantic network modelling is used as described in [33] to represent compromise network connectivity, where a compromise connection (c) represents the potential to

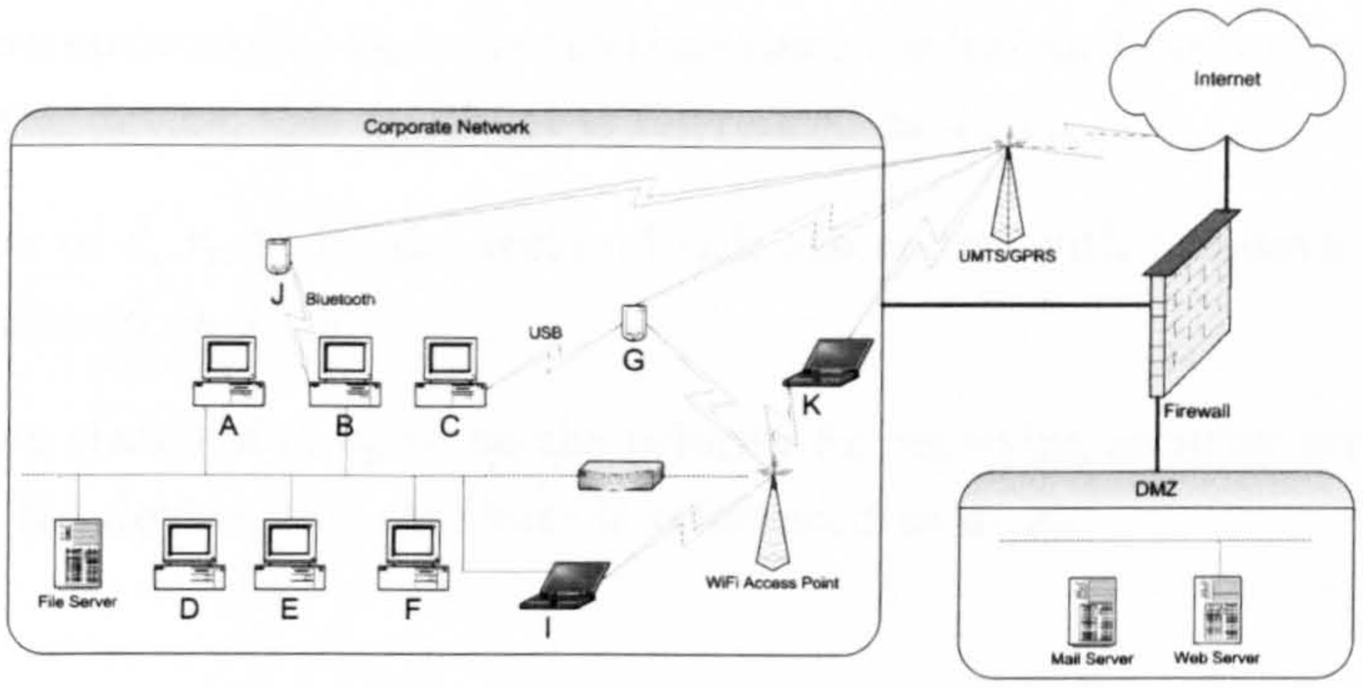

FIGURE 2.2: Evolving networking scenario.

exploit a flaw(s) by one device in another therefore enabling the spread of malware such as viruses and worms. For example if a device hosts a particular web service application, which is subsequently found to have a flaw (e.g. buffer overflow) then all external devices authorised to access the application are then considered to have a compromise connection to the device hosting the vulnerable application. In [33] Monahan identifies the need to use either directed or un-directed edges with the added concept of attributes to provide further details on particular protocols (e.g. http, https) being run between devices. In the model used here a simplified version of this is used, where all links between devices are undirected and attributes are not considered for simplicity.

A criticism of using such a model driven approach for the description of an IT architecture, is that the model may quickly become out-of-date. Particularly where mobile and ad-hoc devices are used, which have the ability to create and destroy links dynamically. However there are a number of mature standards such as the Simple Network Management Protocol (SNMP) and Common Information Model (CIM) with corresponding tools such as **Cheops-ng** [44], **Nmap** [45] and **HPOpenView**, which provide semantically rich network topology information in real-time. These standards are capable of providing descriptions in a number of formats, which may be used by techniques such as that described here for subsequent processing and analysis.

Let us first define, $\forall d \in D$:

- The criticality level of d, c_r to be the security criticality associated with the device, this attribute is referenced as $d \cdot c_r$
- The compromise path risk level of d, p_r to be the risk level of the highest threat compromise path associated with the device, this attribute is referenced as $d \cdot p_r$
- The compromise path length of d, h_o to be the length of the compromise path associated with the device, this attribute is referenced as $d \cdot h_o$

- The anti-malware update of d, a_m to be the time since the last anti-malware update was applied to the device, this attribute is referenced as $d \cdot a_m$
- The residual risk of d, r_r to be the residual risk associated with the device, this attribute is referenced as $d \cdot r_r$
- The prioritisation status of d, s_p to be the priority for receiving security servicing associated with the device, this attribute is referenced as $d \cdot s_p$

And, $\forall c \in C$:

- The risk level of c, v_b be the connection risk level associated with the connection, this attribute is referenced as $c \cdot v_b$
- The distinct devices incident to c, e_s be the two devices associated with the connection, this attribute is referenced as $c \cdot e_s$

From the seven attributes of a device four (c_r, p_r, h_o and r_r) are used to identify the relative risk posed to individual devices in the network G. The technique proposed here uses at least one and at most all four of these attributes to prioritise devices. It is assumed that $\forall d \in D$ the administrators of the network G are able to identify the security criticality level c_r and the time since the last anti-malware update a_m, where c_r is calculated according to a technique such as Her Majesty's Government (HMG) infosec security classifications or OCTAVE [34]. Devices with a higher criticality level c_r have a higher priority, where $c_r \in$ {VH, H, M, L, S}. The time since the last anti-malware update a_m is likely to be available to administrators from a network security management system which maintains the status of all devices under one's administration, where $a_m \in$ {0, 1, 2, 3, ..., 24}. The value of a_m relates to the number of hours since a device's last anti-malware update, where devices which have not received an update for more than 24 hours are assigned a value of 24.

2.3.2 Compromise Path Analysis

Although IP networking has the potential to allow ubiquitous connectivity, communications between devices are limited due to routing restrictions imposed by security services (e.g. embedded firewalls). This implies that if an attacker wants to compromise a given device(s) (victim) using a worm for example, then they must launch an attack from a device(s) which is authorised to connect to the victim otherwise the attempt to connect will be rejected. Here such compromise connections are known as compromise paths, where a compromise path consists of one or a series of compromise connections as described in section 2.3.1. This implies that an attacker has to systematically traverse a number of devices and overcome various security barriers (defence in depth strategy)

such as IP routing rules, traffic analysis and packet inspection to compromise a specific device in a specific manner.

The technique of compromise path analysis as adopted in this work is based upon that developed by QinetiQ for their Domain Based Security (DBSy) methodology [46]. As far as the author is aware there is no other strategy in the literature similar to compromise path analysis, which explicitly identifies the threat to a device based upon the graph theory concept of reachability. Although work on identifying threat paths to whole sub-nets has recently been identified at the Massachusetts Institute of Technology (MIT) [47]. Here security criticalities are used to bound the search space, by only analysing for compromise connections to devices of equal or lower criticality, as they are deemed to pose the main risk. In the DBSy model, compromise paths are rated according to a technique analogous to table 2.1, this is part of an HMG infosec standard [48], for quantifying risks when connecting devices of differing security criticalities.

Devices with a criticality of 'S' are not prioritised at all as they are believed to be out of the current administrative authorities control, they are only used as potential attackers in our compromise path analysis technique. Thus the set of devices in a network G which are prioritised (potential victim devices) is: $D_M = \{d : D | d \cdot c_r \neq \text{'S'}\}$. Therefore if for a set of devices, $D_s \subseteq D_M$, one is unable to prioritise between them based upon security criticality levels (c_r) alone (i.e. they have the same criticality level), then the highest compromise path risk (p_r) associated with such devices may be used to distinguish between them. Where, $\forall d \in D_s$, devices with higher compromise path risk levels are given increased priority.

Victim\Attacker	VH	H	M	L	S
VH	1	2	3	4	5
H	-	1	2	3	4
M	-	-	1	2	3
L	-	-	-	1	2
S	-	-	-	-	1

TABLE 2.1: Risk levels for connected devices.

From table 2.1 it can be seen that a path between a device of 'VH' criticality and one of 'L' criticality has an associated risk level of '4', where a level '1' risk is the lowest level and a level '5' is the highest risk level. It is also illustrated (i.e. lower half of table is irrelevant) in table 2.1 that devices are only perceived to be at risk from those of an equal or lower criticality level.

2.3.3 Extending Compromise Path Risk Analysis

As well as using compromise path risk levels to distinguish between a set of devices as is described in the original method in [46], the technique described here builds upon this by also using the concepts of compromise path length and residual risk. Therefore if a set of devices $D_H \subseteq D_M$, have the same criticality and compromise path risk levels, then compromise path length (number of hops between victim and attacker device) is used to distinguish between them. Where, $\forall d \in D_H$, devices with lower compromise path lengths are given higher priority. This is due to the fact that the attacker device ($d \in D$) is closer to the victim device ($d \in D_H$) resulting in less effort to compromise the victim on behalf of the attacker. The risk analysis process is further extended by introducing the concept of residual risk, therefore if a set of devices, $D_R \subseteq D_M$, have the same criticality level (c_r), compromise path risk level (p_r) and compromise path length (h_o), then residual risk (r_r) is used to distinguish between them. Where, $\forall d \in D_R$, devices with higher residual risk are given higher priority. The residual risk (r_r) of a device $d \in D_R$ is calculated as,

$$d \cdot r_r = \frac{(\Sigma c \cdot v_b + a'_m)^2 + d \cdot a_m}{\zeta} \tag{2.4}$$

where ζ is the number of directly connected devices which have an equal or lower criticality level than the device (d), and v_b is the connection risk level for such connections ($c \in C$) and is calculated by comparing the difference in criticality levels according to table 2.1 of the two devices in $c \cdot e_s$. In-line with HMG infosec standard 3 [3] the use of ζ in equation 2.4 ensures it is biased (associates a higher risk value - $d \cdot r_r$) towards those devices which are connected to higher risk devices. Therefore if the numerator of equation 2.4 is the same for two devices, however one is connected to less devices (i.e. lower value of ζ) then the one with a lower value of ζ is perceived to be at a greater risk. In contrast to [30] the time elapsed (hours) since last anti-malware update for each device is now incorporated into the residual risk calculation, where a'_m reflects this value for all directly connected devices to d of an equal or lower criticality, and $d \cdot a_m$ is the time since the last anti-malware update for device d itself. The use of anti-malware update times (a_m) provides real-time risk assessment. This is similar to the way in which many Network Access Control (NAC) mechanisms such as Microsoft's Network Access Protection (NAP) [49] operate. In NAP devices which do not adhere to anti-malware update policies are appropriately remediated through quarantining and updating of anti-malware patches.

The risk posed to a device is quantified by assigning the four metrics described previously to attributes of each potential victim device $d \in D_M$ in the network G. The attributes corresponding to each metric and in order of precedence are:

1. c_r - Criticality level.
2. p_r - Compromise path risk level.
3. h_o - Compromise path length.
4. r_r - Residual risk factor.

Therefore a metric of lower precedence is only used if a higher order metric is unable to provide prioritisation for a given set of devices.

If for a device $d \in D_M$ both p_r and h_o attributes have a value of '0' this indicates no compromise path exists for d. Otherwise if a compromise path exists p_r takes a value, $1 \leq p_r \leq 5$, where '1' represents the lowest risk and '5' the highest; and h_o takes a value, $h_o > 0$. The r_r attribute is calculated to distinguish between devices, which cannot be prioritised using a combination of c_r, p_r, and h_o attributes, this is specified in the algorithm in section 2.4, where $r_r \geq 0$.

2.4 Specification of the Algorithm

For the purposes of the algorithm to be described, a number of terms are defined to aid in its understanding.

Definition 1: Let $D_M{}^{(x)} \subseteq D_M$ denote the set of devices with a security criticality of x.

Definition 2: $\forall d \in D_M$, let $p(d)$ calculate the highest risk compromise path of device d, calculated according to table 2.1 using a constrained[‡] Depth First Search (DFS). If two or more compromise paths of identical risk level exist for one device then $p(d)$ chooses the one with the smallest length, choosing any one if more than one has the same length as well as risk level. Formally $p : D_M \rightarrow \{1, 2, 3, 4, 5\}$.

Definition 3: $\forall d \in D_M$, let $h(d)$ calculate the length of the highest risk compromise path to d. Formally $h : D_M \rightarrow \{n : N | n > 0\}$, where N is the set of natural numbers.

Definition 4: $\forall d \in D_M$, let $r(d)$ calculate the residual risk (r_r) of the device d calculated according to equation 2.4. Formally $r : D_M \rightarrow \{n : R | n \geq 0\}$, where R is the set of real numbers.

[‡]It must be noted that a number of limitations (attacker device must have a higher criticality level than the current attack criticality level considered and lower criticality than the victim device) have been imposed upon the search algorithm to ensure it is computationally feasible, whilst providing a comprehensive analysis from a security point-of-view.

The consolidated prioritisation technique is recursive and the algorithm is:

1. $\forall d \in D_M$ set $d \cdot p_r = 0$, $d \cdot h_o = 0$, $d \cdot r_r = 0$ and $d \cdot s_p = 0$
2. **foreach** x:=VH:L let $D_S := D_M{}^{(x)} \subseteq D_M$
3. **if** $|D_S| > 1$ **then**
4. $\forall d \in D_S$, let $d \cdot p_r := p(d)$ and $d \cdot h_o := h(d)$
5. $D_H := \emptyset, n := |D_S|$
6. **for** $t := 1 : n$
7. **for** $i := 1 : n$
8. **if** $((d_t, d_i \in D_S) \wedge (i \neq t) \wedge (d_t \cdot p_r = di \cdot p_r) \wedge (d_t, d_i \notin D_H))$ **then**
9. $D_H{}' := D_H \cup (d_t \wedge d_i)$
10. **end**
11. **end**
12. **end**
13. **if**$(D_H = \emptyset)$ **then**
14. $\forall d \in D_S$, set priority attribute $d \cdot s_p$ giving increased priority to devices with higher $d \cdot p_r$ values
15. **end**
16. **else**
17. $D_R := \emptyset, D_\delta := \emptyset, n := |D_H|$
18. **for** $t := 1 : n$
19. **for** $i := 1 : n$
20. **if** $((d_t, d_i \in D_H) \wedge (i \neq t) \wedge (d_t \cdot h_o = di \cdot h_o) \wedge (d_t, d_i \notin D_R))$ **then**
21. $D_R{}' := D_R \cup (d_t \wedge d_i)$
22. **end**
23. **end**
24. **end**
25. **if**$(D_R = \emptyset)$ **then**

26. $\forall d \in D_H$, set priority attribute $d \cdot s_p$ giving increased priority to devices (d) with lower $d \cdot h_o$ values

27. $\forall d \notin D_H \wedge d \in D_S$, reassign priorities for such devices accordingly

28. **end**

29. **else**

30. $D_\delta := D_H \setminus D_R$

31. **if** $(D_\delta \neq \emptyset)$

32. $\forall d \in D_\delta$, set priority attribute $d \cdot s_p$ giving increased priority to devices (d) with lower $d \cdot h_o$ values

33. **end**

34. $\forall d \in D_R$, set $d \cdot r_r = r(d)$

35. $\forall d \in D_R$, set priority attribute $d \cdot s_p$ giving increased priority to devices (d) with higher $d \cdot r_r$ values

36. **if** $\exists d \in D_R$, $d \cdot r_r$ attribute values are identical **then**

37. assign equal priority to such devices

38. **end**

39. $\forall d \notin D_R \wedge d \in D_S$, reassign priorities accordingly

40. **end**

41. **end**

42. **end**

43. **else if** $|D_S| \leq 1$ **then**

44. do nothing

45. **end**

46. **end**

47. $\forall d \in D_M$ reassign priority values for devices according to the c_r attribute values

2.4.1 Algorithm Execution

Figure 2.2 has been analysed to illustrate the prioritisation strategy and the individual attribute values calculated by the algorithm described in section 2.4. It is assumed that the file, mail and web servers are remotely accessible using the Remote Desktop Protocol (RDP) by administrator accounts, where currently those accounts are active on devices 'A', 'E' and 'B' as depicted in figure 2.3. Numerous (i.e. B-D, K-I etc.) other devices are also connected using this protocol as can be seen from figure 2.3. However it is assumed that a flaw has been identified in a common application of this protocol, the connectivity due to this is represented as 'Compromise 1' edges in the graph in figure 2.3, where the criticality level associated with a device is given in brackets. Another flaw allowing a Trojan Horse worm is then assumed to have been discovered affecting certain versions of both user and server operating systems; this is represented by the 'Compromise 2' connections (edges) in figure 2.3. If considered simultaneously the connectivity due to the aforementioned compromises is as depicted in figure 2.3, this is commonly the case with sophisticated worms which exploit numerous flaws or even if one decided to analyse a network for currently popular vulnerabilities.

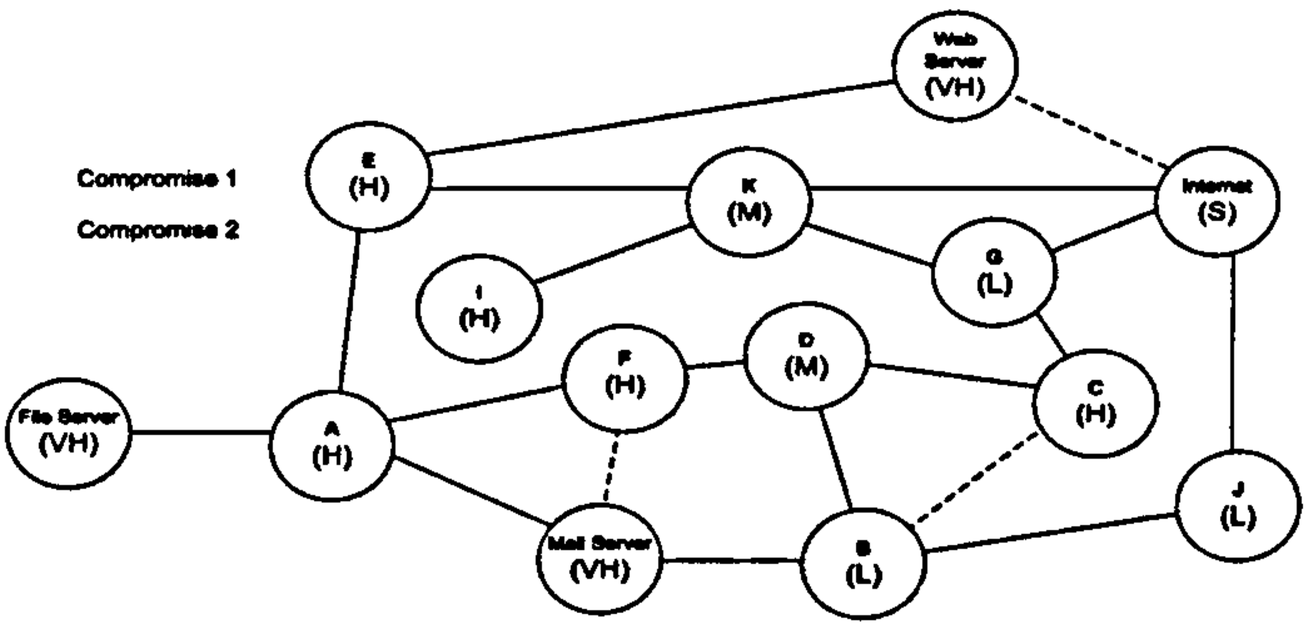

FIGURE 2.3: Compromise connectivity due to two vulnerabilities for the network given in figure 2.2.

The connections between devices B-Mail Server, F-D etc. depicted in figure 2.3 illustrate the fact that these devices are using the vulnerable version of the application running RDP. More importantly this indicates that such devices can communicate using the vulnerable application in question (i.e. firewall on device 'Mail Server' is configured to allow device 'B' to connect and utilise this service). Thus in the case of an attack if device 'B' is compromised it will be able to compromise the device 'Mail Server' even if 'Mail Server' has an embedded firewall enabled.

Table 2.2 details the values for each attribute of each device after the execution of the algorithm specified in section 2.4, when considering only 'Compromise 1'. Table 2.3 gives details when both compromises 'Compromise 1' and 'Compromise 2' are considered simultaneously.

Device	a_m	c_r	p_r	h_o	r_r	s_p
File Server	8	VH	5	4	0.0	3
Mail Server	7	VH	5	3	585.0	1
Web Server	3	VH	5	3	124.0	2
A	11	H	1	1	0.0	8
B	14	L	1	1	0.0	13
C	10	H	4	2	554.5	4
D	13	M	3	3	0.0	10
E	9	H	4	2	401.0	5
F	9	H	4	4	0.0	7
G	15	L	2	1	691.0	12
I	12	H	4	2	268.0	6
J	19	L	2	1	859.5	11
K	14	M	3	1	0.0	9

TABLE 2.2: Attribute values after execution when considering 'Compromise 1'.

Device	a_m	c_r	p_r	h_o	r_r	s_p
File Server	8	VH	5	4	0.0	3
Mail Server	7	VH	5	3	0.0	2
Web Server	3	VH	5	1	0.0	1
A	11	H	1	1	0.0	8
B	14	L	1	1	0.0	13
C	10	H	4	2	843.3	4
D	13	M	3	3	0.0	10
E	9	H	4	2	401.0	5
F	9	H	4	4	0.0	7
G	15	L	2	1	691.0	12
I	12	H	4	2	268.0	6
J	19	L	2	1	859.5	11
K	14	M	3	1	0.0	9

TABLE 2.3: Attribute values after execution when simultaneously considering 'Compromise 1' and 'Compromise 2'.

2.5 Modelling Overview

The algorithm described in section 2.4 has been developed in a C#.NET software version called **pfcca**. A high-level overview of the **pfcca** software is given in figure A.3. The **pfcca** software outputs prioritisation results in an Extensible Markup Language (XML) and .csv file which can be opened as a Microsoft Excel sheet. For input **pfcca** requires an XML file, which adheres to an XML Schema Definition describing devices and their associated compromise connectivity. However due to the time consuming nature of developing such XML input files by hand it was decided that automatic compromise connection network topology generation was required for preliminary testing. Therefore a network simulator named **generator** has been developed, this emulates the potential

for compromise connections between devices using a theoretical attack model based upon that of the Blaster [50] worm and a localised connectivity strategy[§] as described in [52]. A high-level overview of the **generator** software is given in figure A.4. Well known power-law strategies [53]-[56] for IP network connectivity form the basis of our network simulator, where a subset of this connectivity is chosen to represent the potential for exploitation (compromise connectivity) using the theoretical attack. In particular a set of nodes (devices) is distributed randomly in a plane, and an initial backbone of 5% of the devices is randomly connected. After which Waxman's [57] method is used to connect pairs of devices. This is used in popular network simulators such as BRITE [55], according to Waxman's technique the probability of connecting two devices u and v is as given in equation 2.5,

$$P(u,v) = \beta e^{-d/\alpha L} \tag{2.5}$$

where $0 < \alpha, \beta \leq 1$, d is the euclidean distance from device u to v, and L is the maximum distance between any two devices. In **generator** β=0.15 as in [55], and α is varied between 0.0000001 and 1.0. This method was chosen as it enables the modelling of high levels of random connectivity, which is likely to exist in future internet topologies. Such use of Waxman's method alone to model both power-law and random connectivity is in contrast to the technique used in [55], which has been found to provide limited modelling of random connectivity due to its underlying bias towards preferential association. From testing it has been found that lower values of α produce networks which follow power-laws giving hub style connectivity, which is representative of many client-server based applications. However higher values of α result in networks which exhibit more random connectivity globally with local pockets of power law connectivity. This is becoming increasingly apparent in today's networks due to peer-to-peer (P2P) software, where there is no clear client or even server [58]. The **generator** application randomly assigns a criticality level to each device in a network, assigning higher criticality levels to critical nodes such as machines with administrator privileges for servers. The time since a device last received anti-malware updating (a_m) is also randomly assigned by the **generator** software, where increasingly critical devices are likely to have a lower value representing a more up-to-date configuration. However all devices which are out of ones administrative control are assigned an a_m value of 24, this indicates the oldest possible anti-malware configuration and therefore a higher risk of connecting to such devices.

The theoretical attack emulated in the **generator** software is assumed to affect the majority of operating systems in use. This allows a remote attacker to gain unauthorised privileges using an open Transmission Control Protocol (TCP) port to automatically

[§]Much more likely to infect IP addresses close to its own address, which was a strategy employed by the Code Red II [51] worm.

replicate and even compromise the confidentiality and integrity of information on susceptible systems. It is believed that this is a realistically achievable attack[¶] due to the software monoculture which has resulted from the upsurge in use of COTS computing components. Such components have very similar if not identical fundamental structures and therefore limited defence-in-depth. For example the particular bug exploited in the Blaster worm affected default installations of Windows NT 4.0, Windows 2000, Windows XP as well as Windows 2003 Server. Other forms of malware based upon scripting languages such as Visual Basic (VB) and Java Server Pages (JSP) take advantage of popular functionality such as Microsoft Office and the Java Virtual Machine to propagate and cause a major impact. This was illustrated with the spread of the Melissa [59] macro virus, which used VB and various common Microsoft Office components to infect and spread.

In order to visualise the networks (graphs) produced by the **generator** software a **MATLAB M-file** can also created by the simulator, which contains the adjacency matrix of the graph (G). The **gplot** function in **MATLAB** is then used to draw the network and provide a visualisation of the networks described in the corresponding XML file.

2.6 Experiment

To assess the effectiveness of the strategy described in section 2.4, different network types ($0.0000001 \leq \alpha \leq 1.0$) have been tested in **pfcca** with each test consisting of 100 networks (created by **generator**) of random sizes (ranging from 500 to 1000 devices) and compromise connectivity. The aim of the test was to identify what percentage of devices in a network are successfully prioritised by the four attributes of: criticality (c_r), compromise path risk level (p_r), compromise path length (h_o) and residual risk (r_r). The results are given in table 2.4.

α	c_r (%)	p_r (%)	h_o (%)	r_r (%)	**Total** (%)
1×10^{-6}	0.51	0.51	0.17	16.09	17.29
1×10^{-5}	0.51	0.68	0.17	15.65	17.02
1×10^{-4}	0.52	1.55	0.69	29.91	32.66
1×10^{-3}	0.53	2.10	2.45	70.76	75.84
1×10^{-2}	0.52	2.09	2.44	70.49	75.54
1×10^{-1}	0.51	2.05	2.39	70.60	75.55
1×10^{0}	0.52	2.06	2.40	70.73	75.71

TABLE 2.4: Comparison of metric performance for prioritisation purposes.

