Grant agreement no: 644729

European Commission

SAFE crypto

Secure Architectures of Future Emerging cryptography

Risk and Vulnerability Assessment of Lattice-based Cryptographic Architectures

Deliverable	D3.1
Author(s)	Bassem Ammar (HWC), Anthony Barnett (TRT), Andrew Byrne (EMC), Francesco Regazzoni (USI)
Version	1.0
Status	Approved
Date	22nd March 2016
Classification	☒ **White – public** ☐ **Green – restricted to consortium members** ☐ **Yellow – restricted to access list given below** ☐ **Red – Highly sensitive information, access list only**
Access List	

Queen's University Belfast CSIT Centre for Secure Information Technologies

HWCommunications Cyber Security and Resilience

Ruhr Universität Bochum RUB

RSA EMC²

THALES

Inria Inventeurs du monde numérique

Università della Svizzera italiana

Table of Contents

List of Figures

List of Tables

Glossary

AAA	Authentication, Authorisation and Accounting
ABE	Attribute based Encryption
AWS	Amazon web service
AVL	Automatic Vehicle Location
CA	Certification Authority
CESG	Communications Electronics Standards Group
CMOS	Complementary metal-oxide-semiconductor
COMSEC	Communications security
COTS	Commercial Off The Shelf
CPSC-CS	CoTS in Public Safety Communications Case Study
CRL	Certificate Revocation List
CSP	Cloud Service Provider
DEK	Data Encryption Key
DoS	Denial of Service
DPA	Differential power analysis
DTLS	Datagram Transport Layer Security
EM	Electromagnetic emanations
ETSI	European Telecommunications Standards Institute
GCHQ	Government Communications Head Quarters
GS	Ground Station
IBE	Identity based Encryption
KEK	Key Encryption Key
KIM	Key Injection Module
KMS	Key Management System
LBC	Lattice-based Cryptography
LIP	Location Information Protocol
NIST	National Institute of Standards and Technology
OCC	Operational Control Centre
OWASP	The Open Web Application Security Project
PII	Personally Identifiable Information
PKI	Public Key Infrastructure
PPDR	Public Protection and Disaster Relief
PPMDA-CS	Privacy Preserving Municipal Data Analytics Case Study

RR	Recovery or Repair
SKM-CS	Satellite Key Management Case Study
RSA	Rivest-Shamir-Adleman
SAAS	Software as a service
SLA	Service level agreement
SK	Secret Key
SKI	Secret Key Infrastructure
SPA	Simple Power Analysis
SSH	Secure socket shell
TC	Telecommand
TLS	Transport Layer Security
TM	Telemetry
TS	Threat source
UI	User Interface
USB	Universal Serial Bus
VM	Virtual machine
WebRTC	Web Real-Time Communications
WSN	Wireless sensor network

1 Introduction

1.1 Purpose

The purpose of this document is to provide an in depth analysis of the risks and vulnerabilities associated with lattice-based cryptographic architectures for the three case studies introduced in D9.1:

- Satellite Key Management
- CoTS in Public Safety
- Privacy Preserving Municipal Data Analytics

1.2 Scope

The scope of this document is to list the system assets and security vulnerabilities associated with the three case studies described in D9.1, and to identify and analyse threats against the proposed architectures. Traditional threats and vulnerabilities (i.e. attacks not involving the use of quantum computing) are considered with respect to the application of lattice-based cryptography in the case studies. Without loss of generality, this document focuses on:

- Security vulnerabilities, threats, and attacks associated with public-key cryptography, digital signatures and key exchange protocols.
- Impact of successful attacks on the three case studies that are considered in this project.

1.3 Structure of the Deliverable

Chapter 2 presents an overview of all threats and attacks relevant to the scope of the document. In Chapter 3, the definition of the terms used in calculating the risk is given, and the risk calculation method used is explained. Each case study is analysed in a separate chapter. Chapter 4 focuses on the Satellite Key Management case study, while Chapter 5 focuses on CoTS in Public Safety Communication, and Chapter 6 focuses on Privacy Preserving Municipal Data Analytics. For each case study, the system view and system components of the case study are summarised, the critical assets are identified, potential attack points are listed, and finally the risk analysis and countermeasures methods are presented.

2 Overview of Attacks/threats

The security of the critical assets depends, in part, on the strength of the cryptography used to protect the confidentiality and integrity of the data at rest and data in transit. The management of any cryptographic system requires robust key management to prevent an attacker from bypassing the cryptography entirely by gaining access to the keys. If a key is compromised (due to the failing of the protective mechanisms), the key can no longer be trusted to provide the required security and usage of the key to protect information should be limited to processing already protected data only. Physical side channel attacks may also compromise the secret keys and need to be considered when designing and implementing a secure architecture.

It is on these areas of cryptography, key management and side channel attacks that this section will focus its investigation of attacks and threats relevant to lattice based schemes. Subsequent sections will examine the specific challenges facing each of the use cases described in D9.1. The following sub-sections will provide an overview of the three primary classes of threats that are most relevant to SAFEcrypto.

2.1 Physical Side Channel Attacks

This section briefly introduces the problem of physical attacks to allow the reader to completely understand all the threats listed in the following section of this deliverable. An exhaustive description of physical attacks is reported in Deliverable 7.1, [36].

Cryptographic algorithms have to be physically implemented in order to be used in the real world. Typically, they are either implemented in hardware, in software or a combination of both. Unfortunately, in real world applications, the security challenges facing these implementations have evolved as new technologies and use cases emerge to incorporate algorithms and protocols that were (in some cases) designed decades ago. A rising concern is the ability of an adversary to tamper with physical devices and thereby gain knowledge of key material (potentially even the secret key itself) stored on the device. Furthermore, physical devices can and often do leak information through non-destructive physical observations such as the power consumption of the device or timing information gathered during some cryptographic operation involving the key. An attack that exploits the physical weaknesses of the implementation to get access to secret data is called a *physical attack*.

Typically, an adversary can achieve his/her goal in two ways: active or passive attacks [1]. During a passive attack, the attack is performed by observing and analysing physical quantities, such as power consumption, electromagnetic emission, or execution time. During active attacks, the adversary has to manipulate the device by modifying its inputs, its environment or both. The goal is to induce abnormal behaviour in the device and exploit this abnormal behaviour to perform the attack. The remainder of this section will discuss the most common physical attacks in more detail.

Timing analysis was the first form of side channel attack made public in 1996, [2]. Timing analysis attacks exploit the time needed by a device to perform a specific operation. The execution of different instructions to process data (e.g. encryption, decryption) on cryptographic devices result in slight variations in time, revealing information about some input to the instruction (the data or the key).

Such differences are due to many factors such as the time differences needed by the processor to execute two different instructions. For instance division is a significantly more complex, time intense operation than multiplication and therefore is executed over a longer period of time. Other factors affecting the discrepancies in execution time include the latency in fetching the data (cache or memory hit or miss), the algorithm behaviour as a result of branches and conditional statements, and finally the optimization that, for performance reasons, leads to skipping unnecessary operations.

In the particular case of a cryptographic device, the performance characteristics depend on both the secret key and the input data. Although intuition might suggest that unintentional timing characteristics leaked in that way would only reveal a small amount of information from the cryptographic device, the work of Kocher et al [2] presented an array of attacks which can exploit timing measurements from vulnerable systems to discover the entire secret key. Further works demonstrated the possibility of successfully mounting timing attacks on remote devices such as servers or virtual machines running on the cloud.

Power analysis attacks have been the subject of investigation for more than fifteen years [5]. Such attacks are possible because of the intrinsic characteristics of static CMOS, the technology used for the fabrication of almost the totality of modern chips. The instantaneous power consumption of a device strongly depends on both the data it processes and on the operation it performs. Power analysis attacks, the side-channel attack which has received the largest amount of attention from the scientific community, essentially exploit this fact and are an attractive attack vector due to the relative ease in performing them and their applicability to many commonly used cryptographic algorithms. The two most common types of power analysis attacks are distinguished as: *Simple Power Analysis* (SPA) and *Differential Power Analysis* (DPA) attacks.

In an SPA attack, an adversary basically attempts to derive the secret key using only a small set of power consumption traces (possibly only one, though multiple traces improves the accuracy of the attack). In these attacks, the secret key is inferred directly from the collected power traces. A possible target for SPA attacks are cryptographic devices in which the execution path depends on the secret key. For example, in the case of a software-implemented encryption algorithm that includes branches depending on the values of the secret key (such as some implementation of the square and multiply algorithm), there are instructions that occur only when part of the secret key has a specific value. As a result, by simply looking at the power trace and deriving the sequence of instructions performed, the attacker can guess the value of the key (or a part of it). Since their first discovery these attacks have evolved and been significantly improved upon. Template attacks [3] and collision attacks [4] are examples of new developments in SPA attacks.

From an attacker's perspective SPA attacks have a notable drawback: it requires knowledge of the internal details of the target implementation. [34]. More complex power analysis attacks, however, requires only the knowledge of the algorithm running on the target device. DPA attacks are a more advanced form of power analysis attack that are particularly successful as they are able to reveal the secret key without requiring particular knowledge about the device under attack (typically only knowledge of which cryptographic algorithm is being used in the device is sufficient). Furthermore, due to the particular analysis performed on the power traces, DPA may be successful even when the collected power traces are extremely noisy due to interference of other components on the device consuming power.

The major drawback of DPAs is the large number of samples needed to mount an effective attack. The collection of such a high number of samples can require a lot of time, and thus very often the attacker needs to possess the device under attack for an extended period of time. DPA attacks are based on a divide and conquer approach: the general idea is that the attacker selects a small portion of the key, makes a hypothesis on its value and verifies the hypothesis with the power traces. By iterating the same process, the full key can be recovered. Several possible ways have been proposed in the past, the most popular is based on the use of correlation coefficients by Brier et al [7]. To improve differential power analysis attacks, the adversary can target several intermediate points in the power traces, and verify the key hypothesis on all of them. These attacks are called high order differential power analysis attacks. As for simple power analysis, template attacks were also proposed for differential power analysis [8], [9], and [10] .

Exploitable leakage can also come from *electromagnetic emanations* (EM) from devices [6]. It has been extensively demonstrated in the literature that electromagnetic signals contain sufficient information to both break the security of cryptographic devices and to defeat several countermeasures against power analysis attacks. Electromagnetic attacks, contrary to power analysis attacks, do not require direct contact with the device to make the measurements, and allow a more precise positioning of the EM probe.

Concerning active attacks, the most common one is *fault injection* [11]. In these attacks, an adversary explicitly induces a fault into a circuit and exploits the erroneous behaviour to gain information about the secret key. The first step of a fault attack is the introduction of an error, typically transient, into the device. The error can be induced by varying the supply voltage, manipulating the clock, altering the temperature of the operating environment or exposing the device to laser or X-rays. Several successful fault attacks were proposed in literature, including ones targeting the AES and RSA algorithms [12], [13], and [14].

2.2 Logical threats

This section gives an overview of the logical threats that may affect an LBC implementation. Logical threats may target the system and/or software. In general, malicious attackers exploit vulnerabilities and threats that arise due to the use of insecure software components, unsuitable protocols or errors in software implementation. The most common vulnerabilities and threats are:

- Man-in-the-middle attacks – An attacker secretly intercepts and forwards communications between two parties who believe they are directly communicating with each other. In addition, the attack may alter the content of the communication compromising the integrity of the messages. This particular attack can come in many forms, and several types of attacks can be categorised as man-in-the-middle.
- Tampering – An attacker maliciously modifies data whilst it is in transit on a network.
- Spoofing – An attacker bypasses authentication functions using stolen or compromised passwords, tokens or keys to assume another valid user’s identity.
- Hijacking – An attacker breaks into an existing communications session and introduces their own stream of messages and data.
- Capture/replay – An attacker records a stream of data and then replays the same data to the server or application to repeat the effects. This may allow the attacker to access resources which are otherwise inaccessible without the necessary authorization. Without the appropriate cryptographic defences in place, this attack is straightforward to achieve for an attacker listening to network traffic by updating the packet sequence numbering.

Common software implementation errors, which can lead to undesirable outcomes, include

- Memory safety violations
  - Buffer overflows and over-reads – Writing and reading past the buffer’s boundary and into adjacent memory locations.
  - Dangling pointers – Pointers to memory that do not point to a valid object.
- Input validation errors
  - Format string attacks – The use of unchecked user input as the format string parameter, which may access memory locations or the call stack.
  - Code injection – Occurs when an application sends untrusted data to an interpreter, and changes the course of program execution or accesses unauthorized data.

  - Cross-site scripting – A type of injection attack that injects malicious scripts into trusted web sites. For instance, when an attacker uses a web application to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
  - HTTP header injection – Occurs when HTTP headers are dynamically generated based on user input. They can allow for malicious redirect attacks via the location header.
- Privilege-confusion bugs
  - Cross-site request forgery – Exploits the trust that a site has in a user's browser. CSRF is an attack that tricks the victim into submitting a malicious request. It inherits the identity and privileges of the victim to perform an undesired function on the attacker's behalf. For most sites, browser requests automatically include any credentials associated with the site, such as the user's session cookie, IP address, Windows domain credentials, and so forth. Therefore, if the user is currently authenticated to the site, the site will have no way to distinguish between the forged request sent by the attacker and a legitimate request sent by the victim.

The Open Web Application Security Project (OWASP) provides a list of common software vulnerabilities [28]. In addition to those already described above, this also includes:

- Injection – A code injection attacks occurs when an attacker sends malicious payload data to an application that is then executed as part of a command or query. The malicious payload can trick the application into executing unintended commands or accessing protected data.
- Broken authentication and session management - Functions pertaining to authentication and session management, such as password change or account updates, are sometimes implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.
- Using components with known vulnerabilities - If a component with a known vulnerability is exploited, an attacker can cause serious damage. Applications using components with known vulnerabilities may undermine application defences and enable a range of possible attacks and impacts. These attacks are usually made possible when end-users fail to update software with the necessary security patches or fixes.

These well-known vulnerabilities and threats must be taken into account when designing the lattice-based software architecture and key management schemes. These types of vulnerabilities can be guarded against in general, by using secure protocols and secure coding practices.

2.3 Human threats

This section provides an overview for the threats related to human intervention, whether it is accidental or malicious in nature. These threats are not specific to LBC and take a wider view of security that should be considered regardless of the cryptographic technologies used (including LBC). Human threats can be as a result of some malicious intent from an insider or external actor, or simply due to some misconfiguration or accidental violation of security policy.

For an attacker, the route chosen to attack an asset is typically the path of least resistance. Strong encryption algorithms, key sizes and access control mechanisms can be far easier bypassed by focusing effort on the human element of the security system. Common schemes for such attacks include phishing or tricking privileged users into following links to malicious sites or opening a malicious file in an email or URL from a website. While modern corporate firewalls and email scanners can detect the presence of malware preventing the user from exposing themselves to it, password protected files or hidden macros in documents can bypass some security measures. With this approach, even resource constrained attackers can launch an attack on individuals with access to the target asset (or enable the attacker to launch a subsequent attack on the asset).

For example, an employee within an organisation may be the victim of a spear phishing email attack which results in a key logging application to be launched silently. An attacker monitoring the employee's activity would soon be able to determine their logon credentials. With access to the employee's system and applications, the attacker now has a foothold inside the perimeter of an organisation's security defences. From here, the attacker can establish what other systems the compromised account has access to and may attempt to move laterally or vertically through the network until their objective is complete. Similarly, a piece of malware activated by a user inside the target environment may attempt to scan the internal network autonomously in order to identify administrative servers or credentials, opening another route for an attacker to access the management infrastructure.

Another attack vector is to physically breach the security boundaries by placing the assailant inside the network or system. With direct access to a server terminal, an attacker is better positioned to launch an attack. An employee's unlocked and unattended workstation is a perfect opportunity for an attacker to place themselves inside the network. Social engineering techniques can also be employed here to trick an employee to insert a USB drive containing malware (e.g. under the pretence of printing a document). Alternatively, with physical access to a site, key-logging devices can be installed (and later retrieved) in order to obtain credential information. A step further would be to steal devices containing the target data.

In order to minimise the risks of these kinds of scenarios resulting in an actual security breach, it is important that a proper security policy is developed and employees are trained in the effective implementation of those policies. These policies can include, but are not limited to:

- Restrict the usage of USB devices - Malware can be loaded to a system (and therefore the wider network) via USB drives. Anti-malware/virus software can reduce the risk of malware but are not a bulletproof solution as there can be a lag of up to several months between the creation of a piece of a malware and the updated signature in the anti-malware/virus database
- Restrict copying data from systems - To prevent data theft, systems should be locked down so that data cannot be copied to external USB devices or file sharing services such as Dropbox or Google Drive.
- Install security tools - Ensure anti-virus, anti-malware and full disk encryption software is installed and enforced. Anti-virus and anti-malware software can help reduce the risk associated with user interactions with emails and web browsers (by inadvertently clicking a link or downloading a file). Full disk encryption then prevents the risk to confidentiality of data in the event that a device is stolen.
- Enforce physical security polices - Physical security measures should be in place to restrict the movement of individuals through the organisation. Access to areas should be restricted based on the individual, their business unit and their role. For example, finance personnel have no requirement that they should need access to the physical server room of their organisation. Any visitors to a site should be properly registered, provided with a visitor badge with restricted access and be accompanied by a member of the organisation at all times. Similarly, policies should be in place for contract staff (e.g. security, cleaning, maintenance) that determine what they can and cannot access.

Other security best practices that should be implemented include ensuring that there is a segregation of duties across employees so that no single employee has the ability to execute an action and authorise it. This is of particular importance when dealing with IT administrative staff who might have access to every facet of an organisation's infrastructure. Role-based access control mechanisms can reduce the complexity of managing large databases of users with varying access levels across multiple business functions. This can reduce the probability of some misconfiguration in access rules when assigning a new user privileges or updating an existing user that may have changed role or left the organisation.

2.4 Threat sources and summary of threats

There are seven families of cybercrime [27] that can all be relevant threat sources in all three case studies. They are listed as

- Adolescent amateurs
    - Script kiddies
    - Hackers
- Amateurs with a goal
    - Avengers
    - Legal persons
- Resourceful professional
    - Organised crime
    - Terrorist
    - Spies

They differ in motivation and resources. Ideally, a thorough risk analysis is conducted with respect to each of the seven threat sources, where the following are considered

- Capability of threat source to carry a specific attack
- Motivation of the threat source to attack
- Presence of the threat source

Table 1 summarises relevant threats.

Type	Threat	Description
Physical (P)	Simple power analysis attack	Derive the secret key using only a small set of power consumption traces. The secret key is inferred directly from the collected power traces.
P	Differential power analysis attack	Derive the secret key using large number of samples, does not require knowledge about the device under attack
P	Timing analysis attack	Analyse the timing information gathered during the execution of some cryptographic operation to determine secret data.
P	Electromagnetic emanation	Exploit the leaked EM signals to break the security of cryptographic devices and to defeat several countermeasures effective against power analysis attacks.
P	Side Channel: Fault Injection	Explicitly induces a fault into a circuit and exploits the erroneous behaviour to gain information about the secret key.
P	Direct attack: Probing the hardware	Probing the hardware to recover the static private key or session keys from memory.
P	Reverse engineering	Reverse engineering hardware components and replacing with component under the control of the attacker.
P	Stealing	Stealing the hardware so that an attacker can use it themselves
Logical (L)	Brute force attack on session key	Enumerating all possible key values until the right value is found
L	Breach of the underlying lattice hard problem	Comes in the form of a mathematical proof, or a discovered algorithm that reduces the perceived hardness of the lattice-based problem

L	Breach of particular modifications in the lattice scheme Or Incorrect selection of lattice parameters	Comes in the form of discovered vulnerabilities in the optimization methods used to reduce the complexity of implementation of the lattice scheme.
L	Exploitation of integration vulnerabilities in the key management protocol	Vulnerabilities that arise out of introducing flaws whilst integrating lattice-based constructions into existing key management protocols that weren't originally designed for the lattice schemes.
L	Exploitation of implementation flaws	Flaws in the implementation of the scheme or associated protocols that enable an attacker to penetrate the system.
L	Man-in-the-middle	Attacker secretly intercepts and relays (in a potentially altered state) messages between two parties who believe they are directly communicating with each other. Several types of attacks discussed in this table can categorised as a one form of man-in-the-middle attack.
L	Software implementation errors: Memory safety Violations: Buffer overflows and over reads	Writing and reading past the buffer's boundary and into adjacent memory locations.
L	Software implementation errors: Memory safety Violations: Dangling pointers	Pointer to memory that does not point to a valid object.
L	Input validation errors: Format string attacks	The use of unchecked user input as the format string parameter, which may access memory locations or the call stack.
L	Input validation errors: Code injection	Use of malicious payload data sent to an application that is then executed and results in unintended commands or access to protected data.
L	Input validation errors: Cross-site scripting	Exploits the trust a user has for a particular site. It enables attackers to inject client-side scripts into web-pages viewed by other users.
L	Input validation errors: HTTP Header injection	Occurs when HTTP headers are dynamically generated based on user input. They can allow for malicious redirect attacks via the location header.
L	Privilege-confusion bugs : Cross-Site Request Forgery (CSRF)	Typically a logged-on user's browser is forced to send a forged HTTP request, including the victim's authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks

		are legitimate requests from the victim.
L	OWASP: Injection flaws	Injection flaws, occur when untrusted data is sent to the targeted interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorisation.
L	OWASP: Broken authentication and session management	Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.
L	OWASP: Using components with known vulnerabilities	Components with known vulnerabilities are exploited before they are patched or fixed.
L	OWASP: Missing Function Level Access Control	Many web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access functionality without proper authorization.
Human (H)	Phishing	Comes in many forms and attacks the most vulnerable point in the system, the human operator
H	Physical breach of security boundary	As described by the name
H	Misconfiguration or accidental violation of security policy	As described by the name

Table 1: Summary of Threats

3 Definitions and Risk calculation

In this chapter, we list the formal definitions of the terms used in risk calculation. The risk calculation method is then explained. The analysed threats may be in the form of single attack, or form part of an attack scenario (a sequence of attacks that are related to each other). The risk calculation method is adapted for both types.

3.1 Definitions

The objective of information security is to protect the owner's assets from attackers. Figure 1 illustrates the high level concepts and relationships between the assets and the threat agent and the threats, risks and vulnerabilities associated with them. Each component of the figure is described in more detail under their respective headings.