[¶]Even in the presence of distributed firewalls as the attacker may still be able to attack a system indirectly through intermediate systems which may have trust relationships.

2.6.1 Experiment Analysis

From the results in table 2.4 it can be seen that as the randomness (α) of network connectivity increases so does the ability to prioritise using the strategy outlined in section 2.4. Approximately 75% of prioritisation decisions for networks with $\alpha \geq 1\times10^{-3}$ were based upon one of the four attributes chosen. This highlights the usefulness of the consolidated prioritisation technique for successfully prioritising devices in randomly connected networks of the future. However further improvements are required before any such approach can be used in real operations. On average the c_r attribute produced the least distinguishing results. This confirms the belief that criticality levels alone are not a suitable distinguishing metric for prioritisation in large networks (or where the number of devices is greater than the number of criticality levels) as discussed in section 2.2. Similarly the compromise path risk level (p_r) and compromise path length (h_o) have only distinguished between a small number of devices, resulting in the dominance of the residual risk factor (r_r). It is believed this is due to the client-server nature of many applications, where the compromise connectivity between many devices is similar. Therefore the residual risk factor could used as the first rather than the last distinguishing factor, which may save time in assessing for prioritisation.

2.7 Summary

It is believed that prioritisation of network security services is much needed in order to successfully protect critical IT assets, particularly as many networks are allowing more diverse and extended connectivity, which is resulting in increased security risks. It has been shown that security classifications alone, do not provide the required level of granularity upon which to base prioritisation decisions in large dynamic networks. Therefore a strategy has been developed based upon compromise path analysis to differentiate between devices, which are assigned to the same high-level static criticality level. The technique introduced here considers dynamic risks from connected devices and quantifies the highest risk posed to any one device. This builds upon previous work by QinetiQ [46] in the area of compromise path analysis, by enabling the prioritisation of security services for devices in a network, through metrics such as the distance between a victim and attacker (compromise path length), as well as inherent risks due to last anti-malware update time (residual risk factor).

There are two cases in which even a prioritisation technique would be of limited use, firstly if an attacker has complete knowledge of the prioritisation technique as well as the criticality levels assigned to individual devices then they may target specific devices, to cause maximum impact. Related to this is the fact that an attacker may employ traffic analysis techniques to identify in which order and to what degree individual devices (or

network segments) receive security services, thus potentially providing valuable information on the criticality of specific devices (or a group). Secondly it is acknowledged that although our prioritisation technique may limit the impact from malware such as CodeRed, Blaster and even Mydoom [60] it would have limited success in protecting against an extremely sophisticated theoretical attack known as a Flash worm, as described in [52]. A Flash worm is able to accurately pre-compute the network addresses of susceptible devices, enabling it to spread through the entire Internet in seconds.

From preliminary testing the dynamic risk analysis technique for prioritisation described in this chapter has delivered promising results, distinguishing between the majority (75%) of devices in networks of interest. Alternative metrics need to be identified to further improve on the results to date. If the technique is to be deployed comprehensively on large scale real-world networks then one would need to collate the necessary network topology semantics information and analyse[‖] this using a tool such as **Nmap** to identify potential compromise connectivity between devices. The development of an automated technique to achieve this is proposed as future work. A key benefit of the technique is that it can be run in parallel, which is likely to be required for any production system. However further work is required in order to ascertain the effectiveness of the technique, for example when considering criticality systems with differing levels. It is hoped that the $\dot{VP}$ can be used as a standard metric to compare the performance of differing techniques for the reduction of risk to critical assets in a network.

[‖]Firewall rules applied by individual devices and network edge filters at any one time.

Blank Page

Chapter 3

Distributed Security for Decentralised Information Sharing

The main objective of the work described in this chapter was to assess novel access control and authentication mechanisms, which may enable increasingly autonomous agent applications to operate in ubiquitous computing environments. Ubiquitous computing provides opportunities for a distributed and ad-hoc Command and Control (C2) and information sharing environment, but challenges traditional information security techniques, which must be adapted to ensure the best exploitation of the prospect. In particular machines must be able to authenticate and prove authorisation to securely access distributed services, in an agile and robust manner. Currently many ad-hoc computing applications use simple privilege mechanisms or human intuition to control access to potentially sensitive services, such as reconnaissance information. However such mechanisms lead to a rigid and centralised information sharing model which does not scale well, especially when considering future computing capabilities in which machines autonomously provide and consume information based services. Results from testing carried out into the perceived operational benefits of alternative (current centralised and future distributed) access control models for a Mobile Ad-hoc Network (MANET) are discussed. In the MANET individual nodes (or agents) represent mobile devices needing to share information in a decentralised and ad-hoc manner. A candidate architecture for realising a distributed access control mechanism incorporating context is then defined.

This chapter describes work that builds on research described in [65] by providing a quantitative assessment of alternative access control mechanisms for MANETs. Section 3.1 describes how the concept of autonomy could be useful in a ubiquitous computing setting, section 3.2 illustrates the need for alternatives to current centralised access

control mechanisms. The scenario and simulation used to undertake a cost-benefit analysis of using a decentralised as opposed to a centralised access control model is defined in section 3.3, with results and analysis given in section 3.4. Section 3.5 describes a candidate role-based context-dependent access control model which could enable secure decentralised information sharing. Conclusions and suggestions for future work are then given in section 3.6.

3.1 Autonomous Management of Ubiquitous Computing

As alluded to in chapter 2 ubiquitous computing is being driven by mobile devices, which can be used to collect, store, and transmit vast amounts of potentially sensitive data in an always-on and real-time manner. This is evident in projects within the SEAS DTC [61], Autonomous Learning Agents for Decentralised Data and Information Networks (ALADDIN) [62] and related work on decision making with Multi-Agent Systems (MAS) in wireless sensor networks [63], [64]. In all of this work autonomous learning agents* need to share information in a distributed and ad-hoc manner to efficiently achieve time-sensitive objectives. For example imagery sensor agents may be used to monitor and stream reconnaissance for law enforcement purposes. In such scenarios agents need to provide and consume sensitive information services in real-time, therefore it is imperative that information exchange is timely as well as secure.

So far autonomy has been defined and studied using two distinctive approaches in the autonomous agent's research community. Firstly it has been characterised as the degree of self-control an agent has over its own decisions, which is assigned by a higher-level authority, e.g. a supervisor (human or machine), this was discussed in [65] as decision autonomy. Decision autonomy is a satisfactory concept when considering agents in closed environments. However as the operational environment becomes more uncertain (real-life) then an additional understanding of autonomy with respect to self-capability is required as described in [66]; this was discussed in [65] as self-capability autonomy. This latter approach is an assessment of an agent's ability, to accomplish its assigned mission objectives with minimal external co-operative intervention.

The quest for increasing self-capability autonomy can be seen in the field of autonomic computing where research [67]-[70] is focusing on capabilities which can more effectively manage the complexities associated with the ubiquitous nature of modern networks. According to [71] the aim of autonomic computing is to increase the sophistication of individual components and networks, so that they can become "self-managing" and take corrective actions in accordance with overall system-management objectives. This is analogous to the human nervous system which controls bodily functions such as heart

*An agent may be human or machine.

rate, breathing and blood pressure without any conscious attention on our part.

The need for increasingly self-capable systems is partly driven by the cost of human labour, for example the expenditure on managing networks currently exceeds equipment costs by a ratio of up to 18:1 [72]. As well as excessive costs, manual control by human operators results in overly complex and inefficient processes. From a security perspective such complexity can lead to inadequate protection as well as inhibiting business requirements. This is evident in current electronic access control procedures which provide a coarse level of granularity, and are therefore reliant upon human intervention to securely enable the required level of access to services.

The mobile and plug-and-play nature of modern computing devices is facilitating real-time decentralised information sharing. One such technology is the MANET, which enables groups of agents to form ad-hoc coalitions capable of sharing information in a distributed manner. The operational benefits of MANETs are:

- *Fault-tolerance* - No single point of data processing or communications failure.
- *Scalability* - Not subject to communication or computational bottlenecks.
- *Flexibility* - Support on-line addition or deletion of nodes (i.e. plug-and-play).

It is envisaged that MANETs will be used to enable a wide range of applications, from search and rescue operations in the emergency services to the improvement of business processes in a corporate environment. In such scenarios the agents which make up a MANET need to constantly share sensitive information-based services [63], [64]. Therefore it is essential to control access to such services to preserve the security principle of least privileges and prevent information overload.

3.2 Centralised Access Control & Authentication for MANETs?

Access control is based upon the subject-privilege-object methodology, where a subject (agent) has an associated privilege (read, write etc.) with respect to some object (information service). In the majority of current ad-hoc computing applications which adhere to the subject-privilege-object methodology, numerous factors including a subjects identity are manually considered for authorisation purposes by a central 'security-aware' authority (i.e. humans). For example in many Home Land Security (HLS) systems drone agents such as SkySeer [73] are used to carry out surveillance operations, all interactions with such drones are executed via a central control station. For security purposes the control station is used as a human in the loop mechanism to manage the flow of all information to/from drone agents due to its potentially sensitive nature. However

this limits the possibility of forming a MANET based coalition between drone agents and other 'intelligent' agent types such as human patrol guards, where agents could directly interact in a P2P network, sharing information services in real-time. Such P2P coalitions require individual agents to locally (decentralised) enforce authentication and authorisation decisions. This is in-line with the concept of de-perimeterisation described in chapter 2, where individual components are required to secure local data providing much more fine-grained access control capabilities.

The alternative information exchange architectures due to centralised and decentralised access control are illustrated in figure's 3.1 and 3.2 respectively. Due to the largely remote-controlled operating mode of current agent technologies such as an Unmanned Air Vehicle (UAV), their information exchange requirements are limited to a centralised model, between the agent and its current control station as depicted in figure 3.1. In figure 3.1 the agents can be seen to have static and continuous communications with a control station. Such communications may be for telemetry, C2 and sensor feed purposes. However as agents become more self-capable their communications are likely to become less constrained, giving rise to more complex distributed interactions and potentially a swarm MANET.

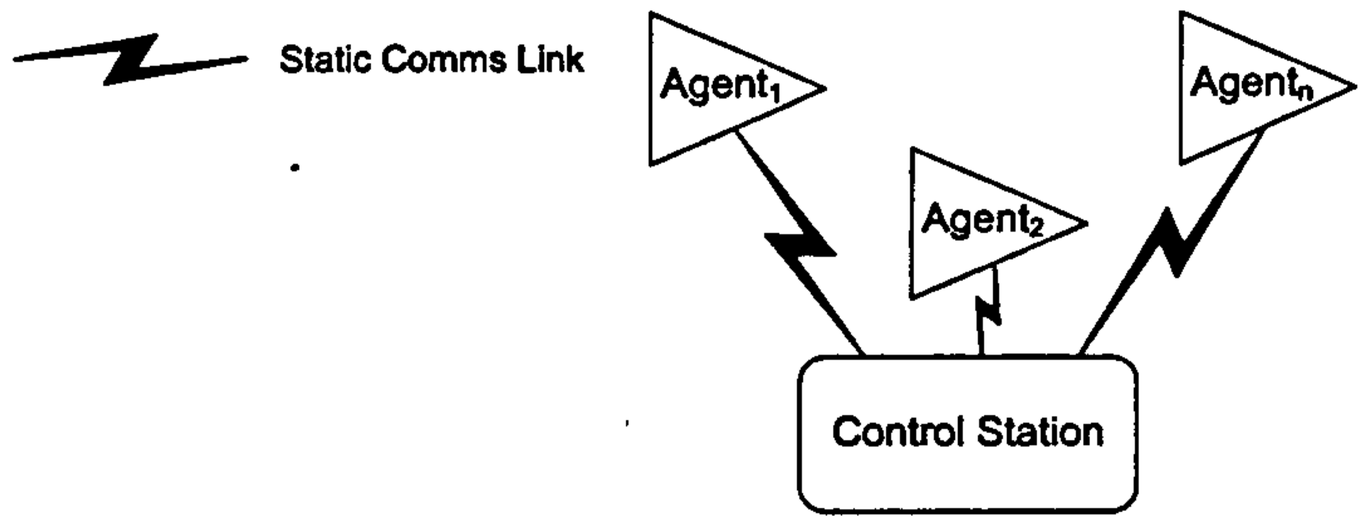

FIGURE 3.1: Current MANET Information Exchange Architecture.

A swarm-like MANET setup is illustrated in figure 3.2, where the information exchange requirements between agents and a control station are much more ephemeral. In this arrangement the individual agents will be capable of communicating amongst themselves enabling increasingly sophisticated Machine-to-Machine (M2M) interactions.

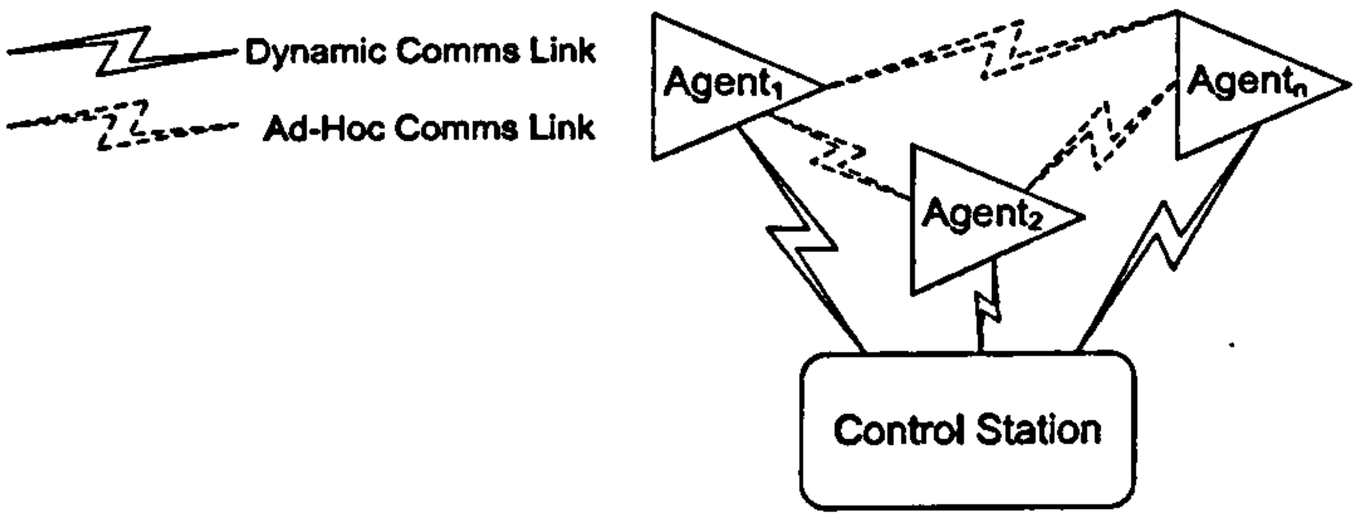

FIGURE 3.2: Envisaged MANET Information Exchange Architecture.

A key requirement of surveillance MANETs is Decentralised Data Fusion (DDF), where sensor agents need to correlate observations by sharing information in a distributed and ad-hoc fashion. Using centralised access control such sensor agents would have to fuse data through an authorised central agent(s), to ensure that only relevant data from those agents with common areas of observation is fused. In this situation the central authority can either fuse the data itself or pass on relevant information to individual sensors for subsequent fusion. However it is hypothesised that a centralised access control model will undermine the potential benefits (described in section 3.1) of a MANET by introducing unacceptable:

- Latencies - due to bottlenecks in information sharing.
- Levels of risk - due to a single point of failure

A simple sensor network is illustrated in figure 3.3 where three image sensors (S_1, S_2 and S_3) are observing an area of interest. Due to their locations S_1 and S_2 have an intersecting observation area (A_{1-2}) and therefore need to share information to correlate observations as do S_2 and S_3 for the area A_{2-3}.

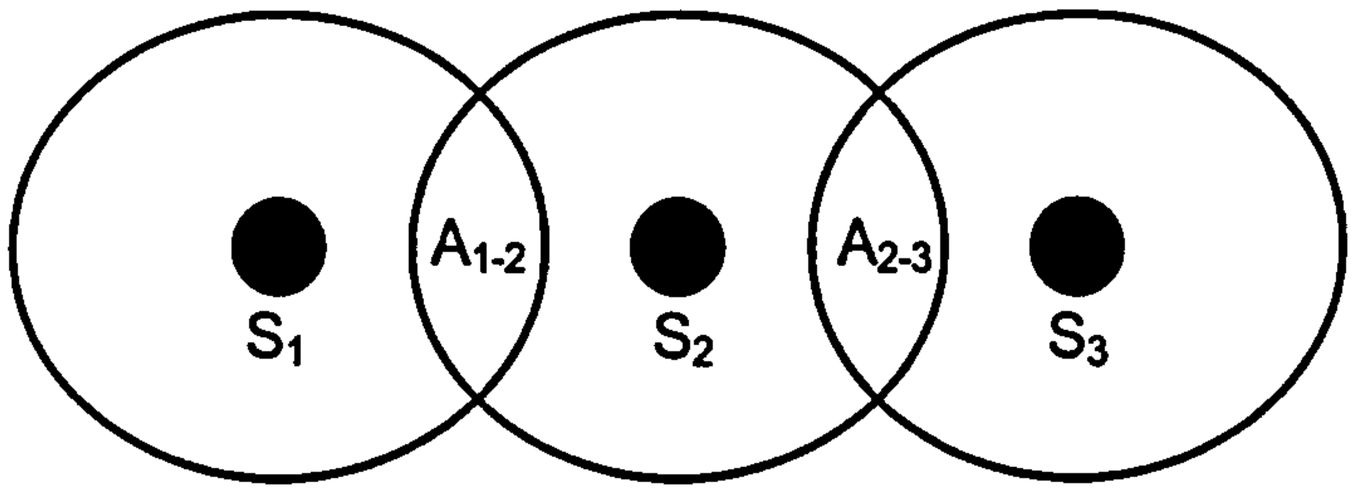

FIGURE 3.3: Sensors need to share information in a distributed and ad-hoc fashion.

3.3 Scenario and Simulation

In order to assess the relative merits of centralised and decentralised access control mechanisms a cost-benefits analysis has been undertaken through simulation. The simulation involves a MANET of multiple ground and air based autonomous agents in a HLS mission, as illustrated in figure 3.4. The objective of the mission is to carry out surveillance and tracking of target agents for their capture. For simplicity, surveillance and tracking activities are performed by air sensor agents and capture activities by ground effector agents. Therefore sensor agents (S_1 and S_2) need to share information to correlate observations on the target (T_1) and provide timely information to authorised effector agent(s) (E_1) regarding the location of a target(s). To represent the movement of sensors and capturers, as well as tasking by mission managers simple probabilistic decision-making models based upon previous modelling in NITEworks [74] are used.

NITEworks is a UK MoD programme which has developed battlespace models to assess future military capability requirements. In the simulation a centralised access control model is represented by a centralised information exchange architecture, and similarly a decentralised access control model is represented by a decentralised information exchange architecture.

The scenario has been implemented in Requirements Driven Development (RDD)-100 and utilises generic models developed previously by BAE Systems. RDD-100 is an object-oriented systems engineering tool which enables requirements definition, simulation and analysis of engineering problems. In the model all air-based surveillance and tracking agents are represented as *sensors*, similarly ground-based capture agents are represented as *effectors* with tracked agents modelled as *targets*. For the purposes of this exercise traditional control station or mission managers are represented as C2 agents.

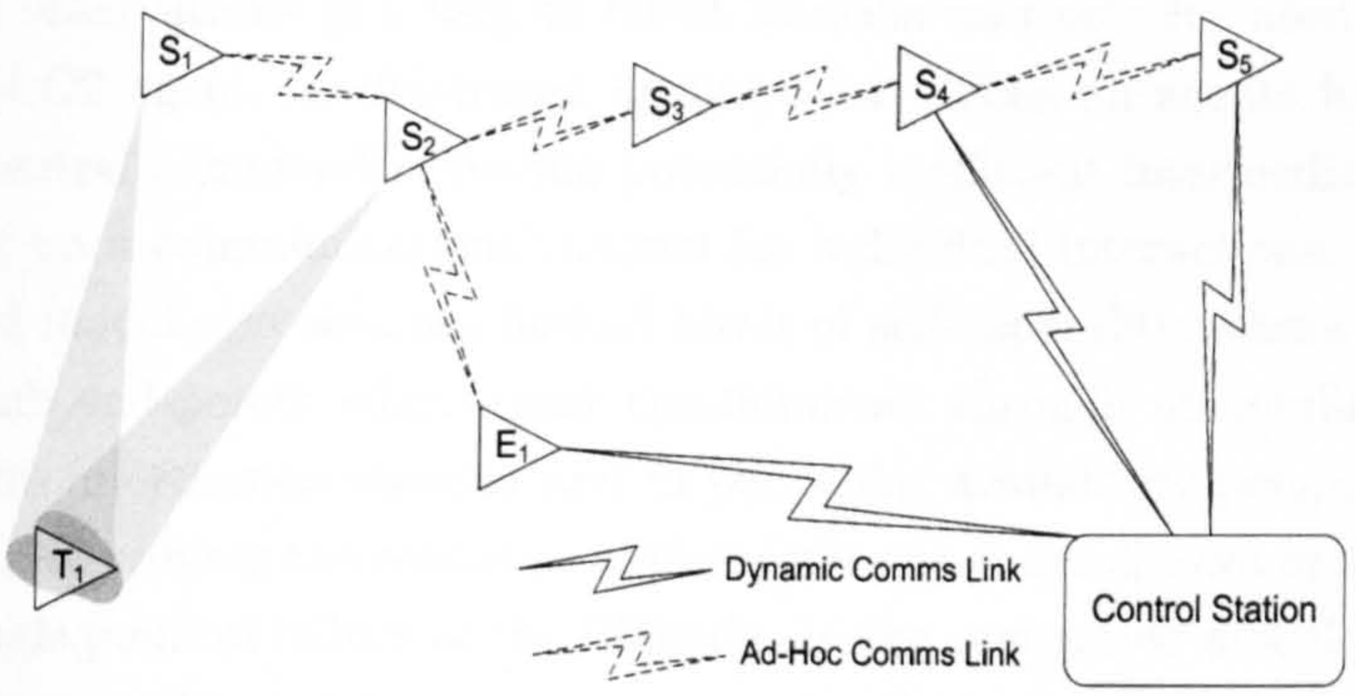

FIGURE 3.4: A decentralised information exchange architecture introduces security challenges.

3.3.1 Modelling

In the simulation sensor agents perform pre-defined formation movements until a target agent has been observed at which point they pass this information to the C2 agent after which a sensor agent may be chosen to track a target. At start up effector agents remain stationary in a randomly assigned location after which they may be assigned to a mission to capture targets. The movement of target agents is based upon a probabilistic mechanism, where all targets move according to a simple bounded random behavior model used in previous NITEworks studies.

This research has utilised fragments of previous NITEworks models, where the tasking of sensors and effectors to specific missions (targets) is allocated by C2 for both centralised and decentralised scenarios. The principal difference between the centralised and decentralised models used here is the way in which sensor observations are correlated

and reported to effectors. In the centralised model all sensor observations are correlated by C2, whereas in the decentralised model all sensor observations are autonomously (or locally) correlated by individual sensors before C2 is notified for mission tasking. In the centralised model target track information is reported by sensors to effectors via C2, however in the decentralised model target track information is reported by sensors directly to effectors. As a result the decentralised model incurs additional latency delays during the mission tasking phase. This is due to the need for a security credential negotiation task, where the C2 agent acts as a trusted third party providing credentials to enable sensors and effectors to mutually authenticate before sharing potentially sensitive information. These differences in the centralised and decentralised models are illustrated in the Use Case diagrams given in Appendix B.

In the case of a centralised access control model the information exchange architecture consists of all agents connected through a central C2 agent. Given this architecture any sensor observations of a targets latest location and velocity need to be communicated via the C2 agent, as illustrated in figure 3.1. Thus all agents have direct interaction with central command removing potentially inefficient intermediary processes, such as setting up a communications channel for individual interactions. However such a centralised model also assumes limited levels of self-capability, where all agents are treated as dumb end-points which must communicate through an intelligent central process. From an information security and in particular availability perspective, the centralised model also reduces the resilience of the system from breakdown or attack due to a potential single point of failure at the C2 node. In the centralised model all sensor observation correlation and target track messages receive the highest priority in the queuing system; thus minimising processing delays due to the C2 agent.

A decentralised access control model enables an ad-hoc and distributed information exchange architecture, where individual sensor and effector agents can securely interact and share observations on target agents, as illustrated in figure 3.2. This introduces the potential for a swarm-like MANET, which can meet high-level operational objectives with minimal external intervention. In the decentralised model sensors correlate observations locally, this could potentially reduce the load on the C2 node and improve the overall operational performance. All latest target track data is sent directly from the sensor to the effector in the decentralised model, this is unlike the centralised model in which all interactions between sensor and effector agents must go via C2. The decentralised model does have additional delays representing security credential negotiation between the C2 and sensor/effector agents for each mission assignment. In the model all sensor agents are assumed to operate under the command of one agent (C2-sensor) at all times, and all sensors are assumed to hold mutually trusted digital certificates (signed by their corresponding C2-sensor agent). However effectors are pooled from a wider range of resources, where an effectors digital certificate is signed by a different trusted

third party (C2-effector agent). Therefore when sensors and effectors are assigned to undertake a joint mission the tasking (C2-sensor) agent authenticates both parties. For the decentralised model an additional *security credential negotiation* step is required to allow sensors and effectors to directly communicate, in the security credential negotiation step the C2-sensor agent provides the effector with the necessary authorisation credentials to receive necessary targeting information from sensors. In the decentralised model the security credential negotiation step adds additional latency (approximately ten times that in the centralised model) to the delay between both sensors and effectors receiving tasking details to pursue a particular target for a given mission, which impacts on a sensors ability to feed target track data to an effector. Ultimately such delays influence the effectors ability to capture a target.

The aim of the simulation is therefore to quantify the benefits (or otherwise) of decentralised information exchange over the currently implemented centralised architecture. This is achieved by assessing the time taken from a targets initial identification to its capture (Total Capture Time), for different values of the Deliver Target Effect (DTE) and Emit Target Stimulus (ETS) tasks. The DTE task represents the final capture phase of a target by an effector, whilst ETS is the sensor observation refresh rate, which determines the freshness of target track data.