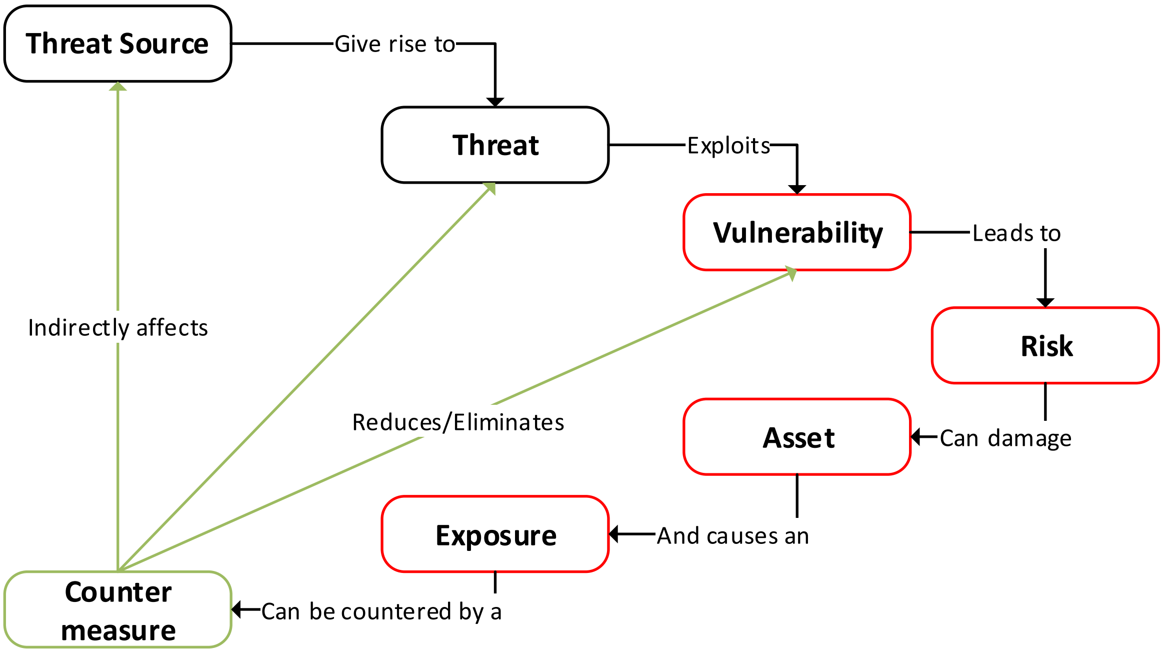

Figure 1: Asset, vulnerability & risk relationships

Asset

An asset is defined as anything that has value to the organization, its business operations and their continuity, including Information Resources that support the organisation's mission [31].

Generally speaking assets can be categorized into: hardware, data and service capabilities.

1. Hardware. This includes servers, devices, mobile phones, satellites, etc. The loss or damage of hardware, or hardware falling into the wrong hands, may affect the other two types of assets. Hardware can be used to mount an attack on crypto-related-assets for retrieving keys, obtaining/modifying cryptosystem configuration settings, etc.
2. Data. This includes the confidential data that is stored or communicated between application layers, meta-data, information about users such as their location, their identities, etc. It also includes information that if acquired in plaintext may lead to decrypting the application layer information, such as keys, system configuration, information obtained from side channel observations and attacks.

3. Service capabilities. This is the ability to provide the service itself. For example, in the COTS use case for public safety, this is the ability to communicate information between system entities and users.

It can be noted that a loss of some asset may affect other assets as well. The scope of the assets considered encompasses the assets that are protected and affected by the use of lattice-based cryptography.

Critical system assets

These are assets that, if successfully attacked or compromised, could potentially have a serious impact or consequence on the system, such as performance degradation, data leaks or total failure. Criticality is defined as a measure of the degree to which an organization depends on the information or information system for the success of a mission or of a business function [32].

Attack

Any kind of malicious activity that attempts to collect, disrupt, deny, degrade, or destroy information system resources or the information itself [18].

Compromise

Disclosure of information to unauthorised persons, or a violation of the security policy of a system in which unauthorised intentional or unintentional disclosure, modification, destruction, or loss of an object may have occurred [29].

Impact

The effect on organizational operations, organizational assets, individuals, other organizations, or the nation of a loss of confidentiality, integrity, or availability of information or an information system [18].

The **impact level** is the magnitude of harm that can be expected to result from the consequences of unauthorized disclosure of information, unauthorized modification of information, unauthorized destruction of information, or loss of information or information system availability [18].

Likelihood of occurrence

A weighted factor based on a subjective analysis of the probability that a given threat is capable of exploiting a given vulnerability or a set of vulnerabilities [18] [20].

Risk

A measure of the extent to which an entity is threatened by a potential circumstance or event, and typically a function of: (i) the adverse impacts that would arise if the circumstance or event occurs; and (ii) the likelihood of occurrence [19].

Information Security Risk

The risk to organizational operations (including mission, functions, image, reputation), organizational assets, individuals, other organizations, and the nation due to the potential for unauthorized access, use, disclosure, disruption, modification, or destruction of information and/or information systems.

Information System-Related Security Risk.

Risk that arises through the loss of confidentiality, integrity, or availability of information or information systems and reflect the potential adverse impacts to organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, and the nation. Adverse impacts to the nation include, for example, compromises to information systems that support critical infrastructure applications or are paramount to government continuity of operations as defined by the Department of Homeland Security [18] [20].

Threat

Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service. Also, the potential for a threat-source to successfully exploit a particular information system vulnerability [18].

Threat Scenario

A set of discrete threat events, associated with a specific threat source or multiple threat sources, partially ordered in time.

Threat Source

The intent and method targeted at the intentional exploitation of a vulnerability or a situation and method that may accidentally exploit a vulnerability [18].

Vulnerability

Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited by a threat source [18].

3.2 Risk factors

The *risk calculation* may be given as a function, *f*, of the product of threat score, vulnerability score and impact score. A threat score is typically a function of the capability, the presence and the motivation of the attacker.

Therefore the *Risk* for a particular incident may be written as:

$$Risk = f(\,Vulnerability, Capability, Motivation, Presence, Incident\ impact)$$

For the purpose and scope of this project, we will ignore the *Motivation*, and *Presence* and will therefore calculate the *Risk* of a particular threat source to carry a particular threat on a particular system as

$$Risk = f(Capability * Vulnerability * Impact)$$

$$Risk = {V * C * K}/{V_{max} * C_{max} * K_{max}}$$

where *C* is the capability score, which represents the technical capability of the threat source to carry a particular attack. *V* is the vulnerability score, which is a measure of the extent to which the system is prone to a particular type of attack. *K* is the impact score, which is a measure of the negative consequences of a successful attack on the system under consideration.

Table 2 and Table 3 show the capability score and the vulnerability score descriptions, respectively. Each has a value range from 1 to 5. Therefore, we have C_{max} = 5 and V_{max} = 5.

Capability Score	*Capability Level*	*Description*
5	Very High	Threat source (T.S.) is currently judged to have "full capability" to carry out this scenario
4	High	T.S. is currently judged to have "capability in the majority of areas and could meet any additional requirements in the short-term“ to carry out this scenario
3	Moderate	T.S. is currently judged to have "capability in some areas but will need to acquire significant additional capability which will take time to do“ to carry out this scenario
2	Low	T.S. is currently judged to not have the "necessary capability but might be able to acquire it in the medium-to-long term” to carry out this scenario
1	Very Low	T.S. is currently judged to not have the necessary capability nor the ability to acquire it in the foreseeable future

Table 2: Capability Score description

Vulnerability Score	*Vulnerability Level*	*Description*
5	Very High	There is "no capability" to prevent this scenario from occurring and causing worst case impacts
4	High	There is "very limited capability" to prevent this scenario from occurring and causing worst case impacts
3	Moderate	There is "moderate capability" to prevent this scenario from occurring and causing worst case impacts
2	Low	There is "significant capability" to prevent this scenario from occurring and causing worst case impacts
1	Very Low	There is "high degree of capability" to prevent this scenario from occurring and causing worst case impacts

Table 3: Vulnerability Score description

Table 4 shows the impact score description. This is dependent on the case study and its business case. Here we take the example of the COTS case study and define the impact of an impeded rescue operation in relation to financial consequences, effect on business contract, the loss of customer trust and the recovery effort. However, in many cases the impact of a successful attack on a case study may not be clear enough to assign a score for it. This might be the case, when the attack consists of a series of consequential attacks forming a threat scenario. This will be addressed in section 3.3.

Score	Operational Level -(COTS case study)	Financial Impact	Contract	Customer trust	Recover Effort
100	Rescue operation is completely obstructed	Company Bankrupt	Invalidates existing contracts	Lost by 100%.	Prohibitively expensive and difficult to recover or repair (RR)
90	Rescue operation is impeded by 90%	> 50 % loss of company value	Invalidates existing contracts	Lost by 90%	Extremely difficult or costly to RR.
80	Rescue operation is impeded by 80%	> 20 % loss of company value	No foreseeable renewing contracts	Lost by 70%	Very difficult or costly to RR.
70	Rescue operation is impeded by 70%	> 5 % loss of company value	affects contracts renewal by > 50%	Lost by 60%	Very difficult or costly to RR.
60	Rescue operation is impeded by 60%	> 1 % loss of company value	affects contracts renewal by >30%	Lost by 50%	Difficult or costly to RR.
50	Rescue operation is impeded by 30%	> 0.5 % loss of company value	affects contracts renewal by >20%	Lost by 30%	Moderately difficult to RR.
40	Rescue operation is impeded by 15%	> 0.1 % loss of company value	affects contracts renewal by > 10%	Lost by 15%	Possible to RR.
30	Rescue operation is impeded by 10%	> 0.05 % loss of company value	affects contracts renewal by > 1%	Lost by 10%	Costs for RR. are not significant
20	Rescue operation is slightly impeded	> 0.001 % loss of company value	affects contracts renewal by > 0.01%	Causes concern	RR.is straightforward
10	Rescue operation is not significantly impeded	> 0.0001 % loss of company value	No effect on contracts renewal	No concern	RR. is simple and easy

Table 4: Impact Score description

3.3 Impact and Risk calculation in threat scenarios

Quite often the full impact of a particular attack can only be appreciated and measured by considering the potential threats, which may occur if that attack is successful. In this respect, a complete risk analysis of a threat would require analysing potential threat scenarios, which are dependent on that threat.

Moreover, a threat can be broken down into a threat scenario with component "sub-threats". For example, a social engineering attack in the form of a phishing email, may lead to installing key-logging malware, which may lead to accessing sensitive data including passwords and financial data, which could potentially lead to an impersonation attack, financial loss, extortion, blackmail, etc. In analysing the above threat scenario, the impact of the phishing email attack is measured based on the impact of the potential subsequent attacks taking into account the capability required to get through each stage of the attack scenario and vulnerability of the system to each attack.

A threat scenario typically consists of a series of consequential threats.

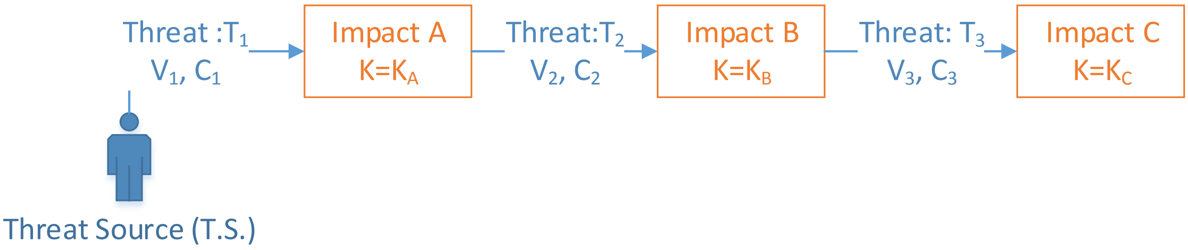

Figure 2: Impact calculation example-1

Figure 2 shows a generic template describing a threat scenario carried out by a threat source. A blue arrow represents an attack, and an orange box represents the impact of the threat. The threat scenario in the figure describes a series of three threats (T_1, T_2 and T_3). Threat T_1, has an impact A, with an impact score K_A, and may lead to the mounting of further threats T_2 and then T_3. For each threat T_i there is a vulnerability score V_i and a Capability score C_i. The overall impact of the threat scenario may be easily determined by the final impact stage, C in Figure 2. To calculate the impact at intermediate stages A and B with respect to C we first calculate the impact score at B, K_B, as a function in K_c, then we calculate the impact at A.

- Impact score at stage B is given by $K_B = K_C V_3C_3/V_{max}C_{max}$
- Impact score at stage A is given by $K_A = K_B V_2C_2/V_{max}C_{max} = K_C V_2C_2V_3C_3/(V_{max}C_{max})^2$

To calculate the risk for the threat scenario that goes from A to B to C and consists of T_1, T_2 and T_3, we use the function *f* described in Section 3.2 to get

$$R(A,B,C) = \frac{K_C V_1 C_1 V_2 C_2 V_3 C_3}{(V_{max}C_{max})^3 K_{max}}$$

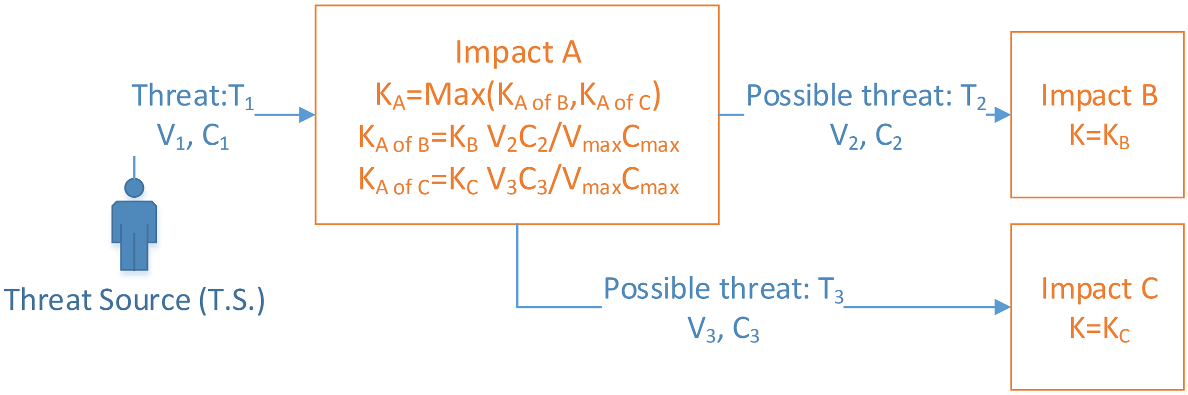

Figure 3: Impact calculation example-2

Figure 3 shows how impact score is calculated at one stage when there may be more than one threat, leading to more than one possible impact, emanating from one point. For example the impact score at A, K_A, is calculated with respect to the impact scores at B, K_B, and C, K_C.

- Impact score at stage A is K_A is given by Max ($K_{A\ of\ C}$, $K_{A\ of\ B}$)
    - ○ $K_{A\ of\ C} = K_C\ V_3C_3/V_{max}C_{max}$
    - ○ $K_{A\ of\ B} = K_B\ V_2C_2/V_{max}C_{max}$

To calculate the risk for the scenario in Figure 3, we calculate the risk of the threat scenarios 'A to B' and 'A to C' as follows

$$R(A,B) = K_B V_1 C_1 V_2 C_2 / (V_{max} C_{max})^2 K_{max}$$

$$R(A,C) = K_C V_1 C_1 V_3 C_3 / (V_{max} C_{max})^2 K_{max}$$

The risk is determined by the maximum value of any of the possible risks as

$$R_{max} = Max\ \big(R(A,B), R(A,C)\big)$$

4 Satellite Key Management

4.1 High level view

4.1.1 System view

The satellite communications case study is primarily concerned with the management of keys required to protect communications between satellites and the *Operational Command Centre*(OCC). The scenario is restricted to the use-case where the keys primarily protect *telecommand traffic* travelling up to the satellite, via one or more *Ground Stations* (GS). In addition, housekeeping telemetry data coming from the satellite may need protection. Protection could be at the Network, Transport or Application Layers.

A potential key management architecture for the management of satellites is shown in Figure 4

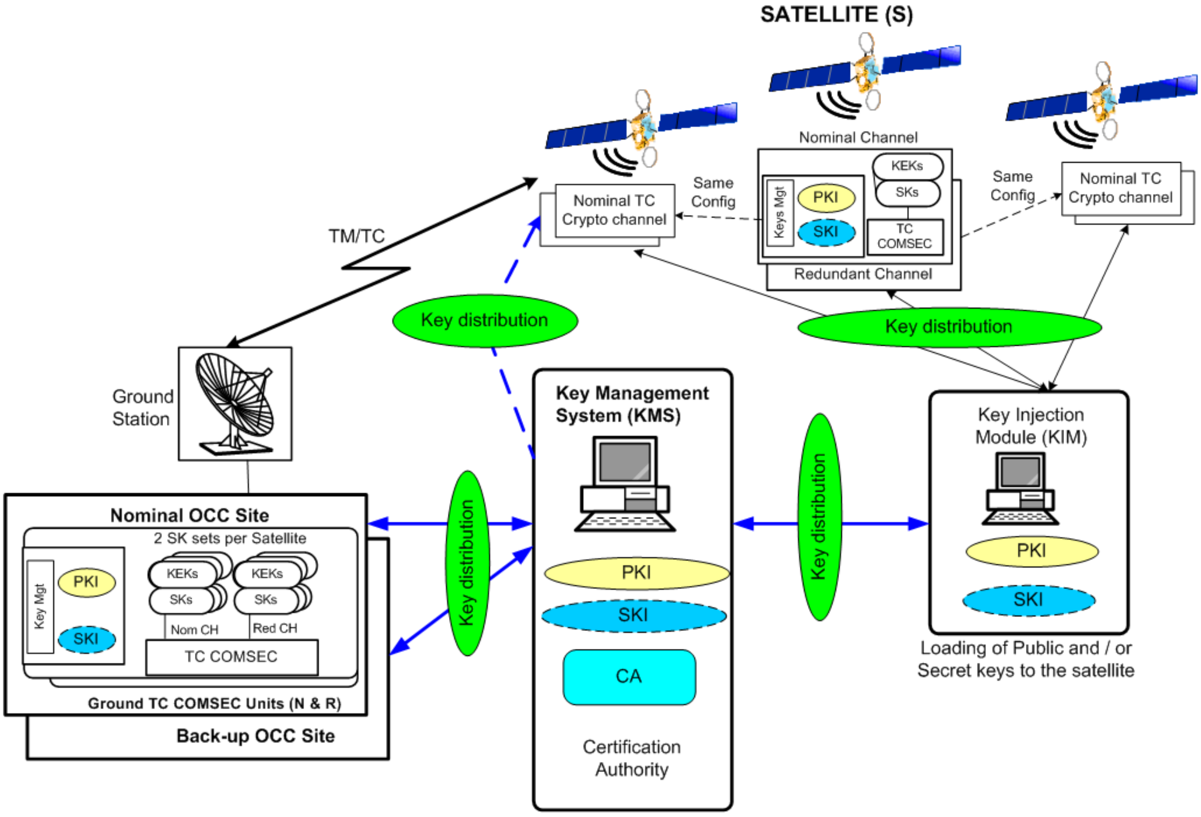

Figure 4: Overview of the satellite scenario use-case.

4.1.2 Components

The primary components in the Telecommand Channel Key Establishment use-case are the

- Operational Control Centre (OCC) - The entity that ultimately controls the satellite, via the so-called “Telecommand Channel”. It also receives so-called “housekeeping” telemetry data from the satellite.
- Satellite - The spacecraft that acquires data and transmits it back to the ground.

The secondary components are:

- Key Management System (KMS) – The entity that provides the key material, certificates, certificate revocation lists (CRLs) and algorithm parameters required by the Key Establishment protocol.

- Key Injection Module (KIM) – This entity can be viewed as a subset of the KMS, responsible for loading initial key material, certificates and algorithm parameters onto the satellite.
- Core Ground Station – This entity simply relays communications between the OCC and the Satellites.

4.2 Critical system assets

4.2.1 Assets Description

1. Satellite
2. Operational Command Centre – This includes the hardware on which the key management systems is running.
3. Ground Station – The hardware transmitting and receiving the communications to and from the satellite.
4. Application layer data - This is the data sent between the satellite and the ground segment. The data consists of:
    a. Telecommand data from the OCC to the satellite.
    b. Housekeeping telemetry from the satellite to the OCC.
5. Key Management System
    a. Session keys - These are the symmetric keys that encrypt the application layer data.
    b. Session key generation function and inputs - This is the function used to generate the session key. Inputs include the source of entropy (generation of random numbers).
    c. PKI private keys
    d. Public/Private key pair generation function and inputs - This is the function used to generate the public/private key pair. Inputs include the source of entropy (generation of random numbers).
    e. Key management protocols – These are yet to be determined in detail for the satellite use-case.
    f. PKI Root Certificate
    g. Certification Authority (CA) – An entity that issues digital certificates.
6. Key Injection Module - This is used to load the key material onto the satellite prior to launch.

4.2.2 Assets summary

Reference	Asset
CS1_A_1	Satellite
CS1_ A_2	The Operational Command Centre
CS1_ A_3	The Ground Station
CS1_ A_4.1	Application layer data - Telecommand data
CS1_ A_4.2	Application layer data - Housekeeping telemetry

CS1_ A_5.1	Key Management System - Session keys
CS1_A_5.2	Key Management System - Session keys generation function
CS1_A_5.3	Key Management System - PKI private keys
CS1_A_5.4	Key Management System - Public/Private key pair generation function
CS1_A_5.5	Key Management System - Key management protocols
CS1_A_5.6	Key Management System - PKI Root Certificate
CS1_A_5.7	Key Management System - Certification Authority
CS1_A_6	Key Injection Module

Table 5: Summary of Case Study 1 assets

4.2.3 Asset dependency diagrams

In this section, the interdependencies between assets are shown.

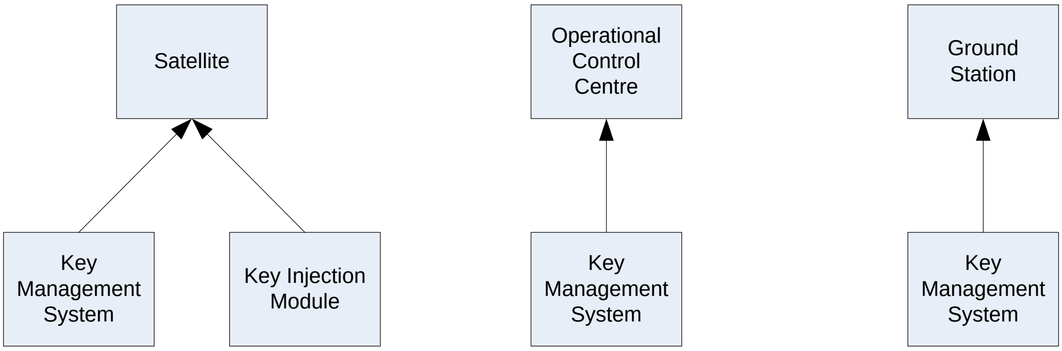

Figure 5: The asset dependencies of the system hardware components.