As described previously in the centralised model target tracks are reported by sensor to effector agents via C2, whereas in the decentralised model sensors report target tracks directly to effector agents. These differences are illustrated in figure 3.5 where sensor agents, sensor 1 and sensor 2 must correlate (information exchange between sensor 1 and sensor 2) and report (information exchange between sensor 1/sensor 2 and effector 1) all observed target track information via C2. In the decentralised model all sensor observation information may be correlated and reported directly without having to go through C2. To assess the impact of communications link latency on performance (Total Capture Time), three different ratios of sensor/effector to C2 and sensor to effector communications link latencies have been tested (1:1, 5:1 and 40:1). For the purposes of correlating and reporting sensor observations the centralised model only uses the sensor/effector to C2 link, whereas the decentralised model mainly uses the sensor to effector link.

In the model individual tasks and communications links are subject to latencies which could be due to any number of factors such as throughput, terrain and jamming. For the purposes of this model standard battlespace simulation practices from [74] have been adopted. This includes characteristics such as average latencies with corresponding probabilistic distributions associated with each agent type for the processing of individual tasks as well as parameter values such as those assigned to both DTE and ETS. This enables the accurate assessment of an effectors performance through the Total Capture Time metric. The model developed outputs log data in the form of .csv files which can be opened as Microsoft Excel sheets.

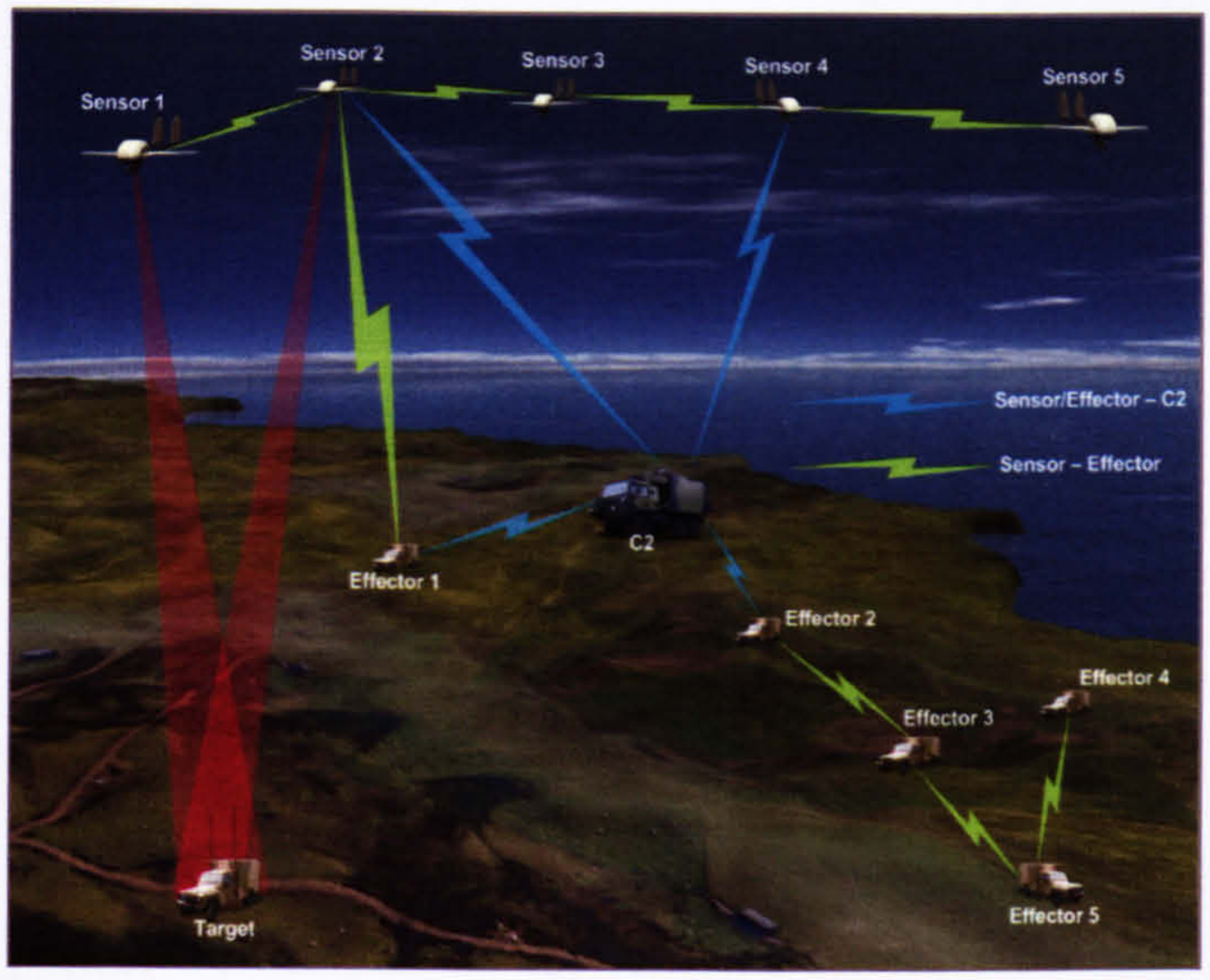

FIGURE 3.5: Scenario considered as part of modelling.

3.4 Results

As suggested previously the scenario was tested for a range of values (0.5 to 9.0 with increments of 0.5) for DTE and ETS, where each test was run for the equivalent of 4320 minutes (or 3 days). In tables's 3.1 and 3.2 Dec represents the decentralised network and Cen1, Cen2 and Cen3 represent centralised networks with communications latencies of 1:1, 5:1 and 40:1 respectively. All units in tables 3.1 and 3.2 are in minutes.

3.4.1 Analysis

The results given in tables 3.1 and 3.2 are illustrated in figure's 3.6 and 3.7.

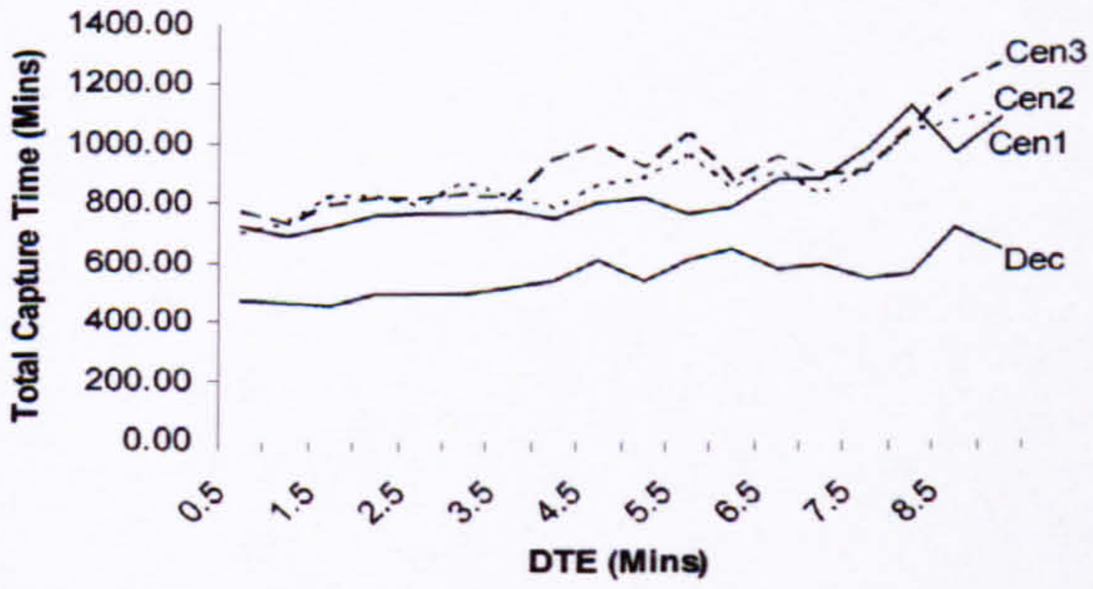

FIGURE 3.6: Average time taken to capture a target when DTE is varied for centralised and decentralised information sharing.

DTE	Dec	Cen1	Cen2	Cen3
0.5	473.14	723.28	700.86	771.57
1.0	462.75	695.01	734.60	733.56
1.5	453.43	725.74	827.56	794.17
2.0	494.48	766.32	824.44	820.72
2.5	494.33	774.52	791.88	820.89
3.0	498.85	770.35	870.38	834.05
3.5	520.04	781.19	822.68	818.50
4.0	541.95	756.47	783.57	950.88
4.5	612.35	812.07	863.41	996.99
5.0	544.27	829.27	891.63	926.84
5.5	616.58	774.63	964.68	1035.42
6.0	649.16	795.16	854.31	881.05
6.5	582.39	888.16	911.89	962.98
7.0	599.10	886.69	836.37	904.13
7.5	549.07	990.41	925.03	914.97
8.0	565.97	1133.72	1043.94	1054.73
8.5	723.32	975.53	1074.94	1201.08
9.0	649.16	1092.42	1108.73	1271.56
Average Difference		285.59	322.25	370.21

TABLE 3.1: Performance Comparison with variance in DTE.

ETS	Dec	Cen1	Cen2	Cen3
0.5	455.00	655.81	708.17	699.40
1.0	414.04	701.99	711.68	719.56
1.5	518.47	768.65	741.65	832.72
2.0	462.35	717.95	788.59	805.24
2.5	581.28	771.64	748.19	786.17
3.0	593.19	744.75	711.62	783.62
3.5	571.69	797.89	834.32	796.69
4.0	514.53	834.62	837.82	805.46
4.5	640.33	738.08	901.73	780.47
5.0	584.97	795.94	837.81	806.05
5.5	514.15	839.07	845.36	809.39
6.0	611.54	843.06	883.29	864.93
6.5	616.27	873.91	924.33	988.95
7.0	609.62	829.26	868.08	880.71
7.5	697.13	972.83	870.03	949.75
8.0	636.61	883.64	985.93	898.10
8.5	666.75	924.37	922.48	944.69
9.0	735.11	1062.97	967.74	1061.06
Average Difference		240.75	259.21	266.11

TABLE 3.2: Performance Comparison with variance in ETS.

From figures 3.6 and 3.7 it can be seen that there is a steady increase in the Total Capture Time (performance) as both the DTE and ETS parameter values increase.

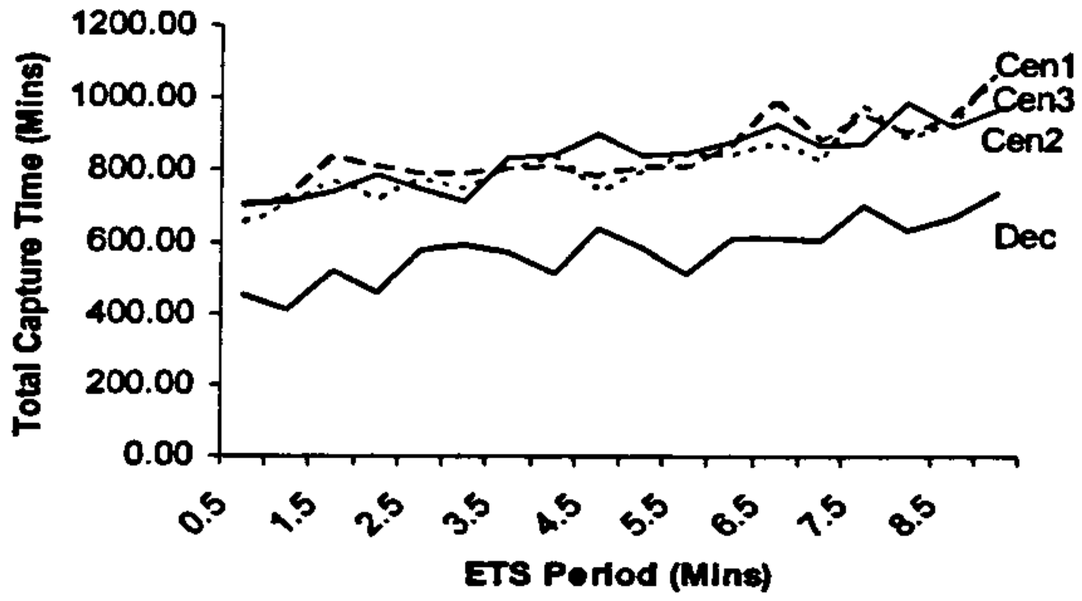

FIGURE 3.7: Average time taken to capture a target when ETS is varied for centralised and decentralised information sharing.

Even when the latencies of sensor/effector to C2 and sensor to effector communications links are equal, the decentralised model (Dec) provides superior performance, enabling more timely capture of target agents. This is in the presence of an extremely high average delay for mission tasking in the decentralised model due to security credential negotiation. Although the performance of the centralised model can be seen to improve with better communications links between C2 and sensor/effector agents, there is still a significant difference in average performance between centralised and decentralised solutions. The average performance difference is calculated as follows,

$$Average\ Difference = \frac{\sum_{x=0.5}^{9.0} \sqrt{(Dec_x - Cen_x)^2}}{18} \tag{3.1}$$

where Dec_x and Cen_x give the performance values for a decentralised and centralised test and x represents ETS or DTE. The sum of all such differences is then divided by 18 which is the total number of tests, thus giving the average.

For both parameters DTE and ETS the best performing (minimum average difference) centralised solution is Cen1, however even this, results in significant differences (ranging from ~40% to ~60%) in performance between Dec and Cen1 for both DTE and ETS. It is believed this is due primarily to the delays in the sensor correlation task, which in the centralised model is reliant upon C2 to coordinate and correlate observations in contrast to the decentralised model, where sensor agents correlate observations autonomously. Underlying this reduced performance in the centralised models is the bottleneck in queuing and therefore processing of target observations, which occurs at the C2 agent resulting in a decay of its ability to execute operations.

In this experiment the shared data (i.e. target location tracks) was both small and fixed in size. However future work may consider data of larger and differing sizes such as image or video files of a target to confirm capture. In such cases the performance of

the decentralised model may decrease if the bandwidth of the sensor to effector link is less than the sensor/effector to C2 link.

3.5 Candidate Access Control architecture

In order to achieve the potential benefits of decentralised information sharing outlined in section 3.4 individual agents must be capable of making authorisation decisions; therefore an alternative to the current centralised electronic access control model is proposed. Traditionally electronic access control privileges and therefore decisions have been based upon identity alone, where a list of agent identities authorised to access a service is predefined, the most prevalent access control model is the ACL as described in chapter 1.

In a MANET where M2M interaction is likely to be prevalent, access control based upon identity alone is likely to inappropriately reflect an agent's need and authorisation to access services. For example to form an ad-hoc coalition two or more agents may need to securely share sensitive services such as location information in a decentralised manner, in such scenarios authorisations for specific agents with respect to specific information services may evolve. Such authorisations may be dependent upon other attributes as well as identity. In particular the temporal and geographic location of an agent as well as available bandwidth may be used to more accurately control information exchanges. Therefore contextual factors may be used as additional constraints, enabling fine-grained decentralised access control. The potential for this form of context-aware computing (or location-based services) [75]-[77] to be used as an access control mechanism has also been identified in, [78]-[81].

3.5.1 RBAC

The proposed security architecture is based upon Role-Based Access Control (RBAC) [82], due to RBACs efficiency and effectiveness in securing large-scale and distributed computing applications as described in [83]. RBAC improves traditional Discretionary (DAC) and Mandatory Access Control (MAC) policies. In both DAC and MAC unique agent/privilege pairs have to be managed for each service; however in RBAC agents are assigned to roles where roles have associated privileges this maps naturally to organisational structures. More specifically it can be shown [83] that the cost of administrating RBAC is proportional to A+P, whereas the cost of assigning agents directly to privileges is proportional to A*P, where A is the number of agents in a role and P is the number of privileges required by the role. In the majority of organisations agent/role assignments change more frequently than role/privilege assignments. Therefore RBAC reduces administrative costs and complexity compared to traditional DAC and MAC policies in which agent/privilege assignments are direct and unique.

In [84] Feinstein et al. define three different RBAC models which build upon each other:

- $RBAC_0$ is the base model where agents are assigned to roles and privileges are assigned to roles.
- $RBAC_1$ is the intermediate model which incorporates and builds upon $RBAC_0$ by introducing the concept of role hierarchies, where super-roles implicitly inherit the privileges assigned to sub-roles.
- $RBAC_2$ is the most advanced model which incorporates and builds upon $RBAC_1$ by introducing constraints on agent/role and role/privilege assignments, therefore enforcing separation of duties.

3.5.2 Context-Dependent RBAC

In previous RBAC models [82] privileges have consisted of operations such as read and write with respect to some object, whilst work in the context-aware access control field has focused on adjusting the privileges assigned to an agent based upon contextual factors. An example of such context-aware access control is Zhang and Parashar's work [78] on Dynamic Context-Aware Role-Based Access Control (DRBAC). The DRBAC model extends traditional RBAC [82], by dynamically adjusting roles and privileges associated with agents according to contextual factors such as an agent's current location. However this introduces significant overhead as well as complexity such as the possibility for conflicts [78] between an agents current and previously assigned privileges. Therefore it is proposed to use traditional $RBAC_1$ as introduced by Feinstein et al. [83] to administrate and not adjust context-dependant privileges, where the context-dependant privilege builds upon traditional privileges by incorporating time and location as additional constraints of the operation.

Context-dependent privileges incorporating temporal and geographic constraints were introduced in [65] as an Authorisation State (AS), the proposed incorporation of this into the overall RBAC model is illustrated in figure 3.8. An AS specifies the authorised location, time and operation for an agent to access a service. In a DDF application this could enable sensor agents to share situational awareness, for authorised geographic areas and times without compromising the security principle of least privileges and removing the need for authorisation via a central control station. It is argued this is viable because in an organisational environment (e.g. CVF, SEAS DTC and IFPA) it is possible to predict the services required by agents fulfilling specific roles, therefore ASs can be pre-assigned to different roles.

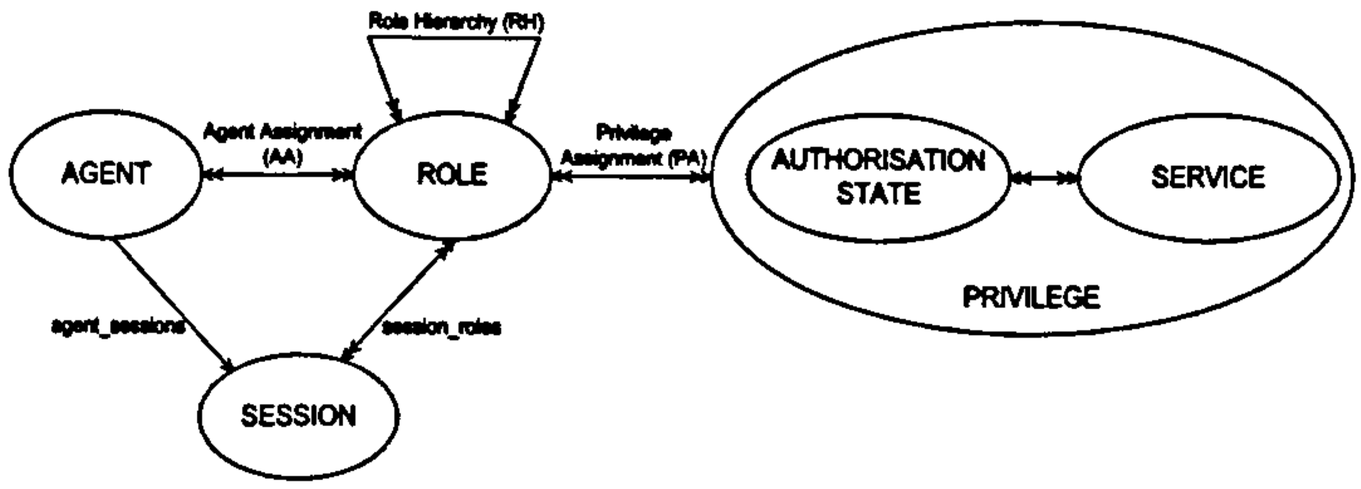

FIGURE 3.8: Introduction of AS mechanism into the core RBAC model.

3.5.3 Achieving Context-Dependent RBAC

As described previously an AS specifies the temporal, geographic and operational (read, write etc.) attribute constraints which are assigned to a given privilege, this is based upon the Attribute Certificate (AC) introduced by Farell and Housley in [85]. An AC is a digital certificate based mechanism which can be used for both authorisation as well as authentication purposes. A separate AS is assigned to an agent for every service the agent is authorised to access, where RBAC enables the efficient assignment of AS's to agents. In our architecture it is proposed to use the EureCA certificate to implement the concept of an AS. EureCA certificates were introduced in the Mobile Workers' Secure Business Applications in Ubiquitous Environments (MOSQUITO) programme [86] as a combined authentication and authorisation credential. The EureCA certificate uses an XML schema to describe a Public Key Infrastructure (PKI) based certificate which must be digitally signed by an authority trusted by the service provider, this could be used as the basis of federated identity techniques such as Security Assertion Markup Language (SAML). An example AS format is given in figure 3.9, where it can be seen that as well as including the public key for authentication of identity the AS also includes the temporal and geographic constraints within which an agent may access a specified service.

To date the literature on context-aware access control [78], [81], [86] has focused on credential pull type systems with centralised authorisation. In credential pull systems the service provider (policy enforcement point) requests necessary security credentials (e.g. username/password) from the client and passes these on to a third party (policy decision point), which makes an authorisation decision based upon the security credentials of the client and applicable policies. Similarly centralised authorisation requires the service provider to refer a client's request for service to a remote decision point, which consults a central policy store. Both credential pull and centralised authorisation based systems assume significant infrastructure. However in ad-hoc environments such as those envisaged and considered in the SEAS DTC [61] and ALADDIN [62] projects, MANETs will need to operate with high degrees of autonomy with minimal infrastructure deployments. Therefore a credential push type access control system has been chosen, which enables local (or distributed) authorisation decisions to be taken by service providers. Using the

```
<Issuer>
        <SignerOverview>
                <Name>Authority xyz</Name>
                        <PublicKey>
                                <!-- content deleted ... -->
                        </PublicKey>
        </SignerOverview>
</Issuer>
<Holder>
        <SignerOverview>
                <Name>Device abc X509 certificate</Name>
                        <PublicKey>
                                <!-- content deleted ... -->
                        </PublicKey>
        </SignerOverview>
</Holder>
<Service>
        <Identity>Target abc location</Identity>
</Service>
<Attributes>
        <Attribute>
            <Name>Temporal Validity</Name>
                <value>
                        <TemporalLocation>
                                <ValidFrom>2007-01-26T16:19</ValidFrom>
                                <ValidTo>2007-08-26T18:19:19</ValidTo>
                        </TemporalLocation>
                </value>
      </Attribute>
      <Attribute>
            <Name>Geographic Validity</Name>
                <value>
                        <GeographicLocation>
                                <!-- content deleted ... -->
                        </ GeographicLocation >
                </value>
      </Attribute>
</Attributes>
```

FIGURE 3.9: Example AS certificate.

push based system the client provides all the necessary credentials (EureCA certificates) to the service provider, the service provider can then make local authorisation decisions.

The use of a credential push based system with decentralised authorisation, improves the autonomy (or self-management) of a MANET by reducing the reliance of individual agents' upon third parties for access control decisions and therefore information sharing. This means when an agent receives a request for service it can locally process this assuming the consumer provides the necessary credentials (e.g. digitally signed certificate from a mutually trusted party). A high level illustration of the differences between decentralised push and centralised pull access control systems is given in figure 3.10. Actions 2a and 2b are additional requirements of the centralised model as it needs to acquire the credentials (for authorisation) of a client after authentication. A benefit of the centralised model is that policies can be implemented without changes to the client and server; however of more importance in this instance is that the centralised model

incurs additional performance costs due to the servers need to pull credential information in real-time. To overcome the compromise of an agent(s) it is proposed to use the periodic dissemination of a black list, although a strategy to detect compromise has not yet been derived. Additional protection is provided through the use of ephemeral digital certificates.

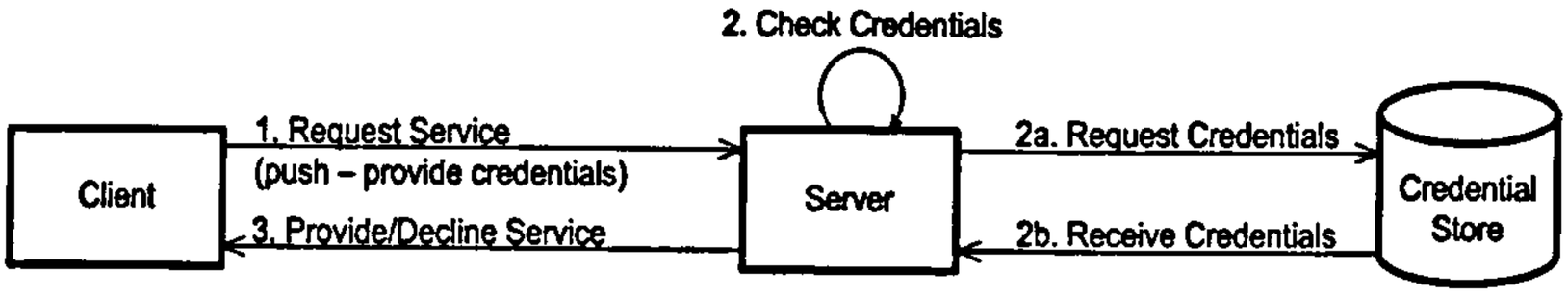

FIGURE 3.10: Decentralised and centralised authorisation for credential push/pull.

Traditionally digital certificates have been used to authenticate an identity, by proving that a specified public key belongs to the claimant. This is illustrated in figure 3.9 through the presence of elements issuer and holder. In AS's this concept is expanded by including the attributes and service elements, where the attributes describe the constraints within which a given agent may access a service(s).

3.5.4 Specification of Context-Dependent RBAC

The data elements which make up the proposed context-dependent access control model and the relationships between these are illustrated in figure 3.8. This is based upon the RBAC model presented in [82]. It can be seen that an agent $a \in AGENT$ is assigned to one or more roles $r \in ROLE$ and a role can be assigned to one or more agents. A session $s \in SESSION$ is a mapping between an agent and an activated subset of roles that are assigned to that agent at a given point in time. An active role is associated with one or more privileges $p \in PRIVILEGE$, where p is an approval for an agent to exercise an authorisation state on one or more services $se \in SERVICE$. An authorisation state $as \in AUTHORISATION\ STATE$ specifies the temporal ($t \in TIME$), geographic ($l \in LOCATION$) and operational ($o \in OPERATION$) constraints (read, write etc.) which are associated with a given privilege with respect to a specific information service *se*. Abstractly many authorisation states can be associated with many services and similarly many services can be associated with many authorisation states.

Figure 3.8 also illustrates *Agent Assignment* (AA), *Privilege Assignment* (PA) and *Role Hierarchy* (RH) relations; these relations are fundamental components of RBAC. There are also two primary functions which relate to sessions, where each session is associated with a single agent and each agent is associated with one or more sessions. The function *session_roles* represents the roles activated by the session and the function *agent_sessions* represents the set of sessions that are associated with an agent. The privileges available to an agent are the privileges assigned to the roles that are activated

across all of an agents sessions. The proposed access control model can be summarised as follows:

- *AGENT*, *ROLE*, *SESSION*, *AUTHORISTION STATE*, *PRIVILEGE* and *SERVICE*.
- $AR \subseteq AGENT \times ROLE$, a many-to-many mapping agent-to-role assignment relation.
- $AS \subseteq TIME \times LOCATION \times OPERATION$.
- $assigned\_agents(r) = a \in AGENT \mid (a,r) \in AR$, a mapping of role r onto a set of agents.
- $PRIVILEGE$: $AUTHORISATION\ STATE \leftrightarrow SERVICE$, the set of privileges.
- $PA \subseteq PRIVILEGE \times ROLE$, a many-to-many mapping privilege-to-role assignment relation.
- $assigned\_privileges\ (r) = p \in PRIVILEGE \mid (p,r) \in PA$, mapping of role r onto a set of privileges.
- $(p{:}\ PRIVILEGE) \rightarrow (as,se) \subseteq PRIVILEGE$, the privilege-to-authorisation state mapping, which gives the authorisation state and service associated with privilege p.
- $agent\_sessions\ (a{:}\ AGENT) \rightarrow S \subseteq SESSION$, mapping of agent a onto a set of sessions.
- $session\_roles\ (s \in S) = r \in ROLE \mid (agent\_sessions(a),r) \in AR$, mapping of session s onto a set of roles.
- $\bigcup_{r \in session\_roles} assigned\_priviliges(r)$, the privileges available to an agent in a session.
- $RH \subseteq ROLE \times ROLE$ is a partial order on $ROLE$ called the inheritance relation, written as $\geq$ where $r_1 \geq r_2$ only if all privileges of r_2 are also privileges of r_1, and all agents of r_1 are also agents of r_2. This is reflected in the fact that agents higher up the organisation hierarchy have the privileges to access information available to subordinates.

In the definitions above, AA defines the relationship between agents and roles; PA defines the relationship between privileges (including authorisation states) and roles. RH defines the inheritance relationship between roles and implicitly privileges. The principal difference between the context-dependent RBAC model described here and that of the original RBAC model in [82] is the use of an AS instead of a basic operation (read, write, execute etc.). This enables more fine grained electronic access control.

3.6 Summary

Ubiquitous computing is being facilitated by concepts such as MANETs, in which a group of mobile devices (or agents) collaborate by interacting and sharing real-time services in a distributed and ad-hoc fashion. In order to enable this, individual agents must be capable of automatically making access control and authentication decisions in a local and decentralised manner, in accordance with the concept of de-perimeterisation. This is in contrast to current practices which require centralised human based mechanisms, to control the flow of information to and from individual agents. Through modelling, this work has highlighted the operational benefits of decentralised over centralised access control for a HLS scenario, in which a MANET consisting of a collaborating group of autonomous agents need to share information to capture targets as soon as possible.

From the modelling it has been found that decentralised information sharing enabled by an appropriate access control model, performs significantly better than corresponding centralised information sharing techniques due to a communications and processing bottleneck at C2 in the centralised model. This is true even when expensive minimal latency communications links are employed between agents in the centralised model and the decentralised model is penalised by the introduction of delays in information sharing due to security credential negotiation. Future modelling is suggested to assess the breakpoints in performance when data of larger and differing sizes such as image or video files of a target are shared.

A candidate context-dependent RBAC model has also been proposed to enable decentralised information sharing. Experimentation and evaluation of this proposal in an analogous fashion to the experiment described in this chapter is needed. In the future the proposed model may be implemented in the Configurable Systems Engineering Research Tool (ConSERT)[†], which is an experimental MANET developed by BAE Systems.

In this research a distinction between context-dependent and context-aware access control has been made, where context-aware access control policies require real-time verification of contextual attributes, such as an agent's location. Therefore context-dependent RBAC has been offered as a lightweight alternative, where context-dependant RBAC builds upon the original RBAC model [82] by introducing the concept of a context-dependent privilege, which considers contextual factors to provide more fine-grained electronic access control. In reality it is likely that a mixture of both context-dependent and context-aware privileges will be used in future context-based access control models. In order to deliver such an access control model much work still needs to be undertaken not least from an information semantics point-of-view, which will be required to automatically filter data using the novel access control methods described here.

[†]Consists of a number of autonomous agents, including an airship and ground vehicles.

Chapter 4

XML-Based Validation for Sensitive Information Services

The requirement for real-time decision-making and delivery of effects is increasingly evident in a wide variety of commercial enterprise operations. This is also true in the military where for centuries the aim has been to out manoeuvre the enemy by making better decisions, quicker or having a higher operational tempo. In this chapter a tactical battlefield scenario is considered, where the IFPA project [87] is used as case study. Section 4.1 highlights the increasing need for timely information exchange in critical operations, section 4.2 introduces the *proliferation problem*, whilst section 4.3 describes how the proliferation problem is occurring due to limitations in current information security practices such as physically partitioned (air gap) infrastructures. Section 4.4 discusses the merits and drawbacks of relevant work on non-partitioned infrastructures, the need for logical infrastructure partitioning is introduced in section 4.5 motivated by an example usage scenario and XML as a candidate delivery technology. A summary and potential further work is then given in section 4.6.

4.1 Need for Timely Information Sharing

Much effort is being applied to advance procedural and technological systems in order to leverage the potential advantages of NEC as defined by the UK MoD in [1]. The concept of NEC is described as a transformer of military operations, the underlying theme of which is to provide the UK military organisation with decision superiority through information superiority. A vital tenet of information superiority is information sharing. According to [1] information sharing in the NEC sense implies rapid and simultaneous information transfer through *automated* processes. All four of these pillars are inherently reliant upon information security as illustrated in figure 4.1. Many current and future military information system developments are driven by the pillars of NEC. If

fulfilled, these pillars will provide agile and dynamic operational processes, based upon an efficient and effective electronic infrastructure.

From an information security point of view, network enabled systems still require the confidentiality and integrity of information to be preserved. However in the context of NEC the third information security aspect of availability is sharply focused on the need for timeliness, due to the need to improve operational tempo by fully integrating information systems. Timeliness implies that all relevant information is delivered to authorised agents at the required time.

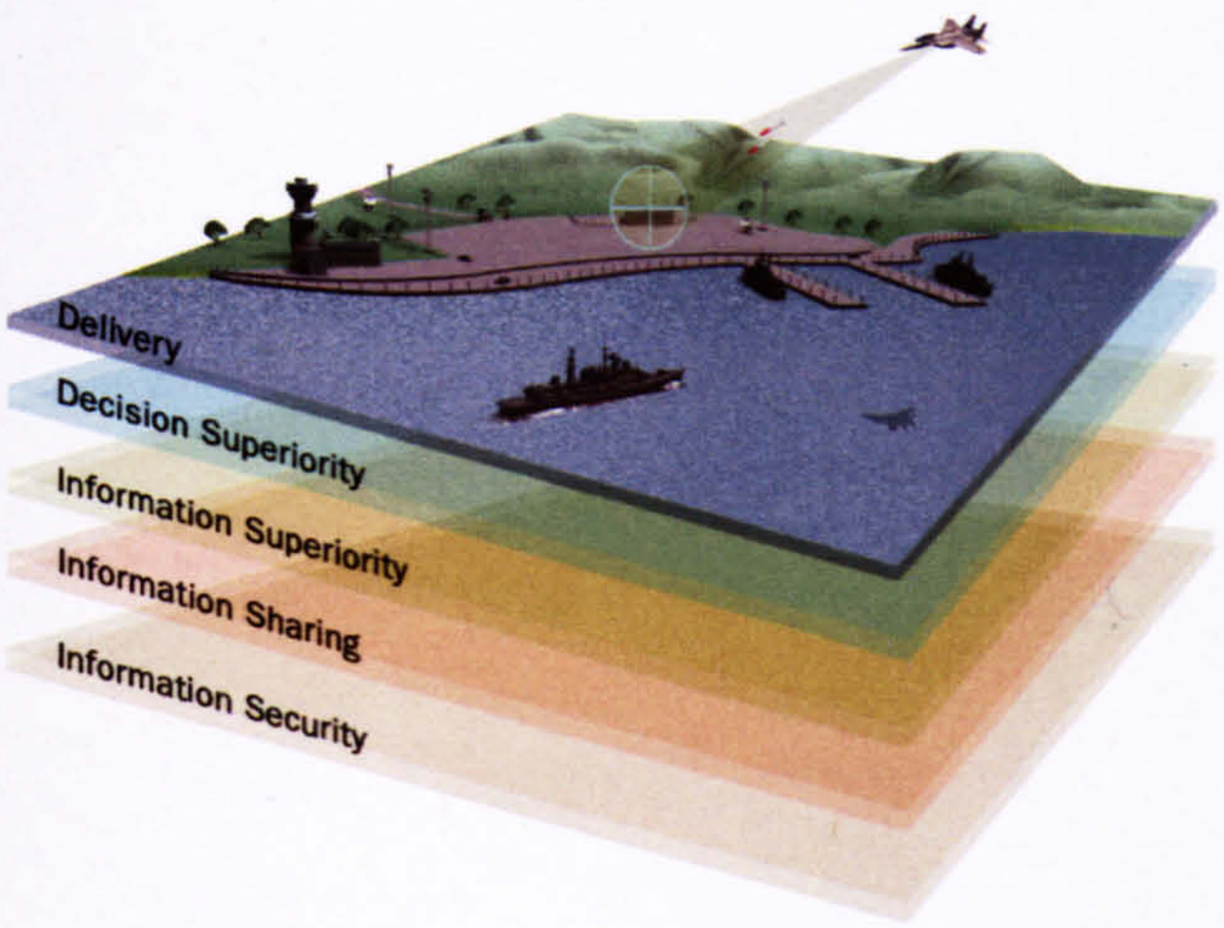

FIGURE 4.1: Fundamental building blocks required to achieve NEC.

4.2 Proliferation Problem

There is currently a significant increase in the utilisation of COTS products in military information systems; this is evident in a number of projects [87], [88]. As well as this the requirement for high-levels of interoperability with foreign nation systems, has lead to increased fragmentation* of the traditional military information system. In this context this is called the *proliferation problem*, where vulnerable connections are no longer confined to communications interfaces between multi-level classification systems. This is illustrated in figure's 4.2 and 4.3, where it can be seen that the risks to security have traditionally related to inter-classification communications such as between 'TS' and 'S' systems; however the problem domain has now augmented to an intra-classification level, where security partitioning is required between systems at the same level such as between 'S' and 'S' (or 'TS' and 'TS') systems.

*Due to the varying degrees of trust one may have in such systems.

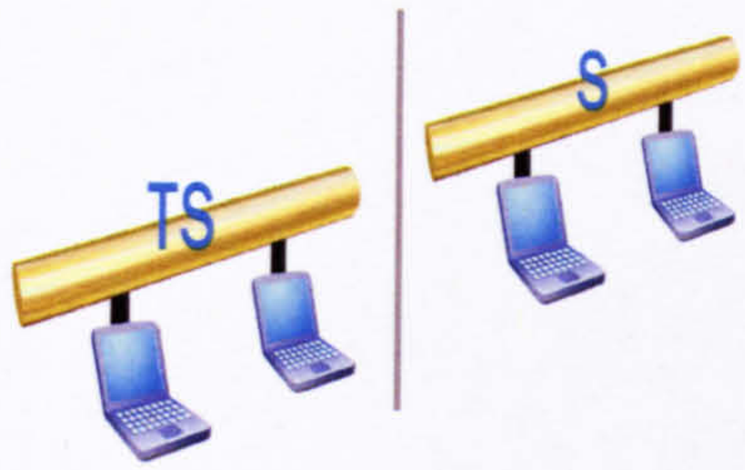

FIGURE 4.2: Highlighting the traditional security issue of interfacing at the inter-classification level.

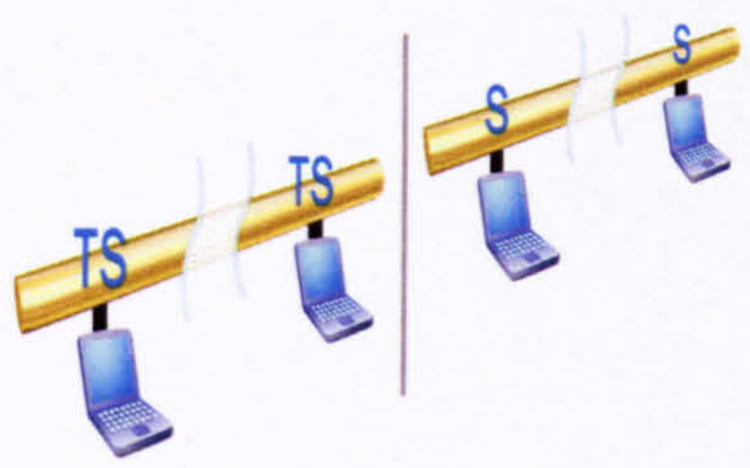

FIGURE 4.3: Highlighting the new security issue of interfacing at the intra as well as inter-classification level.

4.3 Physically Partitioned Infrastructures

It is currently standard practice to satisfy all high-risk connections whether inter- or intra-classification with a physically partitioned infrastructure (or air-gap) solution. The attractiveness lies in its provably simple and inexpensive approach towards managing risks and therefore gaining security accreditation. Air-gaps, however, impose severe restrictions on operational efficiency, therefore impacting upon the information security requirement for timeliness.

Significant advances in providing efficient alternatives to air-gaps have been made ever since the requirement for Multi-Level Security (MLS) was formally identified in the Orange Book [89]. The aim of MLS is to provide integrated and secure information systems. However, the overheads of implementing and maintaining such systems in accordance with common criteria requirements has proved to be very expensive for the military, which is a relatively niche market. This has led to the continued use of air-gaps.

The key NEC requirement for rapid and simultaneous information sharing in combination with the proliferation problem, brings into question the viability of the continued use of physically partitioned infrastructures. Clearly if increasing numbers of high-risk interfaces are introduced then an alternative to the partitioned infrastructure is required, to uphold the requirement for efficient as well as secure communications. Limitations [90] imposed by an air-gap solution such as manual medium transfer and re-typing include:

- Allows only primitive data transfers.
- Error prone due to human deficiencies.
- Slow information sharing.

From these limitations it can be seen that physically partitioned infrastructures prohibit efficient information sharing, therefore disabling the vision of true NEC. This is particularly true if vast numbers of such a mechanism are employed in time sensitive environments. From a security standpoint it has long been accepted that a system *is only as strong as its weakest link*, a direct analogy to this from an operational tempo point-of-view is that a system *is only as quick as its slowest link*. Therefore, partitioned infrastructures could prove to be a bottleneck in the information and decision flow chain, effectively limiting the ability of the UK military to fully leverage the potential of hi-speed and hi-tech systems architectures currently under development. These include applications such as IFPA [87] and Watchkeeper [91] as well as infrastructure capabilities such as Falcon [92] and Skynet V [93].

An instance of the proliferation problem has been identified in a number of projects [87], [88], [95] which require real-time push communications. As described previously the proliferation problem introduces increased fragmentation at the intra-classification level. This is illustrated in figure 4.4, where it is shown that numerous air-gaps may be utilised to separate systems and subsequently sub-systems of differing levels of trust. Such fragmentation can lead to serious operational inefficiencies if a physically partitioned infrastructure is used as the secure communications interface. In [95] an experiment was carried out where it was shown that manual information transfer introduces administrative overheads into the kill chain (F2T2EA) from an air force point-of-view, where the kill chain is considered the time between which a potential target is first identified to the time at which it is effected.

Inefficiencies in the kill chain would limit the ability of systems such as IFPA to engage high value Time Sensitive Targets (TSTs), which is a fundamental operational requirement. TSTs can be seen as those targets which are only active[†] for a limited[‡] period of time, this is exemplified in the current conflict in Iraq, where it can be seen that enemy targets are no longer in the form of large relatively static forces such as the Red Army (former USSR army) or Republican Guard (former Iraqi elite forces). Instead highly dynamic and agile terrorist cells are a new kind of enemy, the invisible enemy who if located must be effected as soon as possible. However this requires minimal latency from the moment that a potential target has first been identified to all other stages in the kill chain.

[†]Available to effect such as sense.

[‡]This period of time is becoming increasingly constrained.

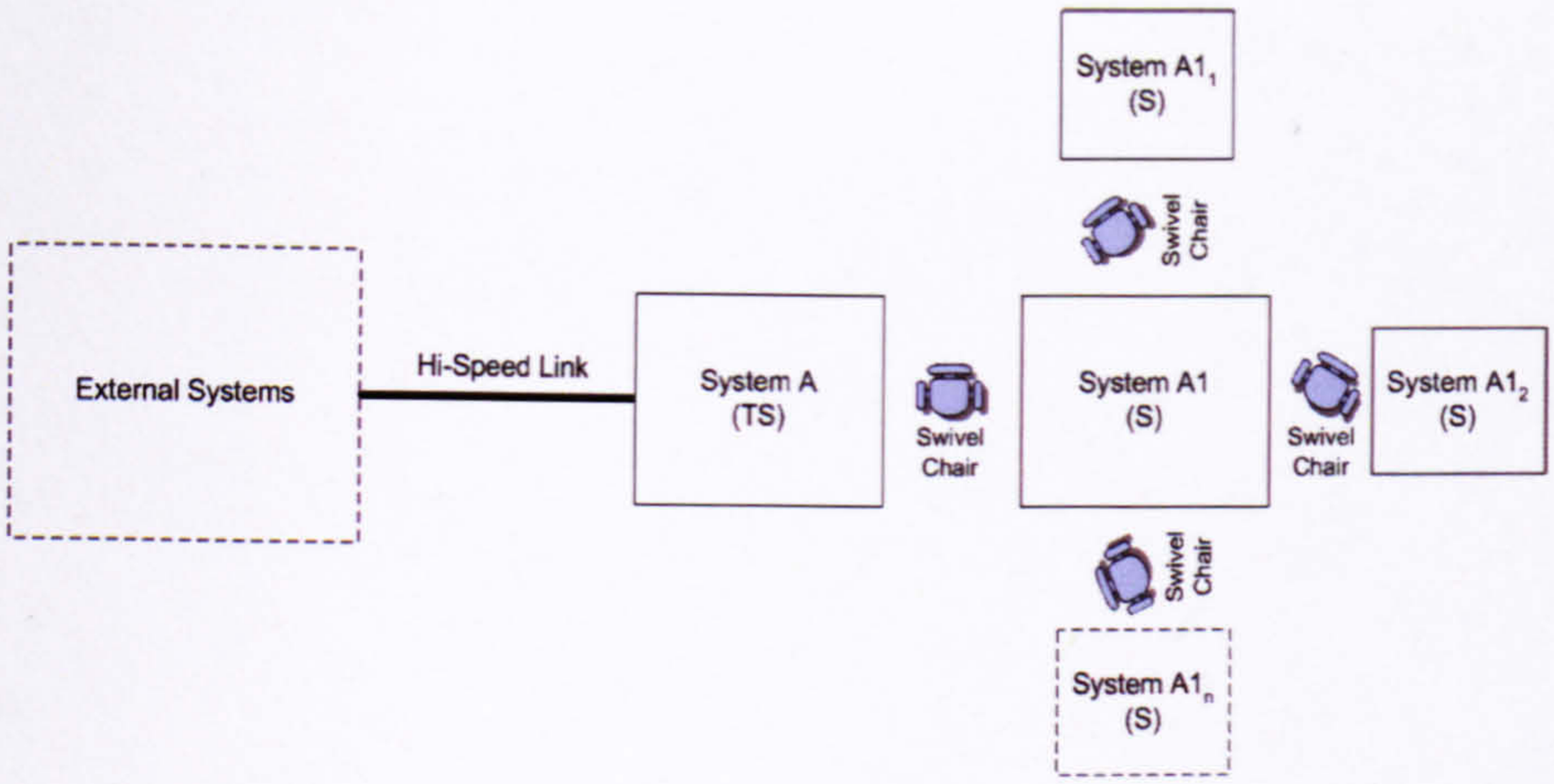

FIGURE 4.4: Generic high-level overview of potential configuration of future military systems.

Previous drives towards reducing the time of the kill chain have focused on processes and technologies such as:

- Precision guided munitions.
- All-weather munitions.
- Integrated sensing and effecting capability.

Such developments have significantly reduced the launch and flight times of munitions. However, in order to realise the potential of such information-intensive enhancements, it is important to remove any unnecessary steps in the information flow and decision chain as alluded to in [96]. Physically partitioned infrastructures are such an unnecessary step, particularly in future battlefield systems, which require increased information processing and sharing for Battle Damage Assessment (BDA) and Shared Awareness (SA) purposes.

4.4 Non-Partitioned Infrastructures

To address the inefficiencies introduced by air gaps two sequential programmes of work were carried out in the NITEworks project. Beyond NITEworks and the military community commercial requirements to securely label, store and transport electronic data have led to the design of mechanisms such as "Enhanced Security Services for S/MIME" [97], however these have been limited to specific applications. The NITEworks Intelligence, Surveillance, Target, Acquisition and Reconnaissance (ISTAR) Theme, Phase 1 study [98], assessed the UK Information Requirement Management (IRM) processes, and demonstrated significant benefits to NEC of automated information transfer in the

form of non-partitioned infrastructures. Principal improvements were identified in information sharing and therefore SA. In relation to information security, these improvements enhance the requirement for timeliness. However, confidentiality and integrity were not considered in this phase in order to highlight the deficiencies of non-partitioned infrastructures from an information security point of view.

In [98] only published (pull) communications were considered it is believed the improvements identified are generic in nature and may be applicable to real-time push communications as is the requirement in the IFPA project. An overview of the setup used in the ISTAR Theme, Phase 1 study is given in figure 4.5, where it can be seen that numerous separate databases have been collated into one. Providing more coherent operations through reduced repetition and synchronisation of information.

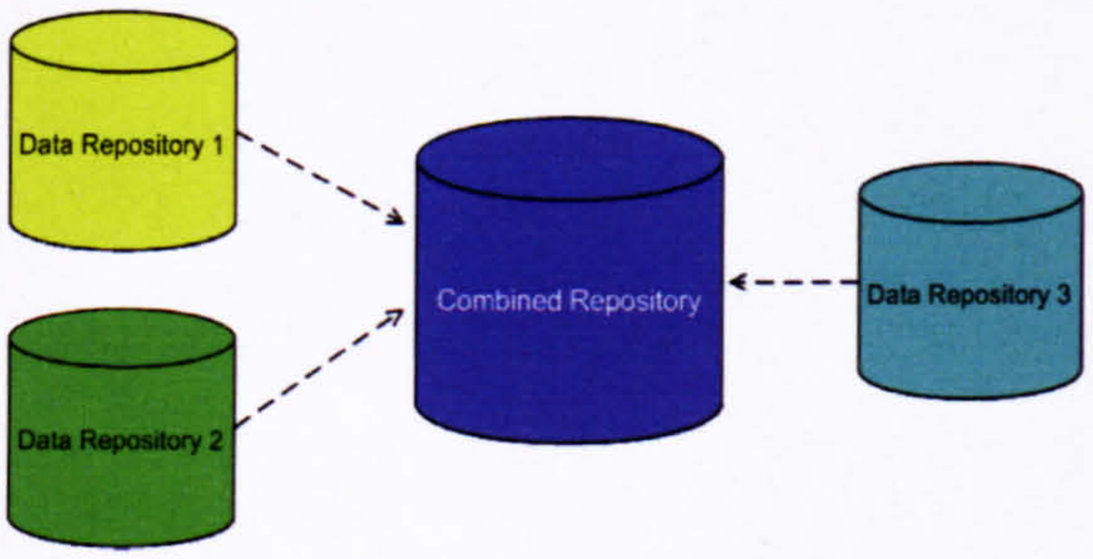

FIGURE 4.5: Overview of NITEworks system setup improving communications.

Clearly the security of physically partitioned, and the operational efficiency of non-partitioned, infrastructures is required in combination. This is known as logical infrastructure partitioning, allowing for successful security accreditation and enabling true NEC through secure and automated information transfer. This was also established in the NITEworks ISTAR Theme, Phase 2 study [99], which addressed UK Collection Coordination issues whilst utilising a US-provided planning tool that would be used in a US-led, US/UK coalition supporting expeditionary operations. The core of this work was the identification of a security architecture that permitted a high level of data flow, enabling the effective transfer of ISTAR information between UK/US and US/UK - as a result demonstrating the ability to achieve enhancements in timeliness, whilst implementing strict security controls to preserve the confidentiality and integrity of information.

4.5 Logical Infrastructure Partitioning

From figure 4.6 it can be seen that as information filters down the command chain, from highly complex strategic aims (high-level government policy reports) down to low-complexity actions (e.g. move to location x at time y), the requirement for timeliness

increases. Moreover the resolution of the information also increases, resulting in well-defined and unambiguous commands. The tactical battlefield can be seen to be at the actions stage in the command chain. Therefore it is argued that although the combination of communications in time-critical tactical battlefield systems such as IFPA are numerous; such exchanges exhibit sufficient precision and redundancy for pre-definition according to well-defined sets. These communications may then be securely executed in the form of automated M2M communications removing the overheads associated with manual information transfer. This is in-line with contemporary risk management thinking, which proposes the limitation of processes[§] to operate only on data, for which the interpretation is bounded. This reduces the risk from exploitation of validation flaw vulnerabilities as discussed in chapter 1. The severity of such flaws is highlighted by the fact that validation flaw vulnerability in the Microsoft Distributed Component Object Module (DCOM) Remote Procedure Call (RPC) protocol was responsible for the Blaster [50] worm.