In Figure 5, we can see that the security of the satellite depends on the Key Management System operating on both the satellite itself and on the ground segments. It also depends on the Key Injection Module, which loads key material onto the satellite before launch. Likewise, the security of the Ground Segments (the OCC and the Ground Station) are also dependent on the Key Management System.

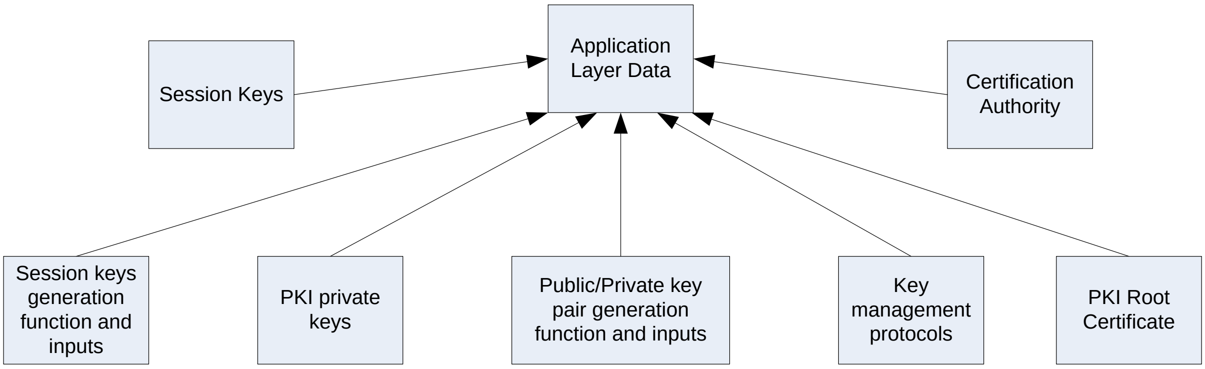

Figure 6: The asset dependencies of the Application Layer Data.

In our satellite scenario, the telecommand data, from the OCC to the Satellite and the housekeeping data from the Satellite to the OCC together form the application layer data. The security of the entire Satellite system depends on the security of this application layer data. This data depends on the individual components of the Key Management System, as illustrated in Figure 6. (i.e. the keys, the key generation functions and inputs, the key management protocols and the PKI infrastructure).

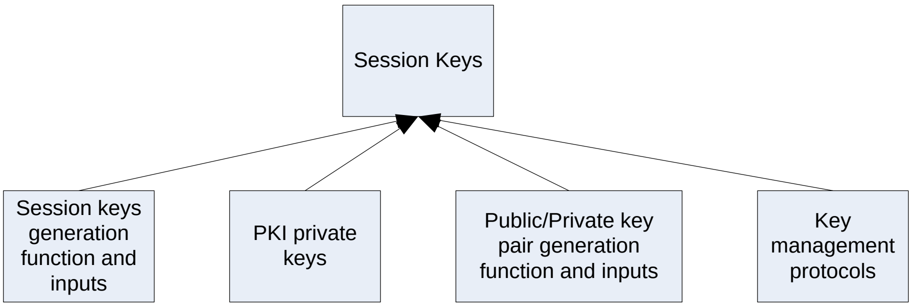
Session Keys
Session keys generation function and inputs
PKI private keys
Public/Private key pair generation function and inputs
Key management protocols

Figure 7: The dependencies of the session keys.

The session keys themselves depend on the generation functions (i.e. the lattice-based algorithms) and random number inputs, the PKI private keys and the functions and random number inputs to generate the public/private key pairs. The key management protocols depend on the PKI elements of the CA and the root certificate. These dependencies are illustrated in Figure 7 and Figure 8.

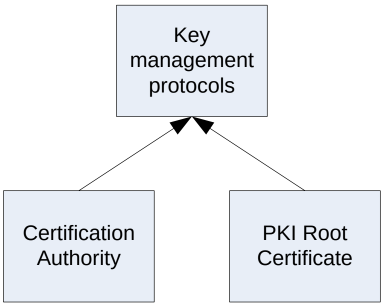

Figure 8: The dependencies of the Key Management Protocols.

4.3 Attack points

4.3.1 System Hardware

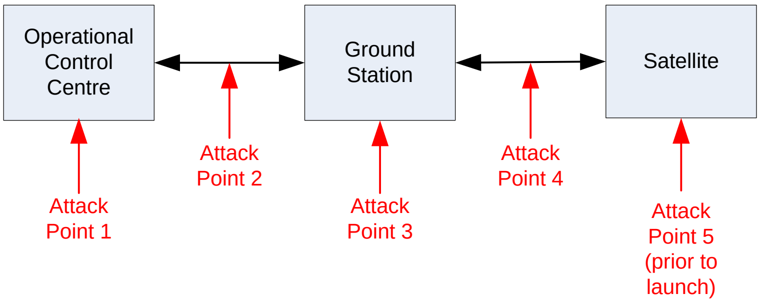

Figure 9: Potential attack points in the system hardware and data links.

The hardware in the Operational Control Centre, the Ground Station and the Satellite provides potential attack points. The attacks could be physical or logical (software-based).

The hardware implementing the cryptography and key management protocols can be attacked physically either directly or indirectly. Direct attacks involve probing the hardware, or reverse engineering it and replacing with an attackers version, or stealing the hardware so that an attacker can use it themselves. Indirect attacks involve monitoring the hardware for heat or electromagnetic emissions, or for power consumption, which, in turn, can leak information about the messages or keys being processed by the hardware processors.

The computing systems in the OCC and the Ground Station in which the hardware is operating could also be breached by the unauthorised access of an attacker.

The data links between the hardware components also provide potential attack points, i.e. the network links between the OCC and the Ground Station, and the over-the-air links between the Ground Station and Satellite. These links could be attacked to disrupt the confidentiality, availability or integrity of the telecommand data.

4.3.2 Key Management System

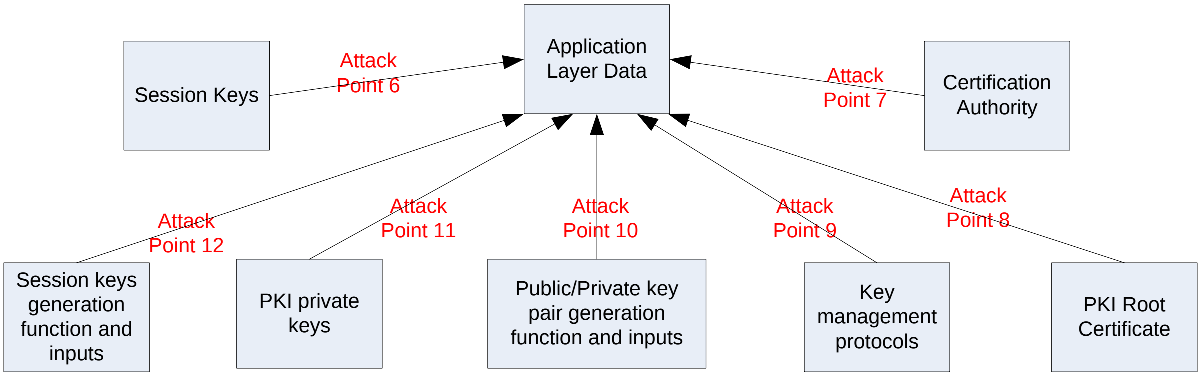

Figure 10: Potential attack points in the Key Management System.

All the components of the Key Management System provide potential attack points. Compromise of any elements of the key management system would compromise the confidentiality, availability and integrity of the application data (the telecommand and housekeeping data).

4.3.3 Key Injection Module

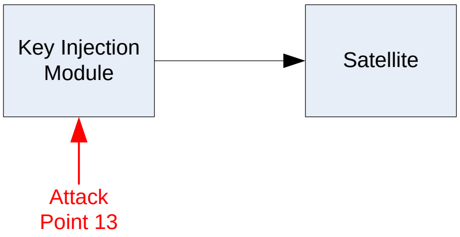

Figure 11: Key Injection Module attack point

The key injection module provides another potential attack point. The module is used to load key material onto the satellite prior to launch. Any compromise of key material at this point would render the key material on the satellite, and therefore any data transmitted, insecure for use for the lifetime of the satellite.

4.4 Threat list

In this section, we identify potential threats to the components in the satellite use-case system.

The logical, physical and human threats described in the following subsections all pertain to particular assets in the list of critical assets. However, ultimately any compromise of these assets leads to vulnerabilities in the application layer data (assets CS1_A_4.1 and CS1_A_4.2), which form the fundamental assets of the system. The security of the entire system depends on the security of these assets.

4.4.1 Threat environment

In the Satellite scenario, it is reasonable to make some assumptions about the threat environment and likely actors. Due to the very high cost and the complex engineering required there are relatively few industrial companies or government agencies which are able to construct and launch satellite hardware and therefore there are relatively few assembly sites. It is assumed that their assembly facilities, OCCs and Ground Stations have adequate measures in place to guard the sites against physical access by intruders. It is also assumed that the computer systems in the OCC and Ground Station are connected to the wider Internet. Even with the assumed protection of adequate firewalls, virus checkers and user authentication procedures, these computer systems may still be susceptible to malicious attack.

The possible attackers who may wish to penetrate the Satellite system, and therefore their capability, depends on the function of the Satellite and the payload it is transmitting. For example, for government sensitive spy satellites, the attackers may range from legions of state-sponsored, highly expert computer scientists working for opposing governments to individual amateur script kiddies. Unclassified satellite data, such as commercial Earth-observation data, may not be of interest to foreign states, but may be of interest to capable amateur hackers who view it as a challenge to break the encryption.

4.4.2 Physical threats

There are a number of attacks that can be carried out in the Satellite scenario that make use of the physical properties of the hardware components in the system. The attacks can be directly or indirectly related to these physical properties. Direct attacks involve hands-on access to the hardware, whereas indirect attacks involve measuring physical quantities emanating from the hardware as a result of the cryptographic algorithm executing.

Direct attacks involve:

- Probing the hardware to read out the static private key or session keys from memory.
- Reverse engineering hardware components and replacing with an attackers version.
- Stealing the hardware so that an attacker can use it themselves.
- Fault injection

Indirect side-channel attacks involve:

- Timing analysis
- Power analysis

All of the hardware components in the Satellite scenario, i.e. the hardware in

- The Satellite (asset CS1_A_1)
- The OCC (CS1_A_2)

- The Core Ground Station (CS1_A_3)
- The Key Injection Module (CS1_A_6)

are potentially susceptible to the direct and indirect physical threats described above. It is obvious, however, that once deployed the satellite is out of physical reach of potential adversaries and therefore several physical risks to the satellite only exists prior to launch. However, it is possible to imagine that fault attacks (which might exploit single event upsets caused by radiation), and timing attacks would still represent a security risk after the launch also.

It is anticipated that there may be particular vulnerabilities which could be exploited physically, with respect to the parts of the lattice algorithms which have variable runtimes, e.g. Gaussian and rejection sampling which are major components of many lattice-based algorithms.

4.4.3 Logical threats

Logical vulnerabilities exist in the software in the system, allowing an attacker to reveal direct or indirect information about the application data, or causing disruption to the normal operation of the system.