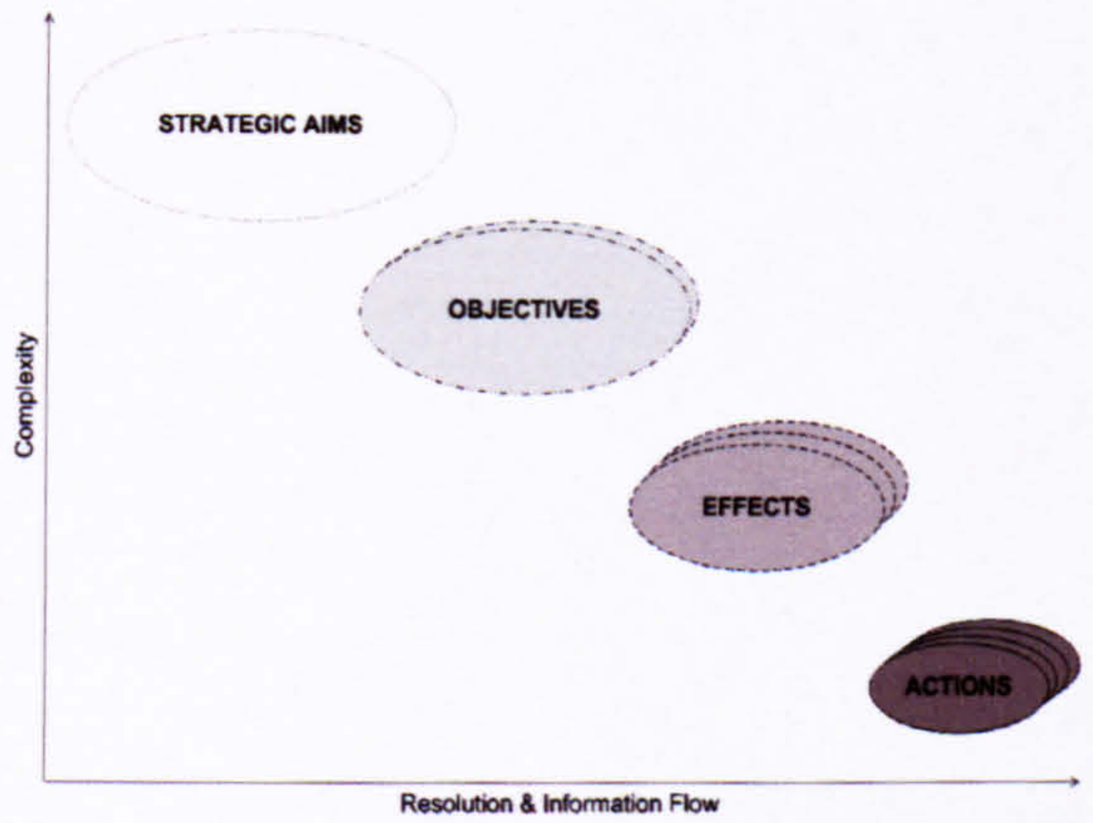

FIGURE 4.6: Illustration of the complexity of information in the command chain.

4.5.1 Tactical Battlefield Scenario Communications

From background studies in IFPA [87] it has been found that M2M communications although operationally efficient introduce inherent risks. These risks were identified using HMG infosec standards 1 [100] and 3 [48]. The primary reason for the increased risk is due to the interactive and extensive nature of M2M communications, allowing potential compromises to execute in a dynamic and rapid fashion. However such risks are not so significant, indeed increases in risk due to electronic connections are offset by the improved auditing and accounting capability introduced by all electronic transfers. Therefore such a method may be acceptable if a *trustworthy mechanism* is used to manage the information exchange. From section 4.4 it can be seen that non-partitioned

[§]A process in this context does not strictly relate to a software process, instead relating to any abstract computing capability operating on information in any way.

infrastructures as discussed in [98], only fulfil the information security requirement for timeliness. A trusted filtering mechanism would aid in the preservation of confidentiality and integrity completing the information security triangle.

It is proposed to utilise HMG Infosec standards to devise a trust relationship model incorporating the various levels of confidence placed in the communicating machines. One option is to restrict machines to data they may receive based upon the *subject-privilege-object* model, where for confidentiality purposes the *subject* refers to the receiving system. The privilege is the level of trust placed in this system and the *object* is the data (with a classification) to be received. Such a model would provide secure logical partitioning and therefore efficient segregation adhering to the core security principles of least-privilege and need-to-know. High-level caveats may be mapped to message labels with the trusted filtering mechanism utilised for the management and routing of such messages.

From background research into current UK military tactical battlefield systems at the application [102] and infrastructure [91] levels, it has been found that fine-grained security is difficult to achieve due to constraints in messaging structures and bandwidth. Therefore in order to achieve fine levels of segregation, changes in application or infrastructure messaging structure would be required along with the increase in bandwidth of the infrastructure, particularly if numerous virtual tunnels are to be created between entities which are securely communicating. However the costs associated with such a change are perceived to be very high, this technique also has drawbacks in terms of its limited extensibility. As applications wishing to add a caveat in the future would require upgrades to all relevant network interfaces, which may be numerous. Therefore it is proposed to utilise an independent technique, which is secure, scalable and efficient: *intra-system message filtering.*

4.5.2 Intra-System Message Filtering

Intra-system message filtering allows individual system owners to apply appropriate segregation rules based upon the trust associated with sub-systems. Figure 4.7 gives a high-level systems overview of the proposed architecture for an intra-system message filtering capability. From this it can be seen that standard messaging is still transported over the infrastructure meaning minimum interruption to other applications utilising the same network. However within an individual business process, sub-systems can be separated and protected from one another and the rest of the network through a message filtering mechanism. Messages are received by the trusted component of the system which will then convert the messages to XML, with only required (mission specific) data

passed on to the end system via a trusted filter[¶]. A possible drawback of using an intra-system message filtering capability is that all systems connected to the classification *high*[‖] backbone must be highly trusted[**]. Therefore any system which is not trusted to this *high* level is unable to connect directly to the backbone, resulting in a hierarchical network structure.

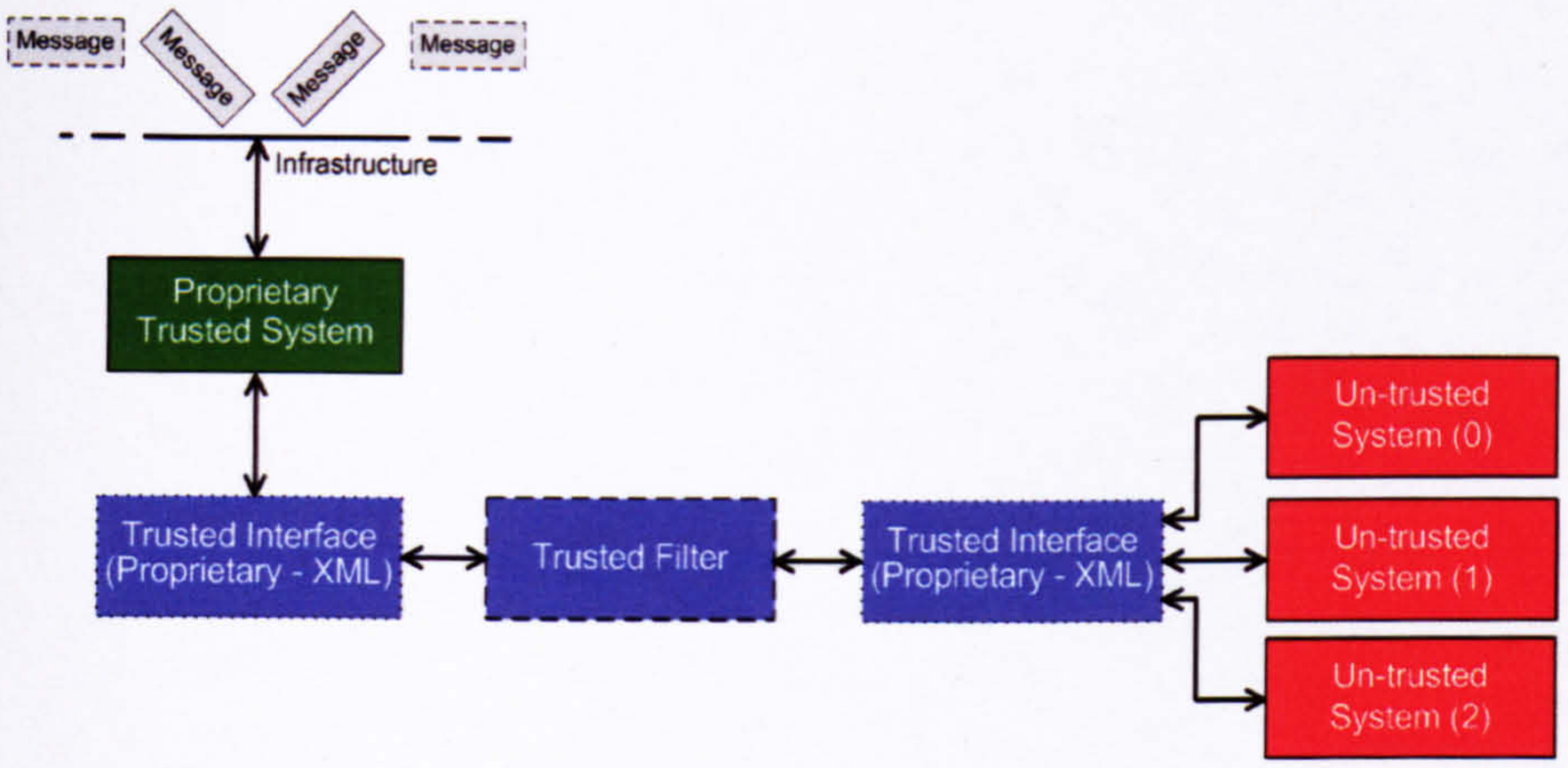

FIGURE 4.7: Systems Overview of the proposed intra-system message filtering.

The main advantage of utilising an XML based solution is due to the Schema Definition functionality provided in the XML standard. This allows for an off the shelf solution for *pre-definition*, *classification* and *validation* of messages[††]. Systems may only send and receive data for which they have the necessary security privileges. Another major driver for using an XML based solution for electronic communications particularly M2M is due to its requirement as stated in [103], XML is mandated as a *non-tradable* standard for data exchange, between UK MoD systems.

System Clearance	Message Classification
0	VH (Top Secret)
1	H (Secret)
2	M (Confidential)
3	L (Restricted)

TABLE 4.1: A simple association between system clearance & message classification.

Messages may be classified on the basis of element values as well as the existence of certain elements. An example would be to base classification of a message on the time

[¶]The actual data passed to the un-trusted system may not be in the form of XML instead, the original message may be passed from trusted to un-trusted system(s).

[‖]All systems are equally trusted with no provision for the principle of least-privilege.

[**]HMG approved design and development processes.

[††]Issues such as leaking and parsing information through commenting and CDATA fields need to be further assessed.

of executing actions. As the time of execution moves further into the future the classification may increase. An un-trusted system may then only receive such information after a given time interval according to its associated clearance, thus preserving the principle of least-privileges. In such a time based approach a higher clearance would indicate the ability to receive planned attack information at an earlier stage. Such information which is only classified for a limited period of time is termed a *short-term secret* in a military context. Overall a context-based access control policy as described in chapter 3 could be used to control the flow of information to different machines. A simplistic model for associating a systems clearance with the level of classification of information is represented in table 4.1, where message classifications are analogous to current HMG protective marking standards.

The *trusted interface* mechanism illustrated in figure 4.7 must efficiently convert messages from proprietary military messages such as those used in Fire Control Battlefield Information System Application (FC BISA) [102] and Bowman [104] to the standard XML format, as well as correctly labelling converted messages with classifications according to the content of the information contained. As a more open systems architecture is adopted in military information systems the *trusted filter* should have the ability to authenticate systems wishing to communicate as well as authorising all M2M information exchanges, according to clearance of systems and the classification of information to be exchanged.

From figure 4.7 it can be seen that messages arrive from external systems using current proprietary formats, those messages which are intended for un-trusted systems must then be converted to a standard XML format via the trusted interface. In order to preserve the confidentiality of information residing on the trusted system(s), XML messages are then passed to the trusted filter, which validates and transmits only those messages, which are authorised according to the clearance of the end (un-trusted) system(s) and the classification of the message(s) in question. Messages sent from un-trusted to trusted system(s) such as BDA are also converted to XML (if not already in XML) and then validated by the trusted filter. This will aid in preserving the integrity of trusted system(s), by ensuring only valid (well-formed) data is received by trusted systems.

4.5.2.1 Formal Representation of Trusted Filter

The security model described previously for M2M communications only restricts the information which a machine may receive, as all transmissions must be arbitrated by a trusted filter mechanism. The objective of the trusted filter is to allow a machine to receive classified information iff that machine has sufficient clearance. More formally:

$$\forall s : \mathbf{S} \wedge \forall o : \mathbf{O} | (CLE(s) \geq_d CLA(o)) \Rightarrow rec(s, o) \tag{4.1}$$

Given some subject $s \in \mathbf{S}$ (where $\mathbf{S}$ is the universal set of subjects) let $CLE(s) \in \{0, 1, 2, 3\}$ represent the clearance of the subject s and given some information object $o \in \mathbf{O}$ (where $\mathbf{O}$ is the universal set of objects) let $CLA(o) \in \{\text{TS, S, C, R}\}$ represent the classification of the information object o.

The relation $\geq_d$ indicates that the clearance of some subject s given by $CLE(s)$ is sufficiently dominant over the classification of some information object o given by $CLA(o)$, for s to receive o. The mappings from clearance of a subject to the classification of objects is given in table 4.1. From table 4.1 it can be seen that if a subject s_1 is cleared to level 0 then s_1 may access information classified up to and including TS, whereas if subject s_2 is cleared to a level 3 then s_2 may access information classified up to and including R. The flow of information object o to subject s is represented by $rec(s, o)$, which is a tuple of the ordered form: $S \times O$.

4.5.2.2 Security Requirements and Limitations

The trusted filter mechanism described previously must be certified and accredited to the appropriate common criteria level before it is deployed to operate on sensitive data. This process itself can be time consuming and costly, however due to its simplicity (use of XML schema definition) it is believed such costs would not be prohibitive. Ultimately however the hardware upon which any such filter may be hosted must also be appropriately accredited, which could increase the costs and potentially limit its deployment in the most sensitive of environments.

4.6 Summary

In this chapter the *proliferation problem* has been identified at the tactical battlefield level. However it is suggested that this will become increasingly prevalent in other scenarios as demonstrated in the NITEworks ISTAR Theme studies. If alternative solutions to the current cautious approach of physically partitioned infrastructures are not sought, then it is likely that security accreditation requirements will prove a major stumbling block in allowing the military to not just *do things better*, but *do better things*, which according to the UK MoD [1] is necessary to realise the potential of NEC.

Intra-system message filtering using XML Schema Definitions has been introduced as a potential COTS technique to overcome such limitations. By removing the human-in-the-loop for the actual transfer of data from one machine to another, the information security requirement for timeliness is addressed, whilst removing the inaccuracies inevitably introduced by a human operator, particularly under stressful conditions. Such

an approach would limit potential bottlenecks in information flow due to air-gaps, allowing secure and rapid transfer of information through automated processes. Unlike other efforts in the past such as "Enhanced Security Services for S/MIME" intra-system message filtering as discussed here is application agnostic, meaning it can be applied to a number of applications as it can be implemented at the networking layer.

Clearly if the technique of intra-system message filtering is to be realised, then a lot of work is needed in order to design effective schemas for information exchange between systems, however once such definitions are in place interfacing systems will become a much simpler and efficient task than at present.

Chapter 5

Conclusions

The increasing sophistication and mobility of computing devices is leading to a ubiquitous computing environment. This has enabled an always-on society requiring the use of information-based services in real-time. Such evolution is reflected in the three primary areas of research undertaken in this thesis:

- **Prioritisation of network security services** - extended the original compromise path threat analysis technique to assess and prioritise dynamic threat's posed to individual devices in a network.

- **Distributed security for decentralised information sharing** - distinguished between context-aware and context-dependant access control and extended the original RBAC model to consider context-dependant RBAC.

- **XML-based validation for sensitive information services** - identified the severity of the proliferation problem due to air-gaps in sensitive infrastructures, and offered intra-system message filtering as an application agnostic approach to providing secure electronic transfer of sensitive information.

From each of the three case studies undertaken it is clearly evident that the information security requirements for confidentiality and integrity remain the same. However timeliness is becoming an increasingly important aspect of the requirement for availability, which has traditionally focused on the need for fault-tolerance and graceful degradation. This is exemplified in the research undertaken here, for example in chapter 2 a technique is described to better manage the risks due to *zero day attacks*. Zero day attacks describe software vulnerabilities which are being identified and maliciously exploited before software vendors can even produce and disseminate relevant patches to end users. Similarly chapters 3 and 4 offer alternatives to current security processes and procedures, which have been shown to introduce bottlenecks due to their time consuming nature. The need to improve timeliness can also be seen as the underlying driver behind the

introduction of technologies such as Single Sign-On (SSO) and Identity Management (IdM). SSO allows users to access many applications by authenticating only once rather than individually for each application. This saves time on the users behalf and reduces the cost [105] due to password reset requests as users only need to remember one set of security credentials (e.g. username/password combination). IdM solutions are used to automatically provision and de-provision user accounts across multiple applications, reducing the time taken to setup and remove user accounts.

The work described in this thesis could be exploited to improve future information security solutions, however much effort is required to realise this. The metrics identified for the prioritisation of network security services must be further refined to improve the percentage of devices distinguished on a given network. Indeed such metrics may be of varying significance and even different for distinct end users. The cost benefit analysis of enabling decentralised information sharing using distributed security mechanisms, has been proposed as a basis for further work in the SEAS DTC, as well as the cross-industry project on Context-Aware Data-Centric Information Sharing (CONSEQUENCE). Similarly since the work on XML-based data validation was undertaken related activity on XML gateways has been identified in both the military [106], [107] and civil [108] domains.

Clearly COTS based systems have many advantages over bespoke solutions, such as reduced Total Cost of Ownership (TCO) and faster times to market/service. However it is imperative that sensitive business operations, such as the military and financial institutions, fully understand and address the security limitations of COTS technologies, which are usually developed with an emphasis on functionality rather than security. Therefore to de-risk any potential project from a security accreditation point-of-view, security requirements must be considered at the earliest possible phase and throughout the systems lifecycle, providing *built-in* and not *bolt-on* security. Today COTS technologies are slowly moving towards built-in and through-life security, which can be seen in concepts such as de-perimeterisation, and initiatives such Open Web Application Security Project (OWASP) and Transglobal Secure Collaboration Programme (TSCP). In order to deliver concepts such as de-perimeterisation and improve best practice initiatives such as OWASP and TSCP, it is believed that the specialist security engineering experience developed by companies such as BAE Systems over many years will play a crucial role.

In summary a pro-active risk management approach must be adopted for information security, where the requirements and practices need to be constantly assessed and evolved. Ultimately it is believed such an approach will deliver appropriate and therefore business enabling security.

Bibliography

[1] MoD, *NEC Handbook: JSP777*, London, UK, 2005.

[2] A.S. Grove, "Only the Paranoid Survive", *Doubleday Publishing 1st Edition*, New York, 1996.

[3] R. Brown, J. Griffin, A. Norman and R. Smith, "Why HP did not get Blastered", *HP Laboratories Technical Report HPL-2004-188*, Bristol, UK, 2004.

[4] ISO, "ISO 7498-2:Information Security Architecture Reference Model", *ISO Information Technology Standards*, Geneva, Switzerland, 1989.

[5] Act of Parliament, "Data Protection Act 1998", *Her Majesty's Stationary Office*, London, UK, 1998.

[6] D. Bell and L. LaPadula, "Secure Computer System: Unified Exposition and Multics Interpretation", *MITRE Corp. Technical Report MTR-2997 Rev. 1*, Bedford, Massachusetts, USA, 1975.

[7] D. Clark and D. Wilson, "A Comparison of Commercial and Military Security Policies", *In Proceedings of IEEE Symposium on Security and Privacy*, pp. 184-194, 1987.

[8] S. Lipner, "Twenty Years of Evaluation Criteria and Commercial Technology", *In Proceedings of IEEE Symposium on Security and Privacy*, pp. 111-112, 1999.

[9] K.J. Biba, "Integrity Considerations for Secure Computer Systems", *MITRE Corp. Technical Report MTR-3153*, Bedford, Massachusetts, USA, 1977.

[10] PCI Security Standards Council, "Payment Card Industry Data Security Standard", Available Online:http://www.pcisecuritystandards.org/tech/, last accessed on 8^{th} May 2007.

[11] A. Perrig, R. Canetti, J.D. Tygar and D. Song, "Efficient Authentication and Signing of Multicast Streams over Lossy Channels", *In Proceedings of IEEE Symposium on Security and Privacy*, 2000.

[12] D.J. Bernstein, "SYN Cookies", Available Online:http://cr.yp.to/syncookies.html, last accessed on 8^{th} May 2007.

[13] National Buraeu of Standards, "Data Encryption Standard", *NBS FIPS PUB 46*, USA Department of Commerce, 1977.

[14] NIST, "Advanced Encryption Standard", *FIPS Publication 197*, Gaithersburg, Marlyland, USA, 2001.

[15] R.L. Rivest, A. Shamir and L.M. Adleman, ."On Digital Signatures and Public Key Cryptography", *MIT Laboratory for Computer Science Technical Report MIT/LCS/TR-212*, 1979.

[16] T. ElGamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", *In Proceedigs of Advances in Cryptology 84*, Springer-Verlag, pp. 10-18, 1985.

[17] W. Diffie and M.E. Hellman, "New Directions in Cryptography", *IEEE Transactions on Information Theory IT-22* pp. 644-654, 1976.

[18] R.L. Rivest, "The MD5 Message Digest Algorithm", *RFC1321*, 1992.

[19] NIST, "Announcing the Standard for SECURE HASH STANDARD", *FIPS Publication 180-1*, Gaithersburg, Marlyland, USA, 1995.

[20] P. Karger and A. Herbert, "An Augmented Capability Architecture to Supprot Lattice Security and Tractibility of Access", *In Proceedings of IEEE Symposium on Security and Privacy*, pp. 2-12, 1984.

[21] R.J. Anderson and F.A.P. Petitcolas, "On The Limits of Steganography", *In Proceedings of IEEE Journal of Selected Areas in Communications*, 16(4):474-481, 1998.

[22] W. Jansen, S. Gavrila, V. Korolev, R. Ayers and R. Swanstrom, "Picture Password: A Visual Login Technique for Mobile Devices", *NIST Technical Report NISTIR 7030*, Gaithersburg, Maryland, USA, 2003.

[23] G.E. Blonder, "Graphical Password", *Lucent Technologies Inc. US Patent 5559961*, Murray Hill, New Jersey, USA, 1995.

[24] R. Dhamija and A. Perrig, "Deja Vu: A User Study Using Images for Authentication", *In Proceedings of 9th USENIX Security Symposium*, 2000.

[25] A. Goldstein and J. Chance, "Visual Cognition for Complex Configurations", *In Proceedings of Perceptions and Psychphysics*, pp. 237-241, 1971.

[26] Microsoft TechNet, "Let Me In: Pocket PC User Interface Password Redirect Sample", *Microsoft Knowledge Base Article - 314989*, Microsoft Corp., One Microsoft Way, Redmond, Washington 980521, USA, 2003.

[27] CERT/CC, Available Online: http://www.cert.org/stats/cert_stats.html, last accessed on 17^{th} October 2005.

[28] R. Dacey, "Effective Patch Management is Critical to Mitigating Software Vulnerabilities", Testimony before the Subcommittee on Technology, Information Policy, Intergovernmental Relations, and the Census, House Committee on Government Reform, United States General Accounting Office, Available Online: http://www.iwar.org.uk/comsec/resources/worm-virus-defense/GAO-final-testimony.pdf, last accessed on 17^{th} October 2005.

[29] T. Jennings, "Developing the Role of the CIO", *Information Economics Journal*, pp. 11-13, London, UK, 2005.

[30] Z. Hayat, J. Reeve and C. Boutle, "Prioritisation of Network Security Services", *In Proceedings of IEE Information Security*, London, UK, 2006.

[31] SETI@home, Available Online:http://www.setiathome.ssl.berkeley.edu/, last accessed on 17^{th} October 2005.

[32] L. Rogers and J. Allen, "Securing Information Assets: Security Knowledge in Practice", *CrossTalk Defense Software Engineering Journal*, US Air Force, Available Online: http://www.stsc.hill.af.mil/crosstalk/2002/11/rogers.html, last accessed on 17^{th} October 2005.

[33] B. Monahan, "Infrastructure Security Modelling for Utility Computing", *HP Laboratories Technical Report HPL-2005-04, Bristol*, UK, 2005.

[34] C. Alberts and A. Dorofee, "An Introduction to the OCTAVE Method", Software Engineering Institute Carnegie Mellon University, Available Online: http://www.cert.org/octave/methodintro.html#chars, last accessed on 17^{th} October 2005.

[35] Network Appliance Inc., "Antivirus Scanning Best Practices Guide", *Network Appliance Inc. Technical Report TR 3107*, Sunnyvale, California, USA, 2005.

[36] T. Nakayama and F. Ladley, "W32.Sasser.Worm", *Symantec Corporation Security Response*, Available Online: http://securityresponse.symantec.com/avcenter/venc/data/w32.sasser.worm.html, last accessed on 6^{th} February 2006.

[37] J. Meier, A. Mackman, M. Dunner, S. Vasireddy, R. Escamilla and A. Murukan, "Improving Web Application Security: Threats and Countermeasures", Microsoft Corporation Technical Library, Available Online: http://msdn.microsoft.com/library/default.asp?url=/library/en−us/dnnetsec/html/ThreatCounter.asp, last accessed on 17^{th} October 2005.

[38] B. Schneier, "Attack Trees: Modelling Security Threats", *Dr. Dobb's Journal*, Available Online: http://www.schneier.com/paper-attacktrees-ddj-ft.html, last accessed on 17^{th} October 2005.

[39] C. Salter, O. Saydjari, B. Schneier and J. Wallner, "Toward a Secure System Engineering Methodology", *In Proceedings of New Security Paradigms Workshop*, Charlotsville, Virginia, USA, 1998.

[40] J. Surdu, J. Hill, R. Dodge, S. Lathrop and C. Carver, "Military Academy Attack/Defense Network Simulation", *Advanced Simulation Technology Conference: Symposium on Military*, Government, and Aerospace Simulation, Orlando, Florida, USA, 2003.

[41] A. Moore, R. Ellison and R. Linger, "Attack Modeling for Information Security and Survivability", *Carnegie Mellon University Technical Note CMU/SEI-2001-TN-001*, Pittsburgh, Pennsylvania, USA, 2001.

[42] Ericsson AB, "Introduction to IMS", *Ericsson Technology*, Stockholm, Sweden, 2007.

[43] N. Bleech, "Visioning White Paper, What is Jericho Forum?", Available Online: http://www.opengroup.org/projects/jericho/uploads/40/6809/vision_wp.pdf, last accessed on 17^{th} October 2005.

[44] Cheops, Available Online: http://cheops-ng.sourceforge.net/, last accessed on 17^{th} October 2005.

[45] Nmap, Available Online: http://www.insecure.org/nmap/, last accessed on 17^{th} October 2005.

[46] K. Hughes and S. Wiseman, "Analysis of information security risks: Policy for protection through to implementation", *In Proceedings of 4th European Conference on Information Warfare and Security*, Glamorgan, UK, 2005.

[47] R. Lippmann, K. Ingols, C. Scott, K. Piwowarski, K. Kratkiewics, M. Artz and R. Cunningham, "Validating and Restoring Defense in Depth Using Attack Graphs", *In Proceedings of* 25^{th} *IEEE MILCOM Conference*, Washington DC, USA, October 2006.

[48] MoD, "HMG Infosec Standard No. 3", London, UK, 2001.

[49] Microsoft TechNet, "Network Access Protection Platform Architecture", *Microsoft TechNet White Paper*, Microsoft Corp., One Microsoft Way, Redmond, Washington 980521, USA, 2007.

[50] D. Knowles, F. Perriot and P. Szor, "W32.Blaster.Worm", *Symantec Corporation Security Response*, Available Online:

http://securityresponse.symantec.com/avcenter/venc/data/w32.blaster.worm.html, last accessed on 6^{th} February 2006.

[51] P. Szor and E. Chien, "CodeRed II", *Symantec Corporation Security Response*, Available Online: http://securityresponse.symantec.com/avcenter/venc/data/codered.ii.html, last accessed on 6^{th} February 2006.

[52] S. Staniford, V. Paxson and N. Weaver, "How to 0wn the internet in your spare time", *In Proceedings of 11th USENIX Security Symposium*, San Francisco, California, USA, 2002.

[53] J. Spencer and L. Sacks, "Modelling IP Network Topologies by Emulating Network Development Processes", *In Proceedings of IEEE Softcom Conference*, Split, Croatia, 2002.

[54] W. Aiello, F. Chung and L. Lu,"A random graph model for massive graph", *ACM Symposium on Theory of Computing*, pp. 171-180, Portland, Oregon, USA, 2000.

[55] BRITE, Available Online: http://www.cs.bu.edu/BRITE/, last accessed on 17^{th} October 2005.

[56] C. Palmer and J. Steffan,"Generating Network Topologies That Obey Power Laws", *In Proceedings of Global Internet Symposium*, San Francisco, California, USA, 2000.

[57] B. Waxman, "Routing of Multipoint Connections", *IEEE Journal on Selected Areas in Communications*, pp. 1617-1622, 1988.

[58] NeuroGrid, Available Online: http//www.neurogrid.net, last accessed on 17^{th} October 2005..

[59] K. Tocheva, M. Hypponen and S. Rautiainen, "Melissa", *F-Secure Corporation Computer Virus Information*, Available Online: http://www.f-secure.com/v-descs/melissa.shtml, last accessed on 6^{th} February 2006.

[60] E. Carrera and G. Erdelyi, "Mydoom", *F-Secure Corporation Computer Virus Information*, Available Online: http://www.f-secure.com/v-descs/novarg.shtml, last accessed on 6^{th} February 2006.

[61] SEAS DTC, Available Online: http://www.seasdtc.com/, last accessed on 11^{th} April 2007.

[62] ALADDIN, Available Online: http://www.aladdinproject.org/, last accessed on 11^{th} April 2007.

[63] J. Sandhu, A. Agogino and A. Agogino,"Wireless Sensor Networks for Commercial Lighting Control: Decision Making with Multi-Agent Systems", *In Proceedings of AAAI Workshop on Sensor Networks*, 2004.

[64] R. Tynan, D. Marsh, D. OKane and G. OHare, "Agents for Wireless Sensor Network Power Management", *In Proceedings of IEEE ICPPW Conference*, 2005.

[65] Z. Hayat, J. Reeve, C. Boutle and M. Field, "Information Security Implications of Autonomous Systems", *In Proceedings of* 25^{th} *IEEE MILCOM Conference*, 2006.

[66] A. Yavnai, "An Information-Based Approach for System Autonomy Metrics Part I: Metrics Definition", *In Proceedings of Performance Metrics for Intelligent Systems Workshop*, 2003.

[67] IBM Research, Autonomic Computing, IBM Corp., Available Online: http://www.research.ibm.com/autonomic/, last accessed on 11^{th} April 2007.

[68] J. Kephart and D. Chess, "The Vision of Autonomic Computing", *In proceedings of IEEE Computer*, pp. 4150, 2003.

[69] V. Tewari and M. Milenkovic, "Standards for Autonomic Computing", Intel Technology Journal, Intel Corp., Available Online: http://www.intel.com/technology/itj/2006/v10i4/3-standards/1-abstract.htm/, last accessed on 11^{th} April 2007.

[70] D. Alberts, J. Garstka and F. Stein, "Network Centric Warfare: Developing and Leveraging Information Superiority", *2nd edition, C4ISR Cooperative Research Program, Washington DC: Library of Congress*, 2000.

[71] IBM Research, "An Architectural Blueprint for Autonomic Computing", IBM Corp., Available Online: http://www-03.ibm.com/autonomic/pdfs/AC_Blueprint_White_Paper_4th.pdf/, last accessed on 11^{th} April 2007.

[72] Wikipedia, "Autonomic Computing", Available Online: http://en.wikipedia.org/wiki/Autonomic$_C$*omputing*/, *lastaccessedon*11^{th} April 2007.

[73] Octatron Inc., "SkySeer", Available Online: http://www.octatron.com/prodSkySeer.html/, last accessed on 11^{th} April 2007.

[74] NITEworks, Available Online: http://www.niteworks.net/publications/, last accessed on 11^{th} April 2007.

[75] G. Chen and D. Kotz, "A Survey of Context-Aware Mobile Computing Research", *Dartmouth College Technical Report TR2000-381*, Computer Science Department, New Hampshire, USA, 2000.

[76] P. Coppola, V. Della Mea, L. Di Gaspero, S. Mizzaro, I. Scagnetto, A. Selva, L. Vassena and P.Z. Rizi, "Information Filtering and Retrieving of Context-Aware Applications Within the MoBe Framework", *In Proceedings of Context-Based Information Retrieval Workshop*, Paris, France, 2005.

[77] A. Smailagic, D.P. Siewiorek, J. Anhalt, F. Gemperle, D. Salber and S. Weber, "Towards Context Aware Computing: Experiences and Lessons", *In IEEE Intelligent Systems*, pp. 38-46, 2001.

[78] G. Zhang and M. Parashar, "Dynamic Context-Aware Access Control for Grid Applications", *In Proceedings of joint IEEE/ACM Internationsal Workshop on Grid Computing*, 2003.

[79] J. Hu and A.C. Weaver, "A Dynamic, Context-Aware Security Infrastructure for Distributed Healthcare Applications", *In Proceedings of 1st PSPT Workshop*, 2004.

[80] A. Toninelli, R. Montanari, L. Kagal and O. Lassila, "A Semantic Context-Aware Access Control Framework for Secure Collaborations in Pervasive Computing Environments", *In Proceedings of International Conference on Semantic Web*, 2006.

[81] S. Yokoyama, E. Kamioka and S. Yamada, "An Anonymous Context-Aware Access Control Architecture", *In Proceedings of International Workshop on Managing Context Information and Semantics in Mobile Environments*, 2006.

[82] S. Gavrila, D.R. Kuhn, D.F. Ferraiolo, R. Sandhu and R. Chandramouli, "Proposed NIST Standard for Role-Based Access Control", *ACM Transactions on Information and System Security*, pp. 224-274, 2001.

[83] J. Barkley, D. Ferraiolo and D. Kuhn, "A Role Based Access Control Model and Reference Implementation within a Corporate Intranet", *ACM Transactions on Information and System Security*, pp. 34-64, 1999.

[84] H. Feinstein, R. Sandhu, E. Coyne and C. Youman, "Role-Based Access Control Models", *In Proceedings of IEEE Computer*, pp. 38-47, 1996.

[85] S. Farrell and R. Housley, "An Internet Attribute Certificate Profile for Authorization", *Internet RFC3281*, 2002.

[86] MOSQUITO Consortium, "Specification of Context-Sensitive Security Infrastructure", *MOSQUITO Consortium Deliverable*, Available Online: http://www.mosquito-online.org/, last accessed on 11^{th} April 2007.

[87] IFPA, Available Online: http://www.mod.uk/, last accessed on 11^{th} April 2007.

[88] CVF, Available Online: http://www.mod.uk/, last accessed on 11^{th} April 2007.

[89] NIST, "Orange Book: Defense Trusted Computer System Evaluation Criteria", *Assistant Security of Defense C3I USA DoD 5200.28-STD*, 1985.

[90] K. Hyman, "RE306 Phase2.2-RNCSS/CMS Interface Options", *EDS Command Support Group Technical Report GB02-108100*, Hook, Hampshire, UK, 2004.

[91] Watchkeeper, Available Online:http://www.mod.uk/dpa/projects/watchkeeper.htm, last accessed on 11^{th} April 2007.

[92] Falcon, Available Online:http://www.mod.uk/dpa/projects/falcon.htm, last accessed on 11^{th} April 2007.

[93] Skynet V, Available Online:http://www.fas.org/spp/guide/uk/military/comm/skynet_5.htm, last visited 2005.

[94] M.J. O'Connell, "ARP RE 306 Combat Systems Interoperability, Integration and Performance", *AMS Integrated Sytems Division Technical Report CSIIP/DM/47*, Frimley, Camberley, Surrey, UK, 2005.

[95] A.J. Herbert, "Compressing the Kill Chain", *Air Force Magazine*, Arlington, Virginia, USA, 2003.

[96] K. Robinson, "Breaking Down the Barriers", *RUSI Defence Systems Journal*, pp.62-65, London, UK, 2004.

[97] P. Hoffmann, "Enhanced Security Services for S/MIME", *RFC2634*, 2004.

[98] S. Philips, "ISTAR Theme Phase 1: Customer Report", *NITEworks Team Technical Report NW/TH/IST/25*, Farnborough, Hampshire, UK, 2004.

[99] S. Philips, "ISTAR Theme Phase 2: Customer Report", *NITEworks Team Technical Report NW/TH/IST/102*, Farnborough, Hampshire, UK, 2005.

[100] MoD, "HMG Infosec Standard No. 1", London, UK, 1998.

[101] Z. Hayat, "IFPA Domain Based Security Modelling Report", *BAE Systems & IFPA JIPT Technical Report BAES-IFPA-SEC-REP-00345 Issue.1*, Frimley, Camberley, Surrey, UK, 2005.

[102] FC BISA, Available Online:http://www.logicacmg.com/constants/publications, last accessed on 11^{th} April 2007.

[103] MoD, "Interoperability & Compliance Assurance Information Coherence Check List: JSP602", London, UK, 2004.

[104] Bowman, Available Online:http://www.mod.uk/dpa/projects/bowman.htm, last accessed on 15^{th} June 2005.

[105] Microsoft Technet, "Password Management", *Microsoft Identity and Access Management Series*, Microsoft Corp., One Microsoft Way, Redmond, Washington 980521, USA, 2006.

[106] A. Thüummel and K. Eckstein, "Design and Implementation of a File Transfer and Web Services Guard Employing Cryptographically Secured XML Security Labels", *In Proceedings of the 7^{th} IEEE Information Assurance Workshop*, New York, USA, June 2006.

[107] T. Dean and G. Wyatt, "Information Exchange between Resilient and High-Threat Networks: Techniques for Threat Mitigation", *In Proceedings of NATO Symposium on Adaptive Defence in Unclassified Networks*, April 2004.

[108] Vordel Ltd., "Vordel XML Firewall", Available Online:http://www.vordel.com/products/vx_firewall.pdf, last accessed on 8^{th} May 2007.

Blank Page

Appendix A

Prioritisation of Network Security Services: Software Overview and Results

This appendix contains the class structures and high-level flow diagrams for the **pfcca** and **generator** software modules. Due to the development of both **pfcca** and **generator** upon legacy software, a device is referred to as a domain throughout this appendix.

A.1 Software Testing

This section contains various screen shots during testing of both the **pfcca** and **generator** software modules. As well as testing for cases of correct and expected input parameters, a few non-expected inputs were entered to test the simplistic error handling code design.

A.2 Source Code and Results

The source code for the **pfcca** and **generator** software modules as well as the results from the experiment described in chapter 2 can be found on the accompanying CD. Both of the software modules have been developed in C#. The appropriate files can be found in the following directories:

- **pfcca** (source directory) contains .cs files for this software module.
- **generator** (source directory) contains .cs files for this software module.

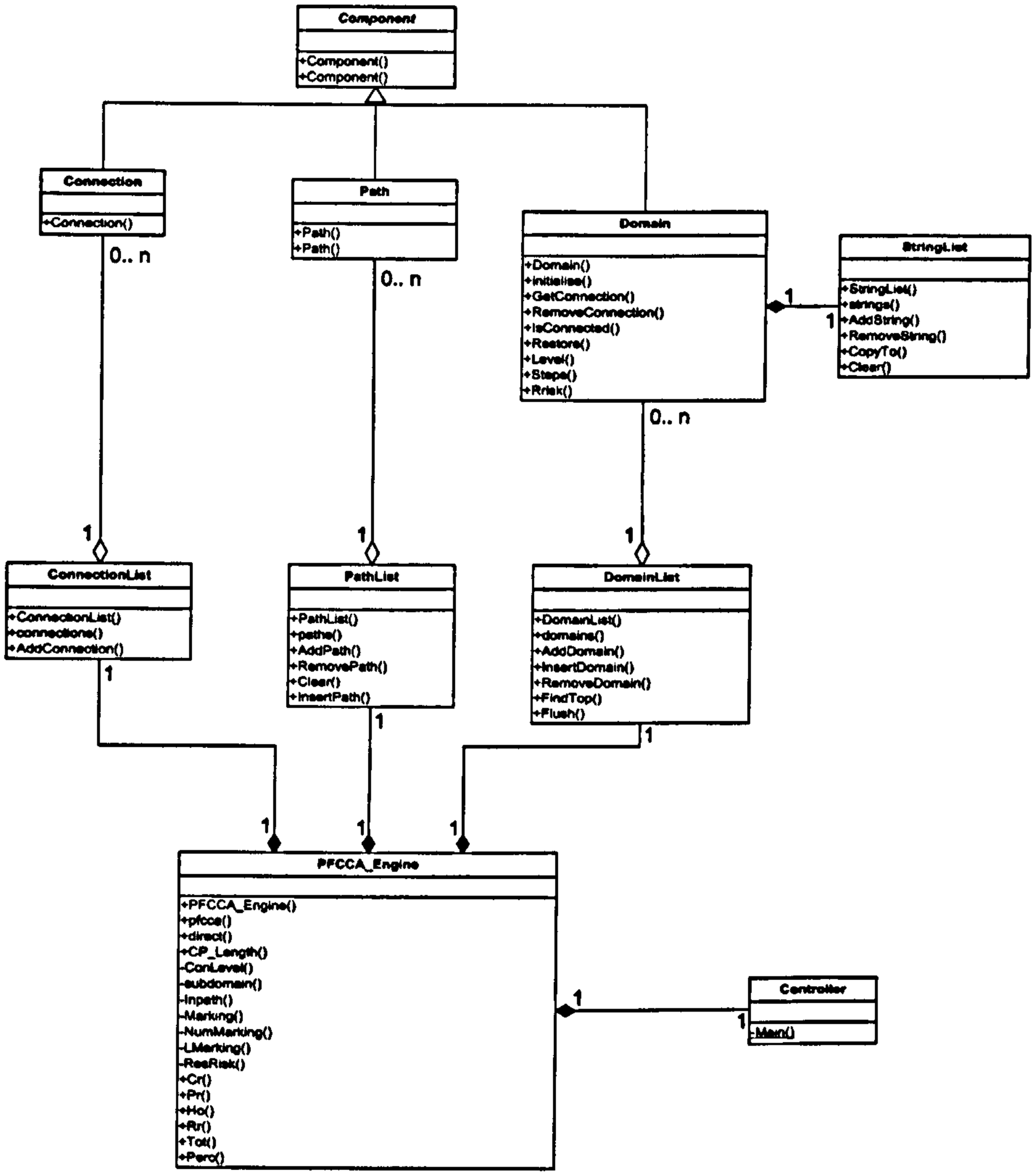

FIGURE A.1: UML diagram for class relationships in pfcca.

- **manual** contains .pdf version of the user manual for pfcca v1.2 and generator v1.0 software.
- **network security** contains the results in .csv format from the experiment described in chapter 2.

It must be noted that an executable for each software module is available in the *bin* directory of each of the source directories listed previously.

A.2.1 Building the Modules

All software modules described here compile and build under Microsoft Visual Studio.NET. You can compile them either by creating a project in the Visual Studio IDE,

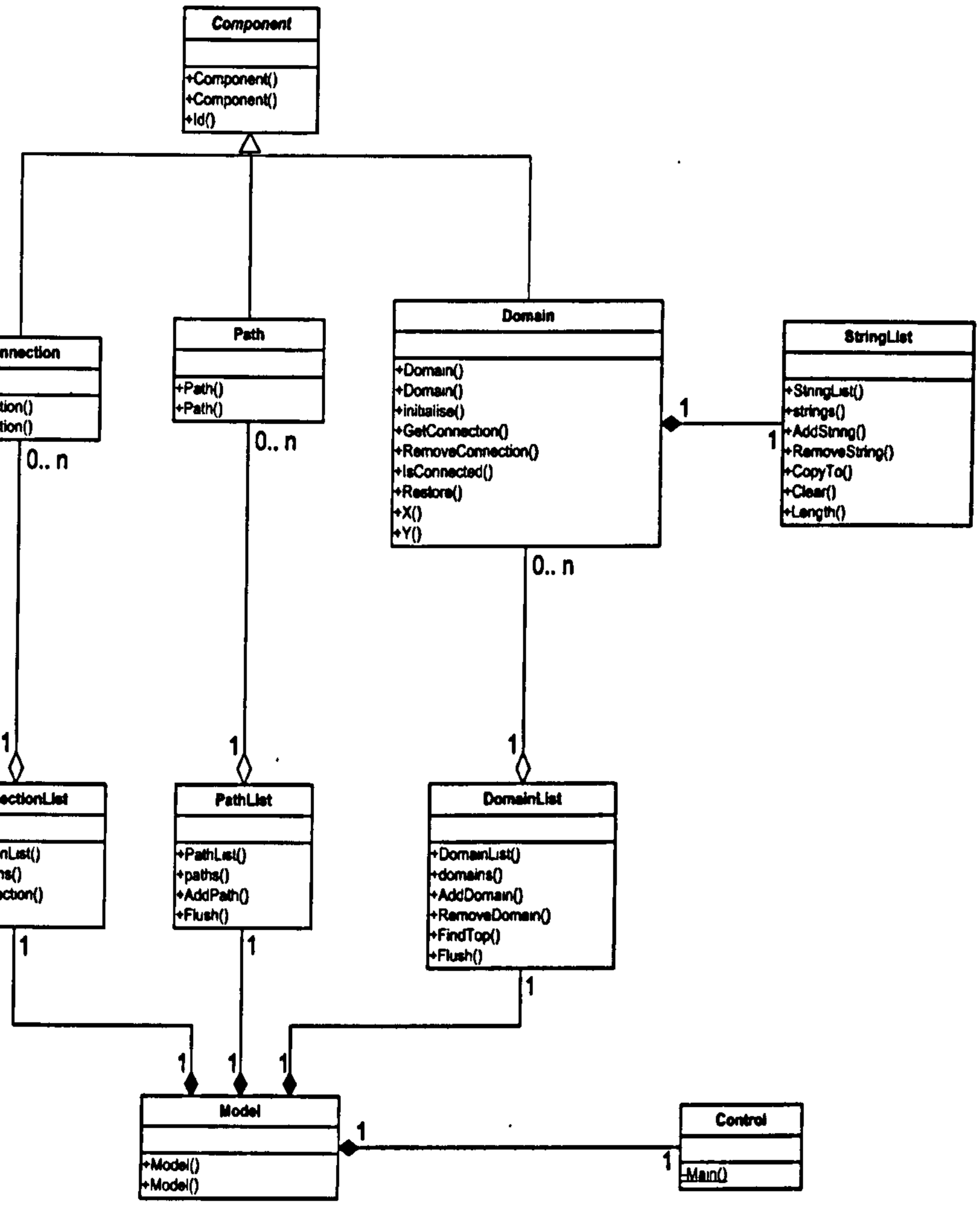

FIGURE A.2: UML diagram for class relationships in generator.

or by using the command line.

To compile any of the modules from the command line, first navigate to the correct directory and enter the command: **csc /define:DEBUG /optimize /out:*OUTPUT*.exe *.cs csc**, where ***OUTPUT*** is the required name of the output file. To run the program, now enter the command: *OUTPUT*.

To compile any of the modules from the IDE open the solution from *File→ Open Solution*, now go to *Build→ Build Solution* and then *Debug→ Start Without Debugging*.

See the *user manual* for the required inputs to each software module and expected outputs.

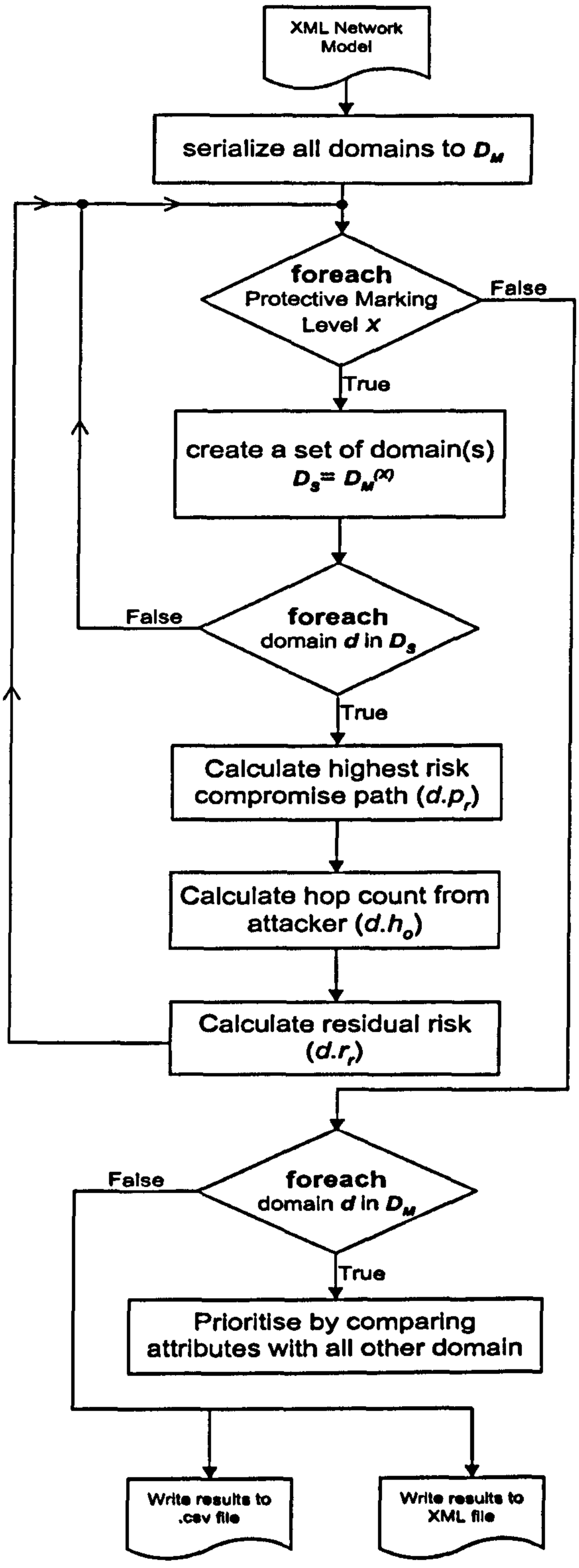

FIGURE A.3: High-level flow diagram for pfcca algorithm.

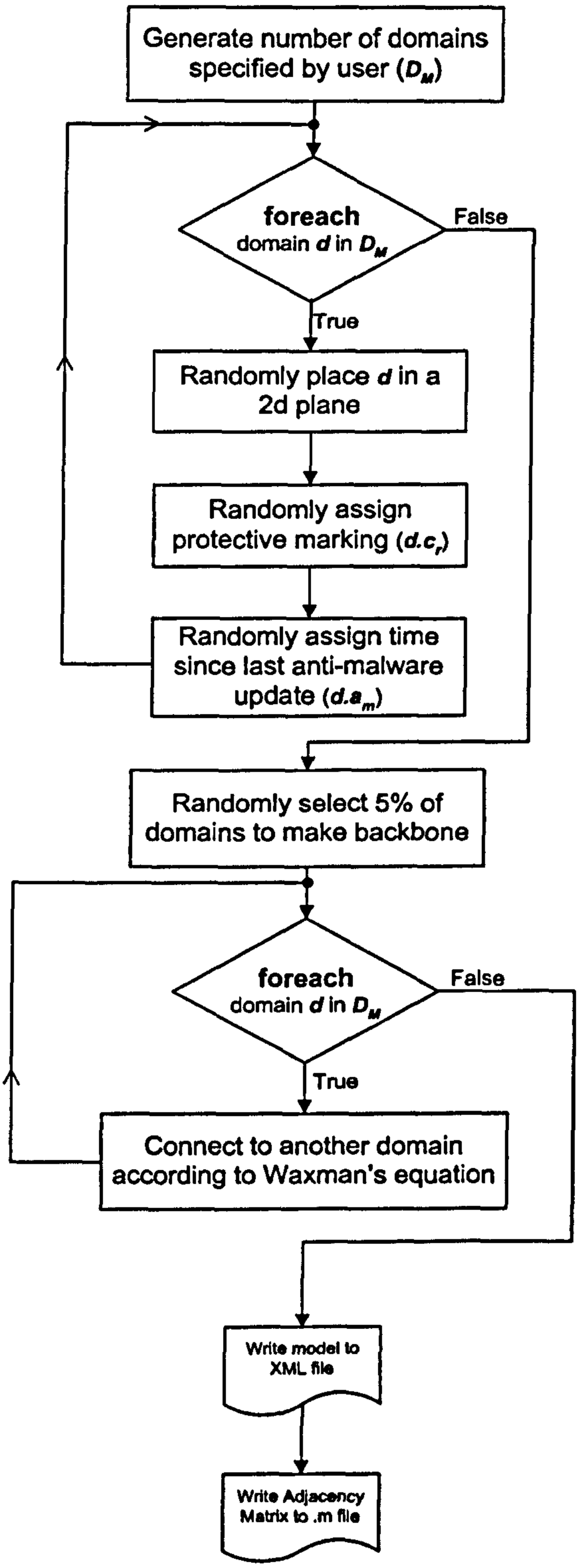

FIGURE A.4: High-level flow diagram for generator algorithm.