The main logical vulnerabilities and threats in the satellite scenario are:

- A session key is directly broken (assets CS1_A_5.1 and CS1_A_5.2) – If a session key is broken due to a brute force attack, the data would be vulnerable until the session key is updated. An attacker could eavesdrop the data by intercepting either the over-the-air communications or the network communications on the ground. It should be noted that brute-force attacks are unlikely to be successful due to the very large number of possible combinations that have to be tried as a result of large key sizes.
- Vulnerabilities in the lattice scheme (assets CS1_A_5.3 and CS1_A_5.4) – If the lattice scheme were to be broken after the satellite has been launched it would be a very difficult, if not impossible, task to re-program the on-board hardware or embedded software with any patches or replacement schemes. This type of vulnerability could render the satellite communications insecure for the lifetime of the satellite. The vulnerabilities in the scheme could arise from
  - A fundamental breach of the underlying hard problem on which the security is based
  - Flaws introduced in particular modifications, patches or software updates released for improved performance or efficiency
  - Incorrect selection of lattice parameters
- Vulnerabilities in the key management protocol (asset CS1_A_5.5) - The vulnerabilities could arise out of introducing flaws whilst integrating lattice-based constructions into existing key management protocols that weren't originally designed for the lattice schemes. For example, if a scheme requires extra hints in a Diffie-Hellman type key exchange, the extra hints could leak information about the key.
- Vulnerabilities in proprietary protocols (asset CS1_A_5.5) – New protocols may need to be developed that utilise lattice-based constructions, which may contain unintended security flaws.
- Implementation flaws in the lattice schemes and key management protocols (asset CS1_ A_5.5) – Although the key management protocols may be designed securely, any bugs in the implementation could result in the disclosure of the private or session keys.

- Vulnerabilities in the PKI infrastructure (assets CS1_A_5.6 and CS1_A_5.7) – If the root certificate installed in the satellite and/or the OCC were compromised or if an attacker was able to penetrate the CA, an attacker could falsely verify a replacement root certificate and gain access to the static private key.

4.4.4 Human threats

Humans are involved at many steps in the development, deployment and operation of a satellite, and have the opportunity to adversely affect the security of the assets identified above. This may be through accidental action, such as mistakes in the configuration of the satellite or the security systems. It may also be through deliberate action. Such actions by unauthorised users are already covered under the physical and logical threats described above. However, accidental or deliberate actions by authorised users should also be considered.

In this use-case, the human threat is limited by the niche application of highly valuable and bespoke items of hardware (the satellite and control equipment) being developed and operated in a restricted environment (protected control centre), i.e. there is a limited number of people accessing a few items of hardware. In addition, once deployed the satellite is out of reach from most human threats, leaving only the OCC and Ground Station vulnerable. Prior to deployment, the satellite and key injection are vulnerable to human threats, thus providing a window of opportunity to degrade the security of the satellite communications. Therefore the physical security of the sites should be protected the most during this window.

In any case, all reasonable steps should be taken ensure that the appropriate access control mechanisms are employed to prevent unauthorised physical access to the operational sites and only allowing accredited persons contact with the hardware components.

In terms of the implementation of the key management system, which is the focus of the use case, most if not all of these threats are out of scope for the SAFEcrypto project. The mitigations of any resulting risks would be primarily procedural and physical measures that are independent of lattice-based cryptography and security protocols. Therefore, we do not consider these threats any further in this document.

4.4.5 Other non-security risks

There are other associated risks, which aren’t a threat to the security of the satellite system, but are a risk to the proper operation of the satellite system as a result of integrating lattice-based security.

It may be the case that key management protocols which utilize lattice-based constructions do not adequately satisfy the functional requirements of the satellite scenario; at present there are no key management protocols designed specifically for satellite communications. The risk is that the protocols required to function for space communications may not be able to accommodate the practical properties of lattice-based schemes. For example,

- The key sizes required by the schemes may be too large to be practically transmitted over the telecommand channels.
- The computation required may result in long signal delays.
- The implementation of the key management protocol may run too slow or be too large to fit on the platforms.
- The protocol may require an increased number of message exchanges, again making it too slow to be practical.
- There are no published lattice-based standards and so there is the considerable possibility that the protocol developed for this scenario will not be interoperable with other systems.

These non-security concerns will be addressed in detail in other work packages, especially in WP8 which deals with the key management systems in each of the scenarios.

4.4.6 Summary of threats

The threats described above are summarised in the following table.

Threat ID	Threat	Assets at risk
CS1_T_1	Probing the hardware	CS1_ A_1[1], CS1_ A_2, CS1_ A_3, CS1_ A_6
CS1_T_2	Reverse engineering hardware components	CS1_ A_1, CS1_ A_2, CS1_ A_3, CS1_ A_6
CS1_T_3	Stealing the hardware and use by attacker	CS1_A_1, CS1_ A_2, CS1_ A_3, CS1_ A_6
CS1_T_4	Timing analysis	CS1_A_1, CS1_A_2, CS1_A_3, CS1_A_6
CS1_T_5	Power analysis	CS1_A_1, CS1_A_2, CS1_A_3, CS1_A_6
CS1_T_6	Fault injection	CS1_A_1, CS1_A_2, CS1_A_3, CS1_A_6
CS1_T_7	Brute force attack on session key	CS1_A_5.1, CS1_A_5.2
CS1_T_8	Breach of the underlying lattice hard problem	CS1_A_5.3, CS1_A_5.4
CS1_T_9	Breach of particular tweaks in the lattice scheme	CS1_A_5.3, CS1_A_5.4
CS1_T_10	Incorrect selection of lattice parameters	CS1_A_5.3, CS1_A_5.4
CS1_T_11	Vulnerabilities in the key management protocol	CS1_A_5.5
CS1_T_12	Vulnerabilities in proprietary protocols	CS1_A_5.5
CS1_T_13	Implementation flaws	CS1_A_5.5
CS1_T_14	Vulnerabilities in the PKI infrastructure	CS1_A_5.6, CS1_A_5.7
CS1_T_15	Impractical schemes and protocols	N/A
CS1_T_16	Un-interoperable protocols developed	N/A

Table 6: Summary of Threats (Satellite)

[1] Prior to launch only

4.5 Risk analysis and Countermeasures

In this section, we attempt to quantify the risks listed previously using the methodology described in Section 3. The method returns a value between 0 and 1, where 1 is the highest risk and 0 is the lowest. The risk is calculated using measures of the vulnerability of the system to the threat, the capability required by an attacker and the impact of the particular threat.

4.5.1 Risk analysis

Most of the physical, logical and human threats described above lead to the compromise of one of the system keys to an attacker. Although the details of the key management system are yet to be decided, it is likely that the system would contain public/private key pairs which could be static and/or ephemeral. Obviously, compromise of static private keys has a high impact on the security of the system whilst compromise of an ephemeral key would have a lower impact because a new one would be generated in the subsequent time period. Similarly, the impact of the session key (either a Key Encryption Key (KEK) or a Data Encryption Key (DEK) derived from the KEK) being compromised would also be lower as they too would be regularly regenerated.

Another possible outcome as a result of the above threats is the denial of service (DoS) through overloading of resources or a software crash. This type of risk is usually associated with bugs or flaws in the implementation.

Should the static private key or session keys be revealed, then an attacker would be able to either eavesdrop on the telecommand data or, more seriously, to take control of the spacecraft by injecting malicious control commands or to deny control of the spacecraft, for the lifetime of the revealed key. If we consider extending the scenario to securing the payload data, then compromise of the private and session keys could result in the same attacks of DoS, eavesdropping or malicious injection of the payload data. In order of the most severe impact to the least severe the possible outcomes affecting the telecommand data are:

- Compromising the integrity – This would allow an attacker to take control of the satellite.
- Compromising the availability – This would allow an attacker to deny service, to send control commands.
- Compromising the confidentiality – This would allow an attacker to eavesdrop on the telecommand data.

If this scenario was extended to securing the payload data channel, then the impact of eavesdropping into the payload data would be much higher, especially if the data was government sensitive. Table 7 examines each of the threats identified in Table 6, providing scores for the vulnerability, capability and impact metrics, and calculates the overall risk score.

Threat ID	Threat	Vulnerability	Capability	Impact	Risk
CS1_T_1a	probing the OCC or GS hardware for static private keys	5	3	4	60/125 = 0.48 (MEDIUM)

Threat ID	Threat	Vulnerability	Capability	Impact	Risk
CS1_T_1b	probing the satellite hardware for static private keys (prior to launch)	5	2	4	40/125 = 0.32 (LOW)
CS1_T_1c	probing the KIM hardware for static private keys (prior to launch)	5	2	4	40/125 = 0.32 (LOW)
CS1_T_1d	probing the OCC or GS hardware for session keys	4	3	3	36/125 = 0.29 (LOW)
CS1_T_2a	reverse engineering OCC or GS hardware components	4	3	4	48/125 = 0.39 (MEDIUM)
CS1_T_2b	reverse engineering KIM hardware components (prior to launch)	4	4	4	64/125 = 0.51 (MEDIUM)
CS1_T_3a	stealing the OCC or GS hardware and use by attacker	4	3	4	48/125 = 0.39 (MEDIUM)
CS1_T_3b	stealing the KIM hardware and use by attacker (prior to launch)	4	2	4	32/125 = 0.26 (LOW)
CS1_T_4	timing analysis of OCC or GS hardware	4	3	4	48/125 = 0.39 (MEDIUM)
CS1_T_5	power analysis of OCC or GS hardware	4	3	4	48/125 = 0.39 (MEDIUM)
CS1_T_6	fault injection of OCC or GS hardware	4	3	4	48/125 = 0.39 (MEDIUM)
CS1_T_7	brute force attack on session key	4	1	3	12/125 = 0.1 Low
CS1_T_8	breach of the underlying lattice hard problem	5	1	5	25/125 = 0.2 Low
CS1_T_9	breach of particular tweaks in the lattice scheme	4	3	4	48/125 = 0.39 (MEDIUM)
CS1_T_10	incorrect selection of	4	3	4	48/125 =

Threat ID	Threat	Vulnerability	Capability	Impact	Risk
	lattice parameters				0.39 (MEDIUM)
CS1_T_11	vulnerabilities in the key management protocol	4	2	4	32/125= 0.26 (LOW)
CS1_T_12	vulnerabilities in proprietary protocols	4	5	4	80/125 = 0.64 (MEDIUM-HIGH)
CS1_T_13	implementation flaws	4	5	4	80/125 = 0.64 (MEDIUM-HIGH)
CS1_T_14	vulnerabilities in the PKI infrastructure	5	3	4	60/125 = 0.48 (MEDIUM)
CS1_T_15	Impractical schemes and protocols	N/A	N/A	N/A	LOW
CS1_T_16	Un-interoperable protocols developed	N/A	N/A	N/A	HIGH

Table 7: Risk assessment for the Satellite scenario

In general, most of the identified risks in Table 8 are deemed to be Medium or Low. The greatest risks are in using proprietary protocols, because the protocol would not be analysed by a large number of cryptanalysts and therefore could contain security vulnerabilities in any implementation flaws, which could introduce unintended weak points into the system.

The largest vulnerabilities arise if the lattice scheme on which the entire security of the system is based is broken or if the static private key is revealed. However, the capability of an attacker to break the lattice scheme is deemed to be very low, because of the known hardness of the underlying security reduction in the case of the lattice scheme. If the static private key is obtained by an adversary, they would be able to generate their own session keys leading to the loss of the security of the system. Furthermore, if the compromise of the static key was undetected, the adversary would be able to mount a man-in-the-middle attack and be able to attack the telecommand data without the OCC knowing.

4.5.2 Risk Calculation Scenario

We now perform a risk calculation for the threat scenario of losing the integrity of the telecommand data. In this case, an adversary would be able to take over control of the satellite, possibly moving it out of its orbit or completely losing control of it which would be catastrophic for the satellite operators. The methodology follows the approach detailed in Section 3.3.

Figure 12 shows the threats that pose a risk to the static private key (threats, CS1_T_8, CS1_T_9 and CS1_T_10) and to the session key (threat CS1_T_7). The outcome of the final stage is to lose the integrity of the telecommand data. The impact of this has the highest score of 100 as this outcome would be catastrophic for the satellite operator. Working backwards through the stages, the vulnerability and capability required to achieve this if the session key were compromised would be straightforward and so these variables have the highest scores of 5 out of 5. In turn, the vulnerability and capability to achieve this, given that the private static key is compromised also have the highest scores of 5. This leads us to the overall risk calculation as given in Table 8.

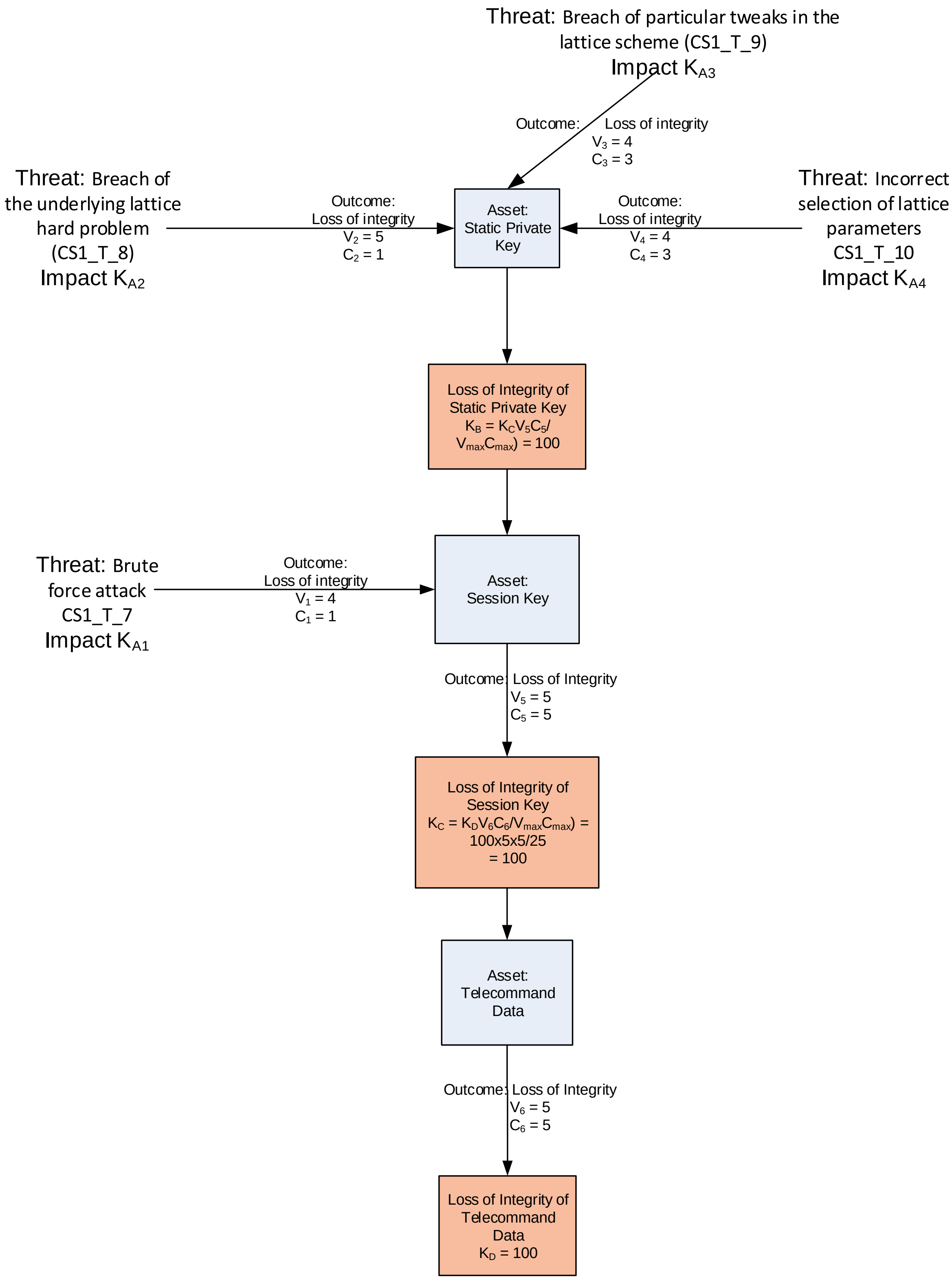
Threat: Breach of particular tweaks in the lattice scheme (CS1_T_9)
Impact K_{A3}
Outcome: Loss of integrity
$V_3 = 4$
$C_3 = 3$
Threat: Breach of the underlying lattice hard problem (CS1_T_8)
Impact K_{A2}
Outcome: Loss of integrity
$V_2 = 5$
$C_2 = 1$
Asset: Static Private Key
Outcome: Loss of integrity
$V_4 = 4$
$C_4 = 3$
Threat: Incorrect selection of lattice parameters CS1_T_10
Impact K_{A4}
Loss of Integrity of Static Private Key
$K_B = K_C V_5 C_5 / V_{max} C_{max}) = 100$
Threat: Brute force attack CS1_T_7
Impact K_{A1}
Outcome: Loss of integrity
$V_1 = 4$
$C_1 = 1$
Asset: Session Key
Outcome: Loss of Integrity
$V_5 = 5$
$C_5 = 5$
Loss of Integrity of Session Key
$K_C = K_D V_6 C_6 / V_{max} C_{max}) = 100x5x5/25 = 100$
Asset: Telecommand Data
Outcome: Loss of Integrity
$V_6 = 5$
$C_6 = 5$
Loss of Integrity of Telecommand Data
$K_D = 100$

Figure 12: Threats to the static private key and session key

Impact Stage	Threats leading to the next	Impact Calculation	Impact Score

	stage		
D	-	-	K_D= 100
C	-	$K_C = K_D V_6 C_6/(V_{max} C_{max})$	K_C = 100
	CS1_T_7	$K_{A1} = K_C V_1 C_1/(V_{max} C_{max})$	$K_{A1} = 16$
B		$K_B = K_C V_5 C_5/(V_{max} C_{max})$	K_B = 100
A	CS1_T_8	$K_{A2} = K_B V_2 C_2/(V_{max} C_{max})$	$K_{A2} = 20$
	CS1_T_9	$K_{A3} = K_B V_3 C_3/(V_{max} C_{max})$	$K_{A3} = 48$
	CS1_T_10	$K_{A4} = K_B V_4 C_4/(V_{max} C_{max})$	$K_{A4} = 48$

Table 8: Impact Calculation Table for Threats in the Satellite Scenario

From Table 8, we can see that the risk of compromise of the session key leading to the loss of control of the satellite is Very Low. The risk of the breach of the lattice scheme is Low, whilst the risk to the satellite of incorrect scheme parameters and of introducing flaws by tweaking the scheme are calculated to be Medium.

Now that the main vulnerabilities and threats to the assets in the Satellite scenario have been identified and assessed, we propose some countermeasures to mitigate these risks.

4.6 Countermeasures

In this section, we propose some security strategies and preventative measures that should be adopted in order to mitigate the risks posed by the threats listed in the previous sections.

In terms of physical security, the obvious initial measure that must be taken is to prevent physical access to the hardware by an attacker. It is reasonable to assume that in our satellite scenario, the limited number of sites involved and the high value of the equipment and infrastructure required to deploy such a specialised system would be adequately secured and guarded, with only accredited individuals being granted access. However, given the expected long-life service of such a system, possibly operational for decades, it is possible that the human security at the Operational Control Centre or at the Ground Station could be breached at some point. Therefore, countermeasures to physical attack need to be built into the devices.

There are a number of standard measures that can be taken to protect the hardware in the satellite, the OCC and the Ground Station against physical attack.

A commonly-used approach to guard against timing analysis is to remove any differences in computational time which depend on the secret key. This can be achieved by implementing constant-time functions in the algorithms, so that no matter what the input parameters are to the functions, they execute in the same constant-time and therefore will not leak any information about the input (the secret key). Similarly, conditional branches which depend on the secret key should also be avoided.

Attacks using power analysis can be countered by masking, where the dependency between the data processed by the device and the secret data is removed, or by a different technique, known as hiding, where the dependency between the data being processed and the power consumption of the device is removed, or by a combination of the two.

To counteract fault attacks, measures used for fault tolerance support guaranteeing the reliability of computations are employed. These countermeasures usually require some type of redundancy, either temporal, where the operation is repeated multiple times and the results are checked against each other, or spatial, where multiple copies of the circuit are instantiated and the results are compared. Contrary to redundancy for fault tolerance, redundancy implemented for resistance to fault attacks does not necessarily have to produce the correct result. In the case of fault attack countermeasures, it is often sufficient to simply interrupt the computation before the adversary learns the wrongly computed information which could be exploited to extract the secret information.

In respect to the security of the lattice scheme and the associated key management protocols, there are a number of strategies that can be used to reduce the risk of them introducing flaws. The underlying security reductions of the lattice schemes are well understood and thought to be secure, but nevertheless, the schemes themselves and the recommended parameters should be circulated to cryptanalysts and security experts for thorough scrutiny. An ideal way to do this might be through standards bodies such as ETSI and NIST, who are seeking to define new standards for quantum-resistant cryptography. Candidate schemes are subjected to expert analysis and approval before being standardised.

A related approach could be to publish an open competition offering a prize to anyone who breaks the scheme with the given parameters. This has been done by Philips with their HIMMO scheme [30] and it encourages a widespread analysis of the scheme.

The same approach, of distributing the scheme through various channels for open scrutiny, could be taken when proposing key management protocols, either when modifying existing schemes or when defining new ones from scratch. Where existing PKIs are used, well-known and widely used implementations and services should be used to reduce the risk of using untested ones, which may have un-noticed flaws.

Turning to implementation vulnerabilities, there are a number of steps that can be taken to reduce the risk of introducing unintended security flaws. The SAFEcrypto project will ensure high quality crypto implementations by setting out detailed software coding standards within deliverable D6.1 – Lattice Based Software Requirements Specification. Adherence to those standards will be enforced via formal software code review by experienced security software practitioners from Thales, the applied crypto team in Queen’s University Belfast or optionally an external review from a government agency such as CESG[1]

[1] The Communications-Electronics Standards Group, part of UK GCHQ and the UK government’s National Technical Authority for Information Assurance.

The code should be written adhering to a strict coding standard which focuses on adopting best practice for security software (this is addressed in Deliverable 6.1[1]). The developed software can also be checked for potential flaws using a static analysis tool such as Lint, which would highlight any buffer overflows, dangling pointers or uninitialized memory, for example.

The countermeasures are summarised in Table 9.