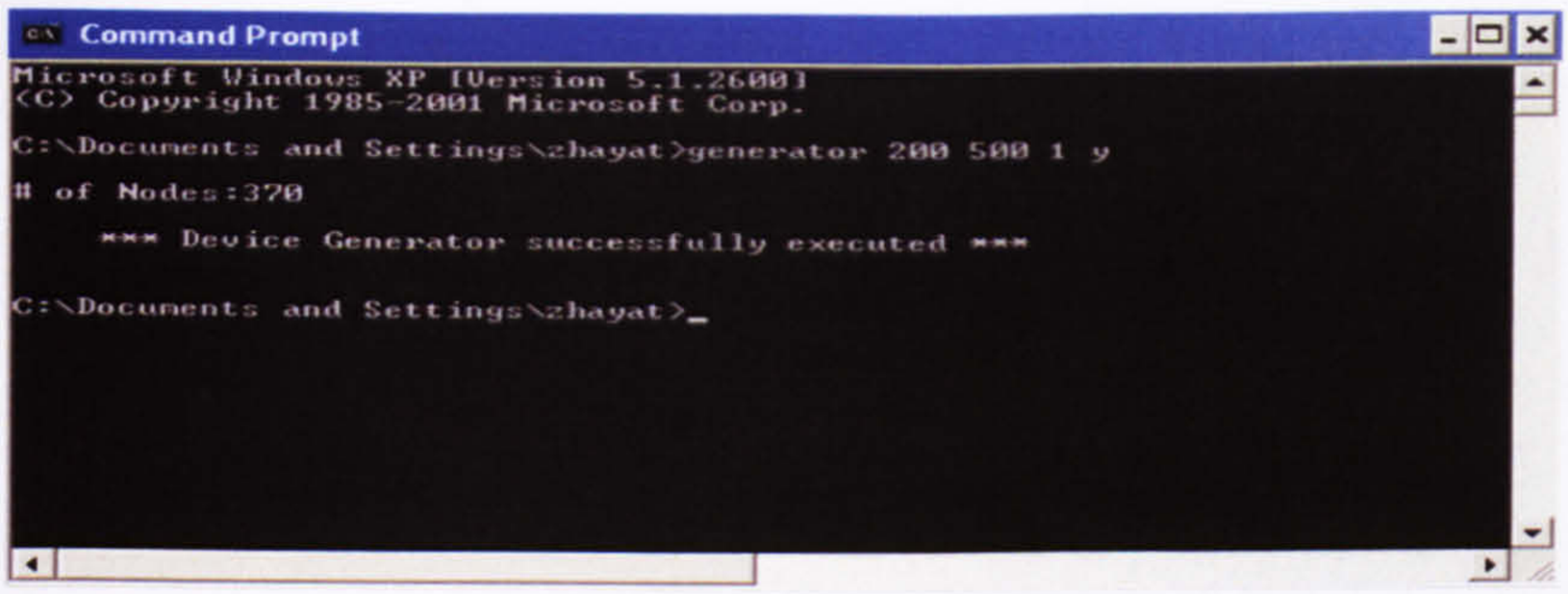

FIGURE A.5: Sufficient and correct input parameters to generator.

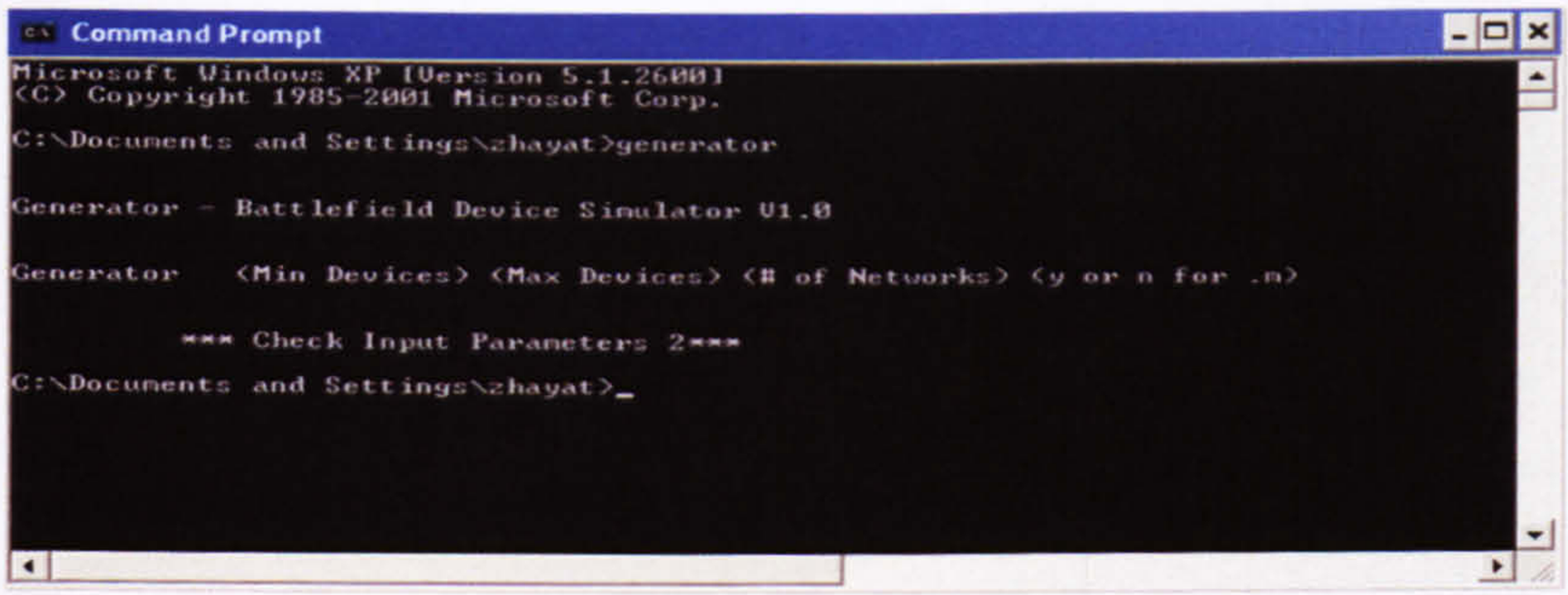

FIGURE A.6: No input parameters to generator.

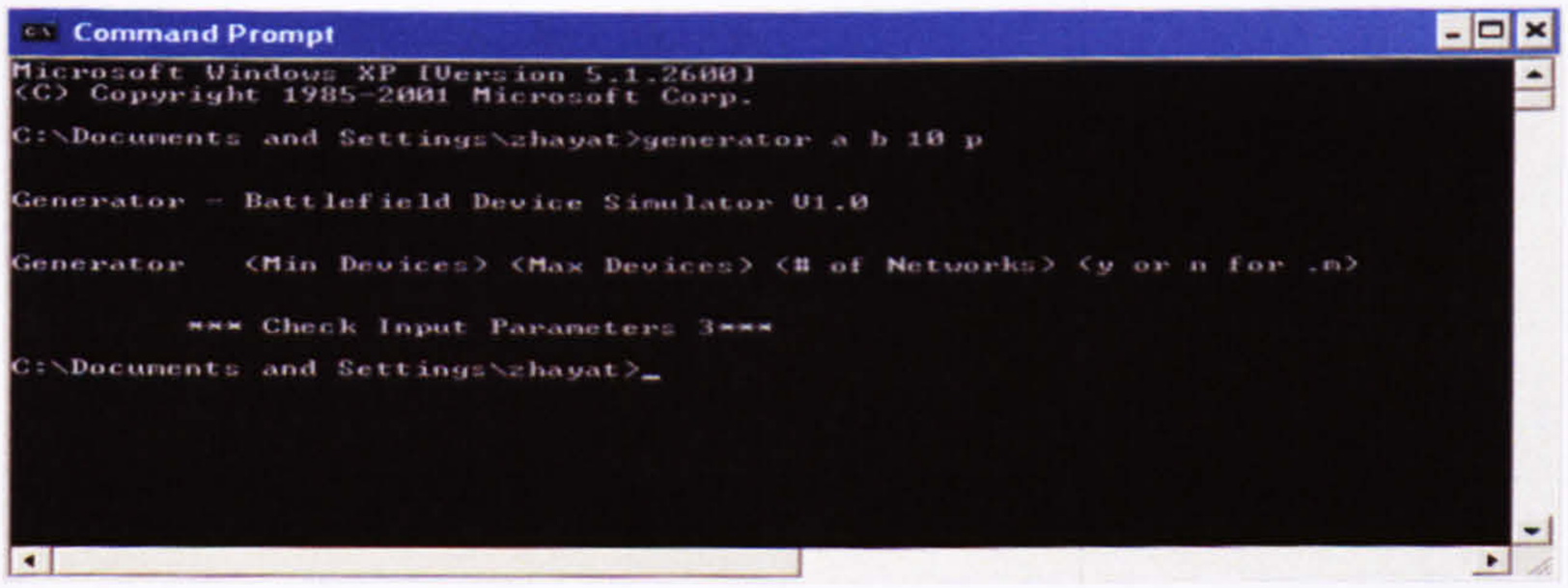

FIGURE A.7: Error in input to generator.

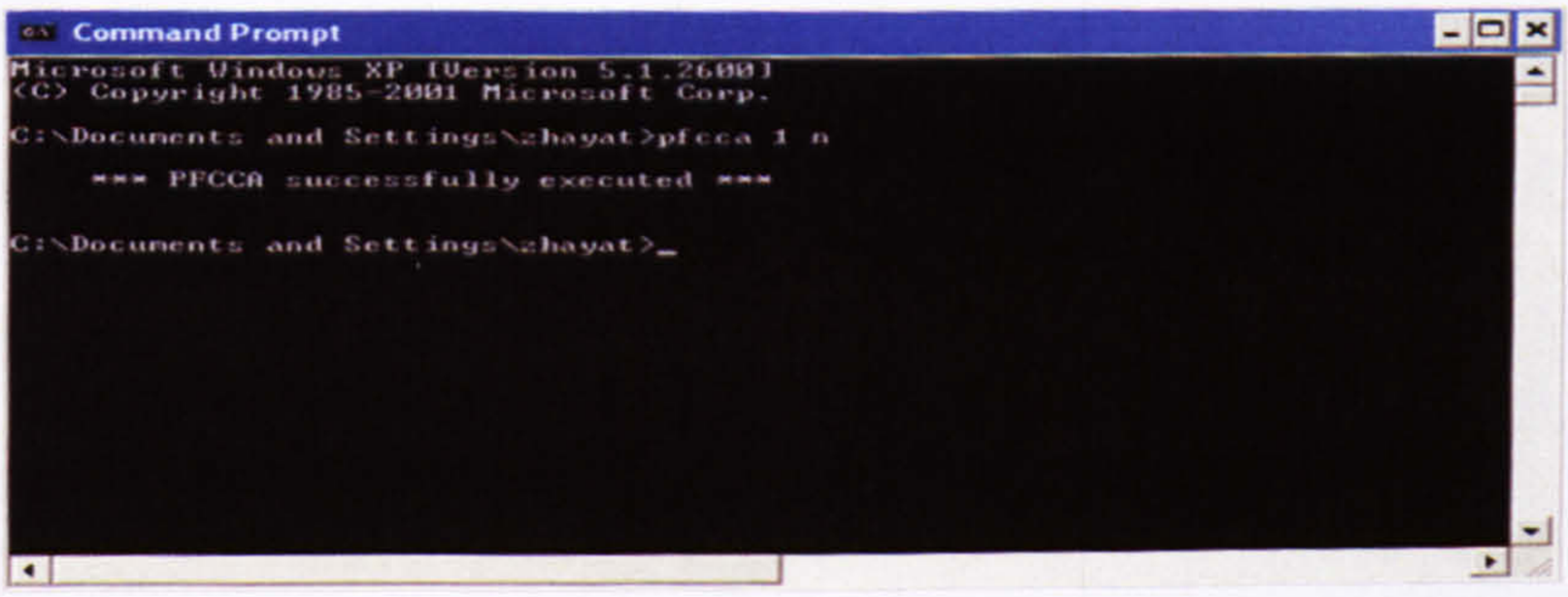

FIGURE A.8: Sufficient and correct input parameters to pfcca.

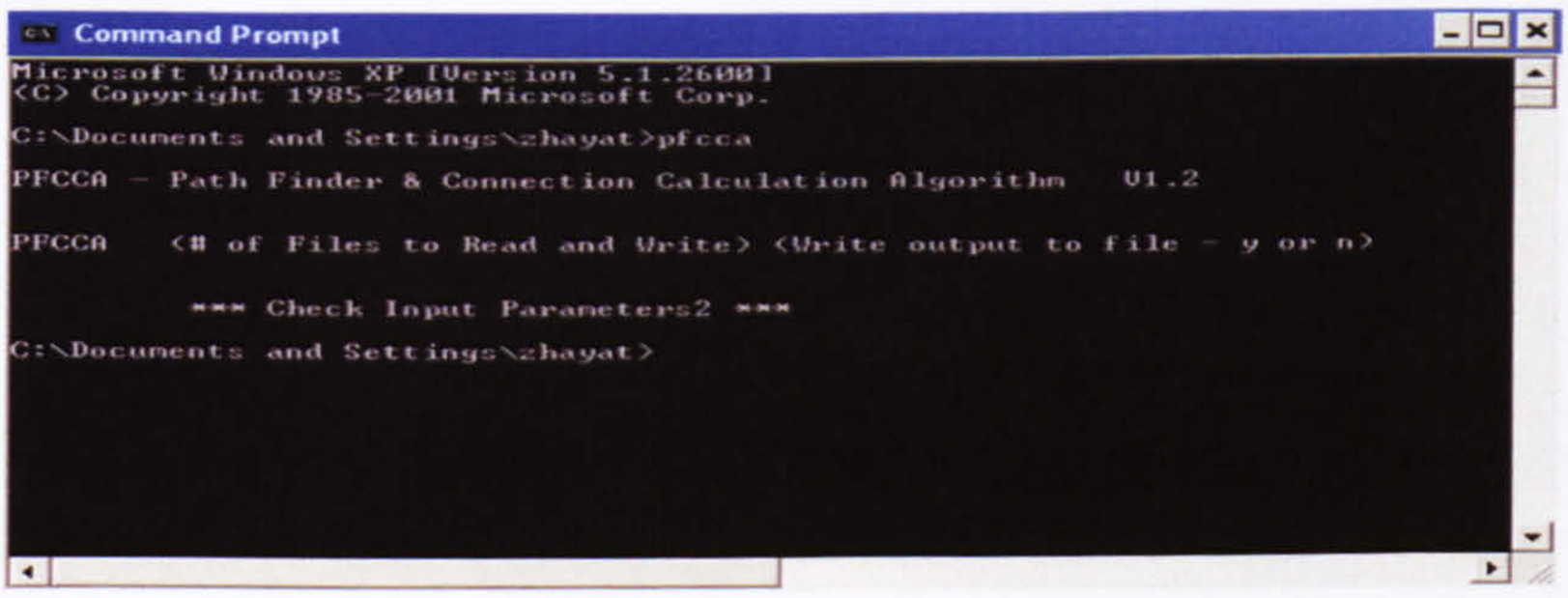

FIGURE A.9: No input parameters to pfcca.

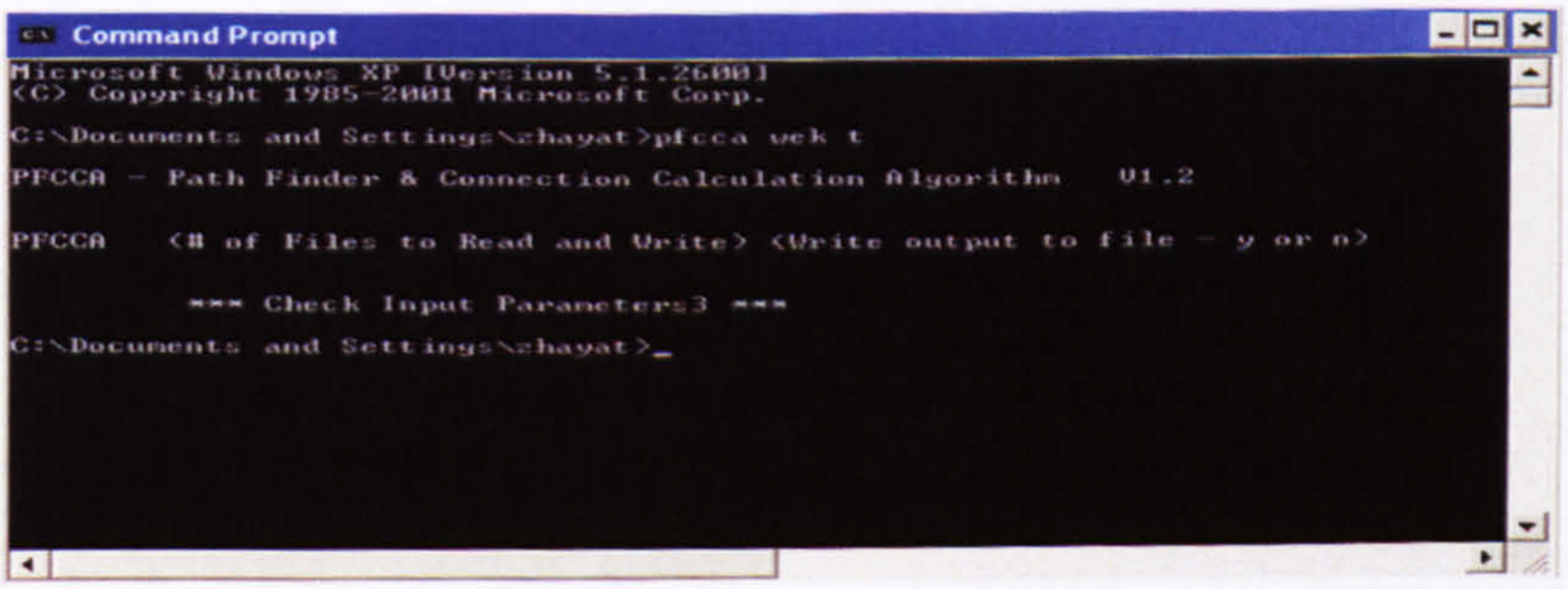

FIGURE A.10: Error in input to pfcca.

Blank Page

Appendix B

Distributed Security for Decentralised Information Sharing: Software Overview and Results

This section contains the class structures and use cases of the centralised and decentralised information sharing models. These models have been used to compare the benefits (and drawbacks) of centralised and decentralised security and therefore information sharing. The key differences between the two models are:

- All targeting information is reported to the effector via C2 in the centralised model, whereas it is reported directly from sensor to effector in the decentralised model.
- All sensor correlation activity is processed by the C2 agent in the centralised model, whereas in the decentralised model sensor observations are correlated by individual sensors.
- An additional delay is present in the decentralised model to represent security credential negotiation between sensors, effectors and C2 (trusted third party).

Detailed results (in .csv format) produced by the software can be found on the accompanying CD in the **distributed security** directory, under the the **DTE** and **ETS** subdirectories respectively. These results have been appropriately interpreted in order to compare the time taken for an effector to capture a target after the initial identification of the target by a sensor(s).

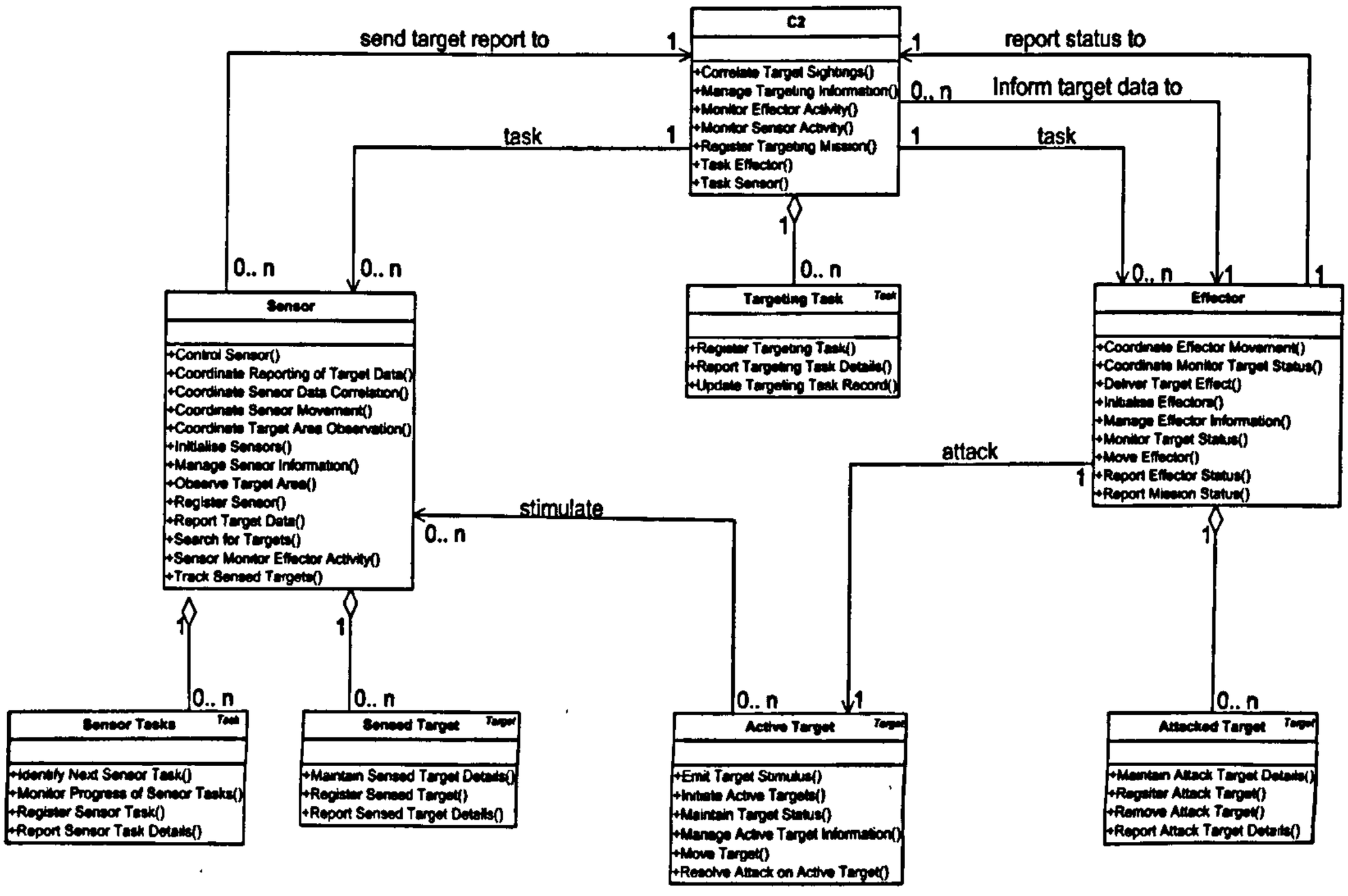

FIGURE B.1: UML diagram for class relationships in centralised model.

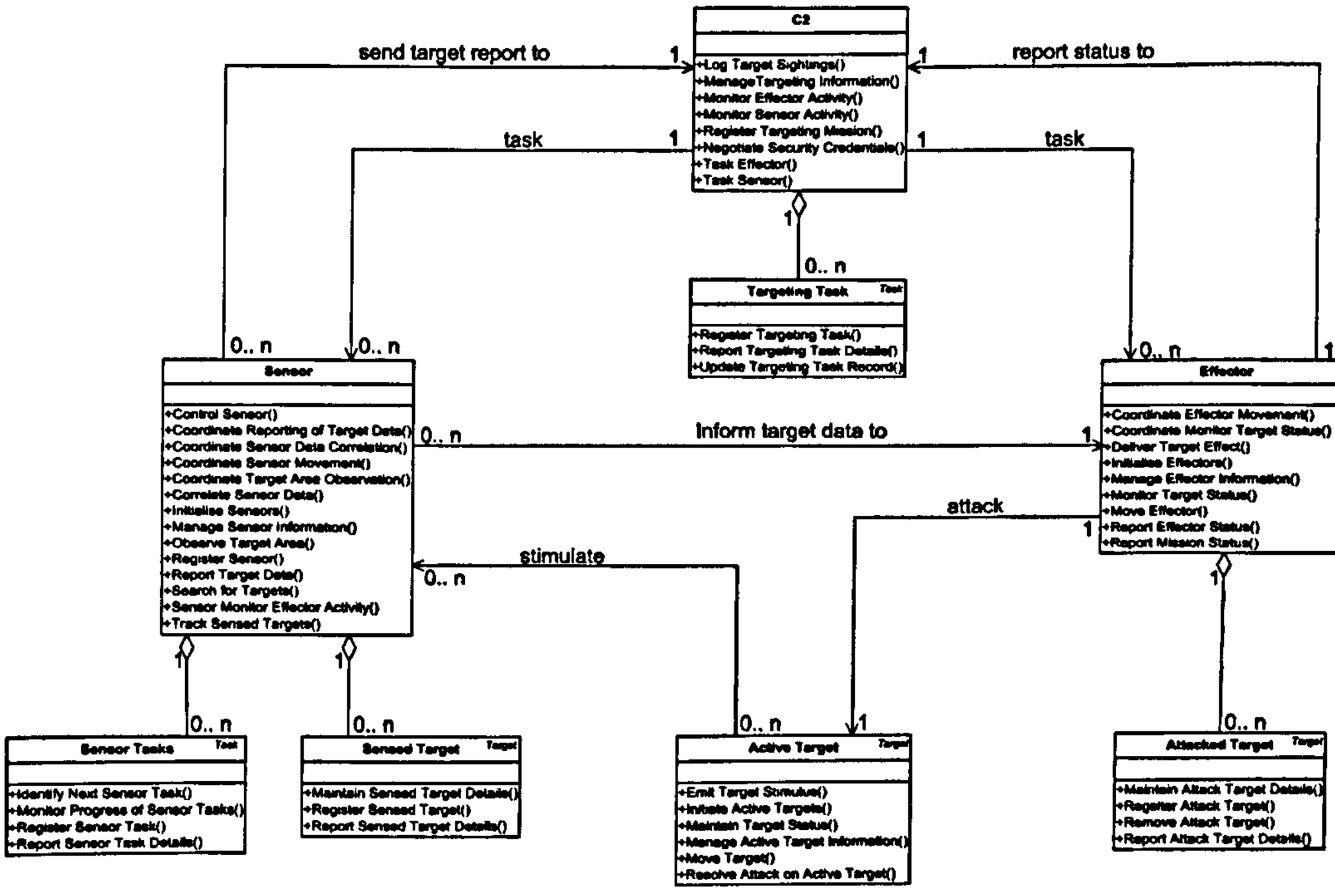

FIGURE B.2: UML diagram for class relationships in decentralised model.

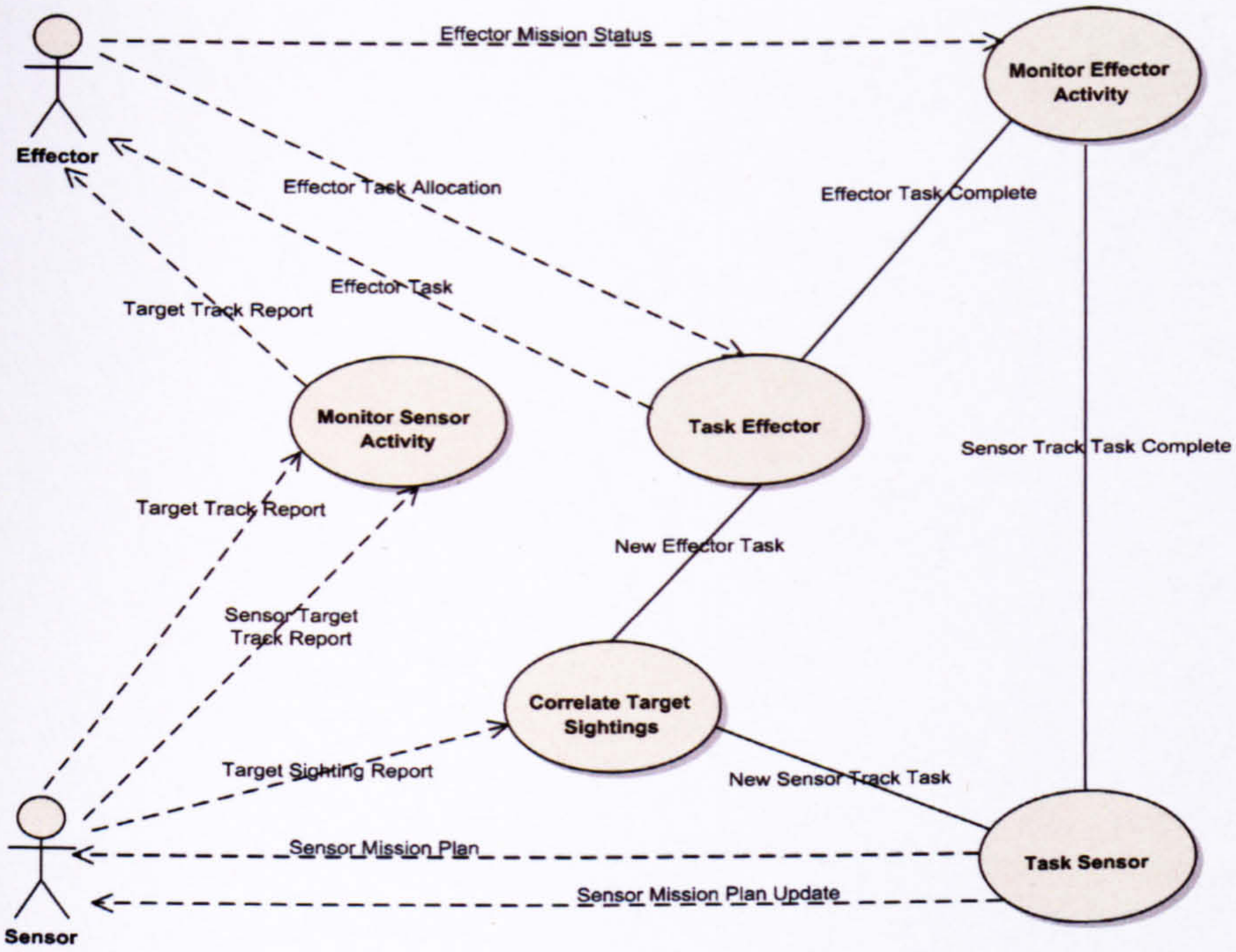

FIGURE B.3: Use case diagram for C2 agent in centralised model.

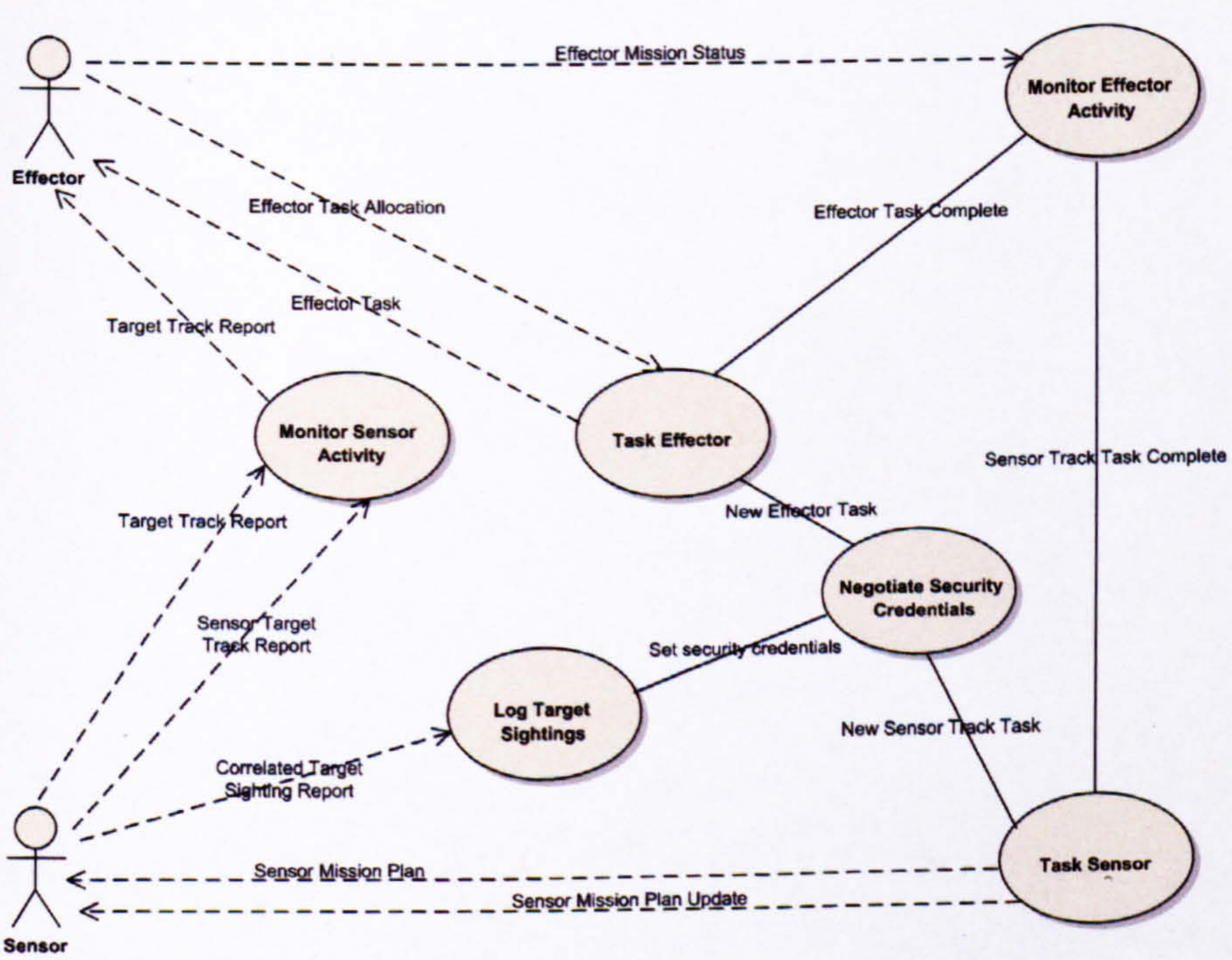

FIGURE B.4: Use case diagram for C2 agent in decentralised model.

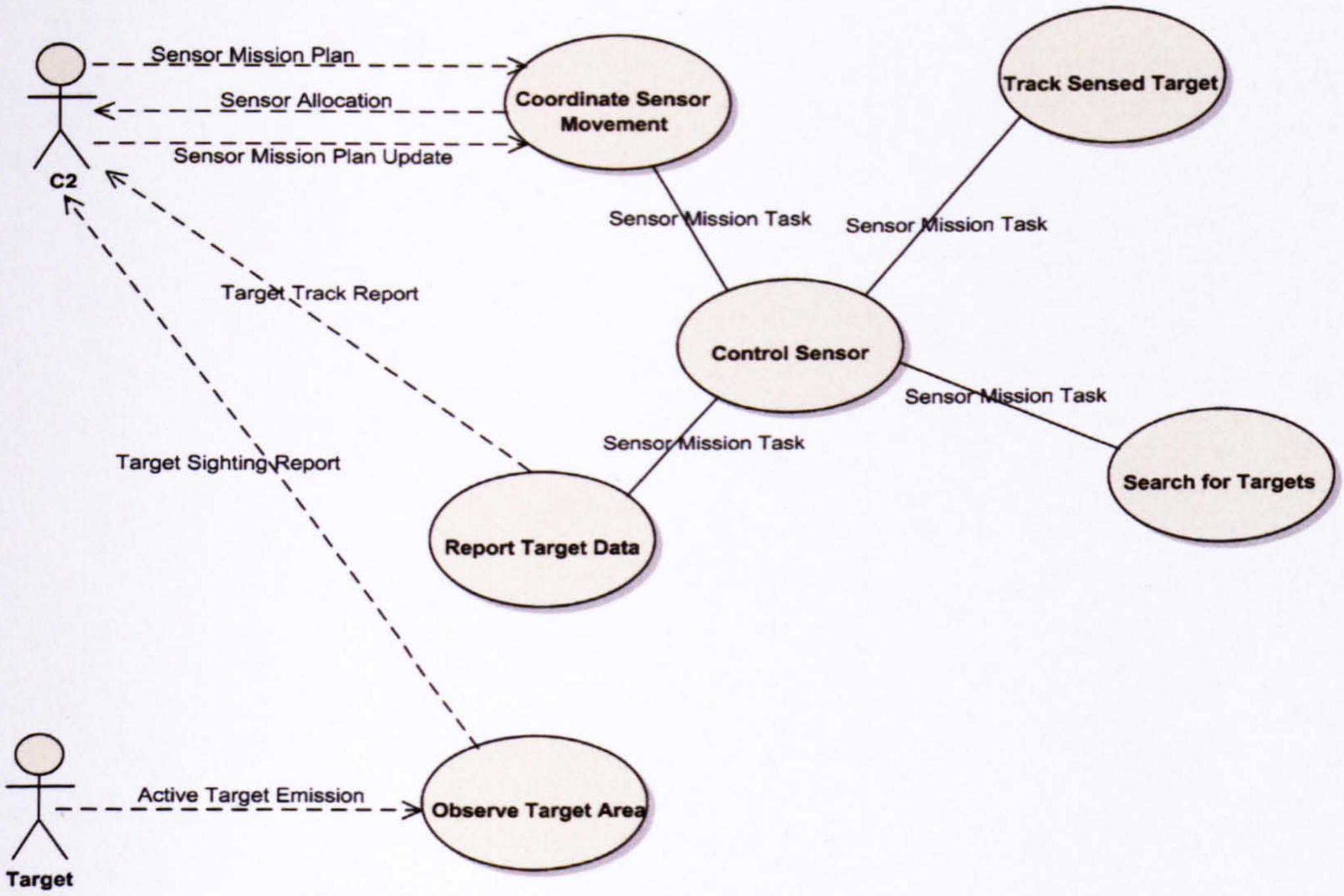

FIGURE B.5: Use case diagram for sensor agent in centralised model.

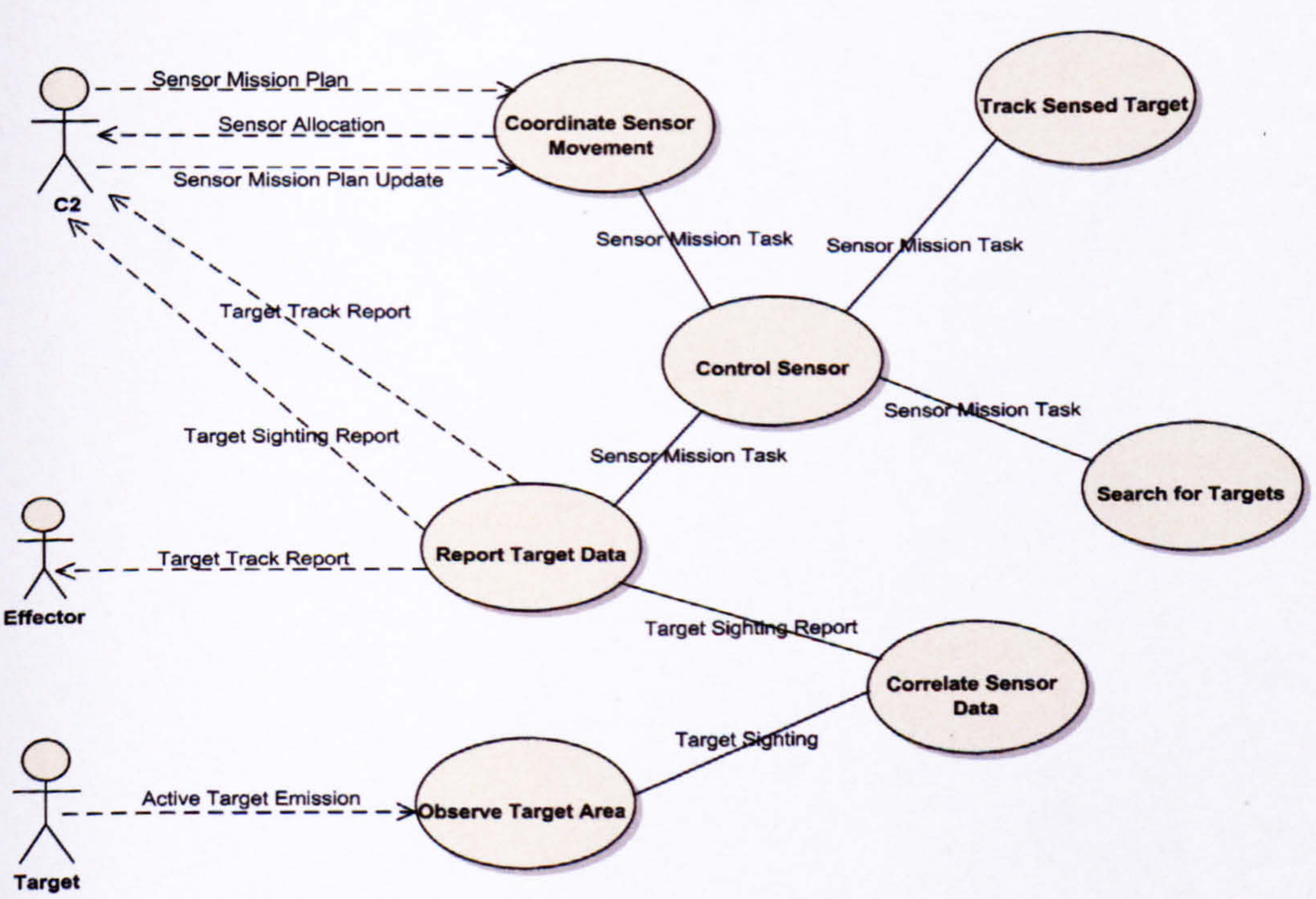

FIGURE B.6: Use case diagram for sensor agent in decentralised model.

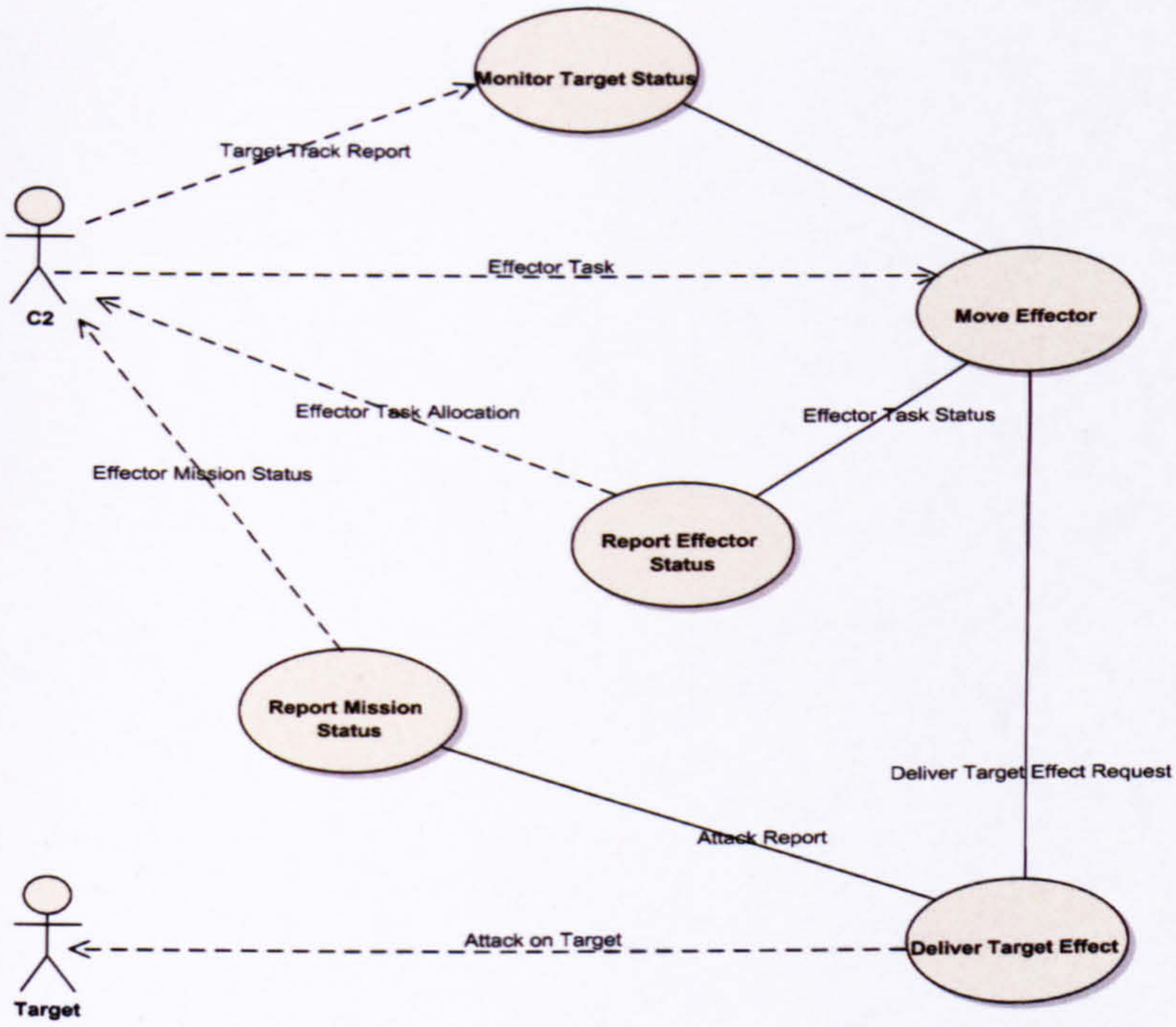

FIGURE B.7: Use case diagram for effector agent in centralised model.

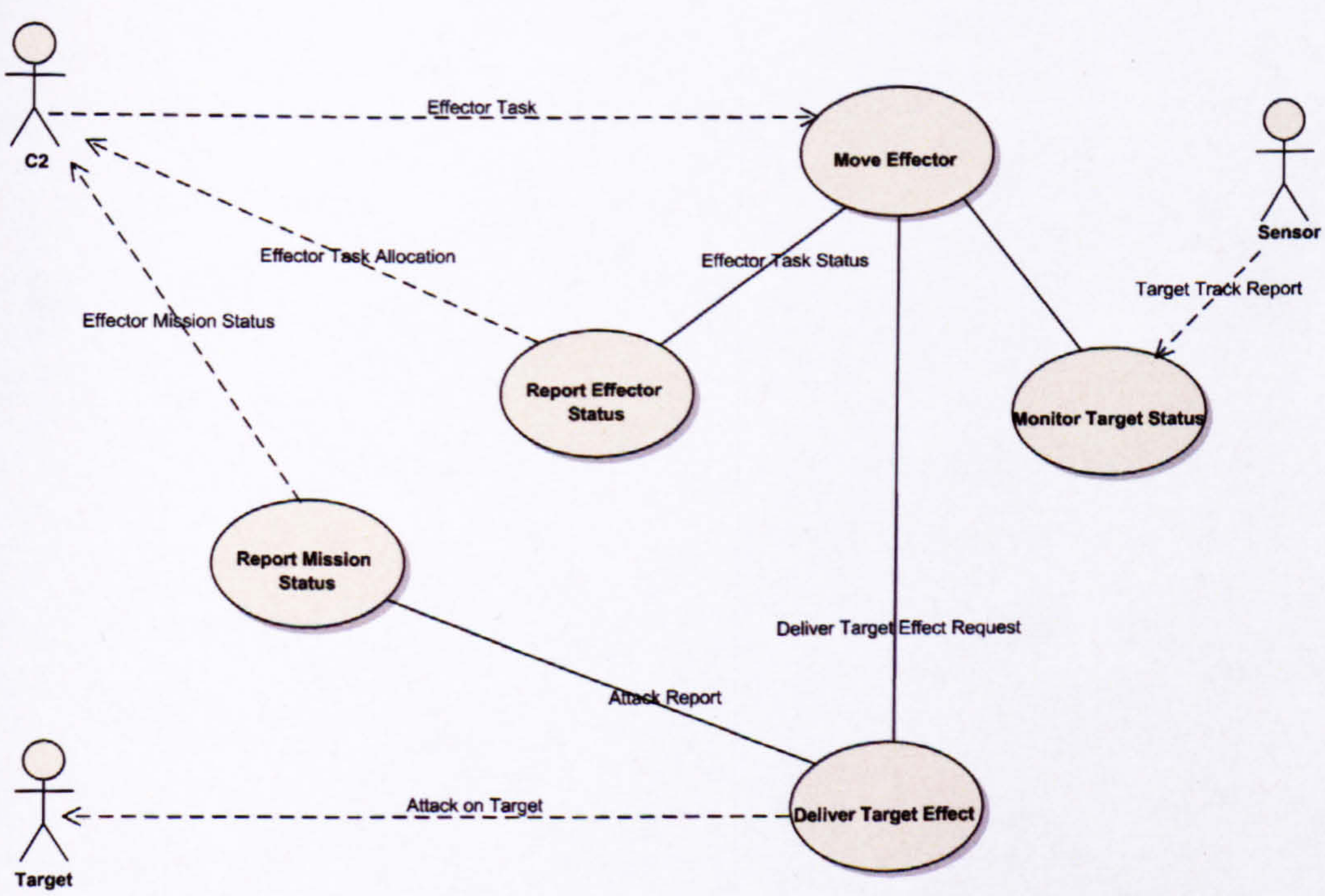

FIGURE B.8: Use case diagram for effector agent in decentralised model.

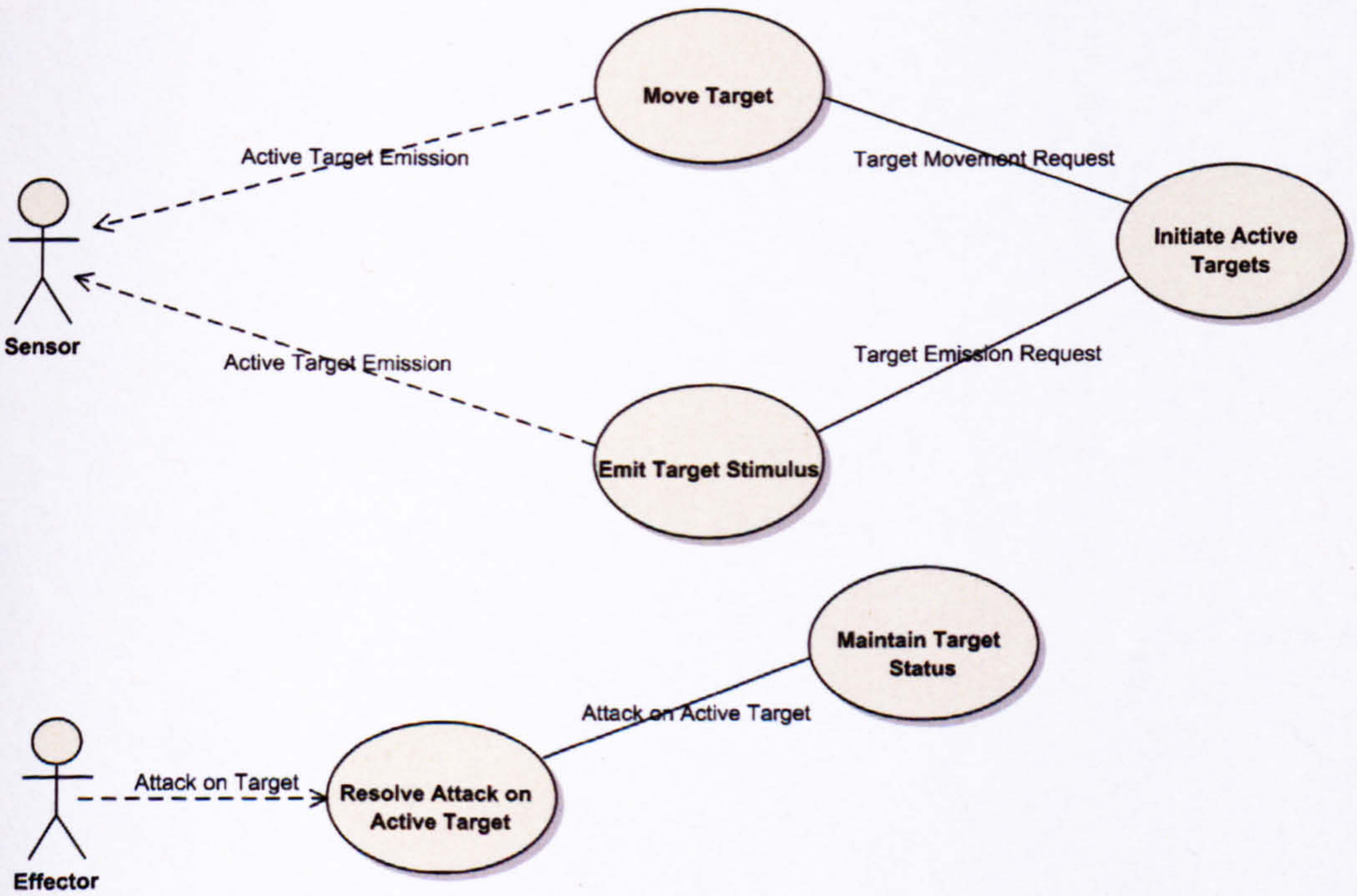

FIGURE B.9: Use case diagram for target agent in both centralised and decentralised models.

Appendix C

Industrial Activities

This section briefly describes projects and other industrial activities undertaken by the author, whilst completing the Engineering Doctorate, projects include:

- CVF - Automated a time consuming and error-prone threat analysis technique as part of the systems security team, saving time and cost.
- IFPA - Delivered part of the ADS, including DBSy model and infosec risk assessments.
- SEAS DTC - Outlined need to build security into the underlying systems architecture, scoped high-level security requirements, identified limitations in current security practices, offering alternative methods which leverage technology used in location based services.
- WiMAX Study - Assessed suitability of WiMAX for potential military operations, investigating current and planned future security properties.
- NRUC System - Principal security consultant in a cross industry consortium, advising on security risk assessment approach, regulatory issues and specific authentication mechanisms.
- Other - Worked as a security manager on other projects, and liaised with company head of IT security on various aspects of research.

Represented BAE Systems on various cross-industry steering and working groups including:

- MVCE security steering group.
- InnovITS technical working group.

- DTI cyber security KTN working group.
- CONSEQUENCE proposal technical working group.