Threat ID	Threat	Countermeasure
CS1_T_1	Probing the hardware	Physical security. Measures to protect against physical attack, constant time, masking power fluctuations etc.
CS1_T_2	Reverse engineering hardware components	Physical security.
CS1_T_3	Stealing the hardware and use by attacker	Physical security.
CS1_T_4	Timing analysis	Constant time implementations
CS1_T_5	Power analysis	Hiding and Masking techniques
CS1_T_6	Fault injection	Temporal or spatial redundancy
CS1_T_7	Brute force attack on session key	Choose secure parameters
CS1_T_8	Breach of the underlying lattice hard problem	Publish the scheme openly for cryptanalysis by a wide range of experts. Maybe offer a prize for breaking the scheme, before it is deployed, to encourage wide-spread cryptanalysis
CS1_T_9	Breach of particular modifications in the lattice scheme	Publish the scheme openly for cryptanalysis by a wide range of experts. Maybe offer a prize for breaking the scheme, before it is deployed, to encourage wide-spread cryptanalysis
CS1_T_10	Incorrect selection of lattice parameters	Use the parameters recommended for higher levels of security than thought to be needed. Use parameters recommended by multiple independent sources
CS1_T_11	Vulnerabilities in the key management protocol	Use well established protocols wherever possible. Submit modifications to cryptographic community for expert analysis
CS1_T_12	Vulnerabilities in proprietary protocols	Use standardised protocols wherever possible
CS1_T_13	Implementation flaws	Code reviews by security experts
CS1_T_14	Vulnerabilities in the PKI infrastructure	Use a well-established PKI protocol and/or implementation
CS1_T_15	Impractical schemes and protocols	Liaise with Thales Alenia Space to develop practical schemes
CS1_T_16	Un-interoperable protocols	Liaise with standards bodies to agree interoperable

[1] A crypto-specific coding standard is currently being developed in Deliverable D6.1 "Lattice-based Software Requirements Specification."

Threat ID	Threat	Countermeasure
	developed	protocols and algorithms.

Table 9: Summary of the countermeasures against the identified risks.

5 COTS in Public Safety Communications

5.1 High level view

In this use case, public safety personnel communicate using COTS, LBC is used primarily for authentication and session key exchange. IBE is also being investigated for use in securing group calls, as well as providing fast group establishment, dynamic over-the-air regrouping and revocation of rogue group members. Hence, LBC may also be used to provide IBE.

5.1.1 System view

Figure 13 shows an overview of COTS public safety communication systems. A number of group calls can be established. The system has three primary actors:

- The calling parties - Public safety communication callers and receivers.
- The Signalling Server - Responsible for establishing the communication link between the calling parties. This includes passing call and routing parameters to the calling parties.
- Identity provider - Responsible for asserting the identity of the callees.

In this use case, we assume that WebRTC, [33], is used in an IP-based communication network to make group calls for public safety personnel. WebRTC is an API that supports browser to browser applications for voice calling, video chat and peer-to-peer file sharing.

When making a call, a caller connects first to the Signalling Server, which is responsible for call initiation and passing call and routing parameters to the call members. An Identity Provider is responsible for asserting the identity of the calling party members and the signalling server.

WebRTC encrypts its real-time (application layer) data using the Datagram Transport Layer Security (DTLS) protocol, which is defined by [21], [22], [23] and [24]. DTLS is a standardized protocol that is built into all browsers that support WebRTC. It is modelled on the TLS protocol. For more detail on the use case please refer to D9.1.

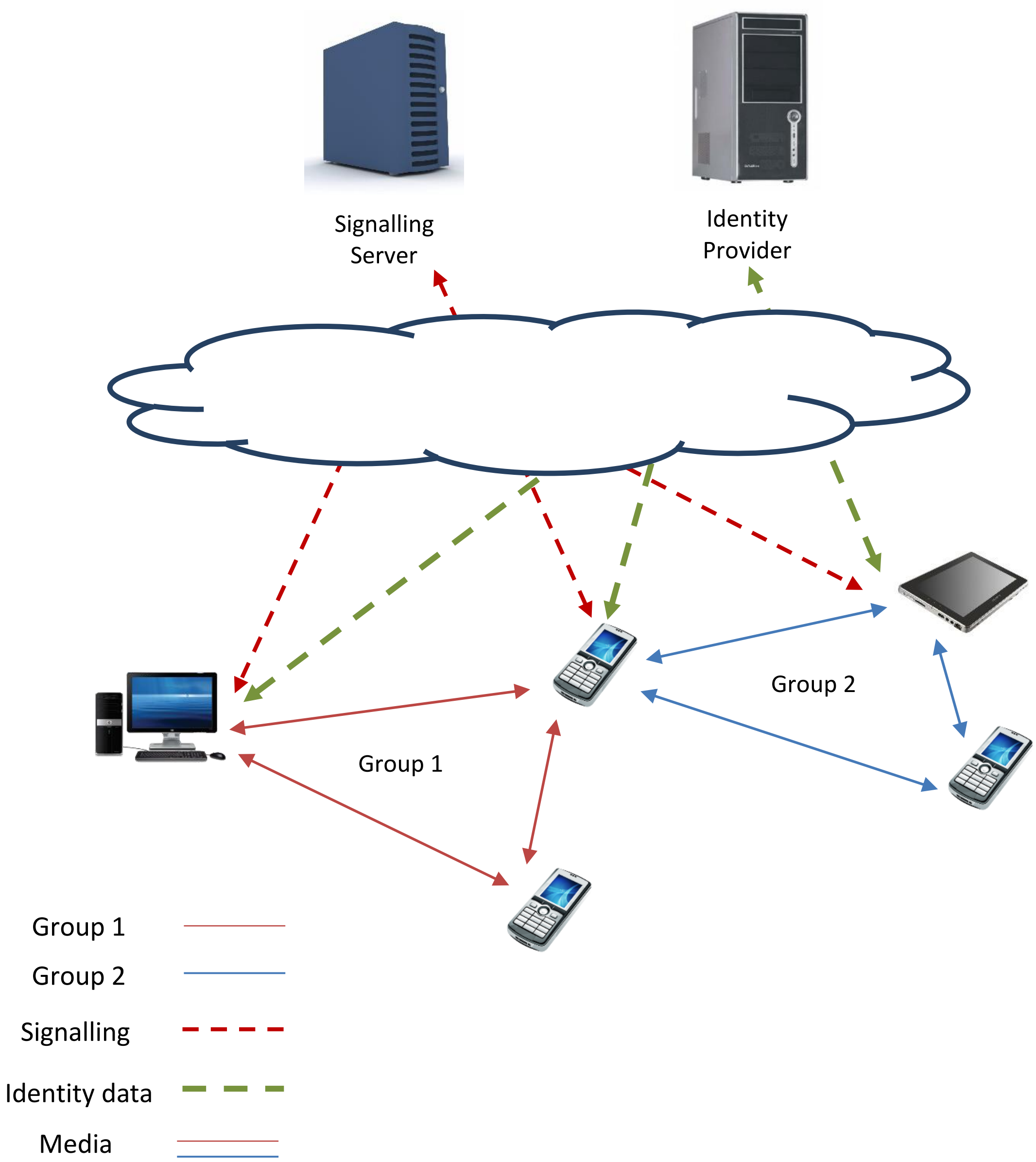

Figure 13: Overview of COTS Public Safety Communication System

5.1.2 Components

The COTS in Public Safety use case is comprised of the following components:

- Calling parties
  - Public safety communication callers and callees
  - Browser
  - Devices e.g. laptop, smart phone, etc.
- Signalling Server
- Identity Provider

5.2 Critical system assets

These are assets that, if successfully attacked or compromised, could potentially have a serious impact or consequence on the COTS in Public Safety use case.

5.2.1 Assets description

1. Application layer data - This is the actual data-content communicated between application layers. This includes
   a. Video streams, voice calls, file attachments, messages, chats, shared desktop etc.
   b. Thin client monitoring and control data. With broadband being rolled-out to the critical communications user community, it is likely that application layer data will include sensor readings, and remote command and control applications, etc.
2. Calling party's identities - The importance of this asset depends on the context and whether the communications sessions is meant to be confidential or not.
3. Calling party's location - This asset is obtained via observing and analyzing the data traffic. Location data can be obtained when using the Automatic Vehicle Location (AVL) and Location Information Protocol (LIP) services. Location information is communicated as a type of application layer data. The importance of this asset depends on the context.
4. Session keys - These are the symmetric-keys used in encrypting the application layer data
5. Session key generation function and inputs - This includes the function used to generate the session key and its inputs. The session key generating function should not be a secret. If knowledge of the session key generating function and its input values are wholly or partially acquired, it is possible to generate the session keys, albeit it is over a reduced key space.
6. Private keys of a public/asymmetric key pairs.
7. Asymmetric key pair generating function and its inputs.
8. Calling party's device configuration data - This includes types of the devices used by the calling parties, operating systems, web-browser, etc. The importance of this information depends on the context, and whether they may be used to mount another attack on another more critical asset.
9. Signalling server location and configuration - This is the server used to connect the calling parties at call set-up. The importance of this information depends on the context.
10. Identity provider server location and configuration - This is the entity responsible for proving the authenticity of the identities of the calling parties. The importance of this information depends on the context, and whether it may be used to mount attacks.
11. Calling party's devices - This asset is considered because accessing it, may lead allow an attacker to obtain downstream critical assets such as: application layer data, identities, stored keys, implementation information, etc.
12. Signalling server - This is the physical server itself. This asset is considered because accessing it may allow an attacker to obtain data assets stored on the server.
13. Identity provider server - This is the server itself (similar to 12).

Table 10 summarizes and provides reference IDs for the assets identified in the COTS in Public Safety use case.

Reference	Asset
CS2_A_1	Application layer data
CS2_A_2	Calling parties identities
CS2_A_3	Calling parties location
CS2_A_4	Session keys
CS2_A_5	Session keys generation function and inputs
CS2_A_6	Private keys of the asymmetric/public key pair
CS2_A_7	Asymmetric key pair generation function and inputs
CS2_A_8	Calling parties' devices configuration data
CS2_A_9	Signalling server location and configuration
CS2_A_10	Identity provider server location and configuration
CS2_A_11	Calling parties devices
CS2_A_12	Signalling server
CS2_A_13	Identity provider server

Table 10: Summary of COTS assets

5.2.2 Assets classification and dependency relationship

Assets described in the 5.2.1, may be classified into

- Goal (Top-level) assets. These include:
    - Voice, video, chat messages
    - Attachment files,
    - Shared desktop
    - Interactive maps
    - Data at rest
    - Over the air software updates
    - User's location, user identity and operations and locations of operations and incidents
- Key (Root) assets. These include:
    - Key generating functions
    - Calling devices
    - Signalling server
    - System configuration
    - Keys; Session keys and private keys

The classification helps to determine the criticality of an asset in terms of its impact when successfully attacked. Goal (top-level) assets, can help to determine the impact of an asset on the use case from a top-level perspective. For example, the impact of eavesdropping on "Voice" data on the operational success of an emergency response situation. On the other hand, Key (Root) assets, as the name implies, are the key assets which, if compromised, can lead to the increased vulnerability of the goal assets. There is an interdependency relationship between the key assets and the goal assets, where the latter is threatened if the former is successfully attacked.

The listed assets are not necessarily mutually exclusive.

In Figure 14, the asset dependency diagram is shown for one of the top-level assets (voice in this case). In this figure, we show key assets that if successfully attacked can lead to (or affect) our top-level asset. The relationship between assets in the assets dependency diagram, is needed when analysing risks in a threat scenario.

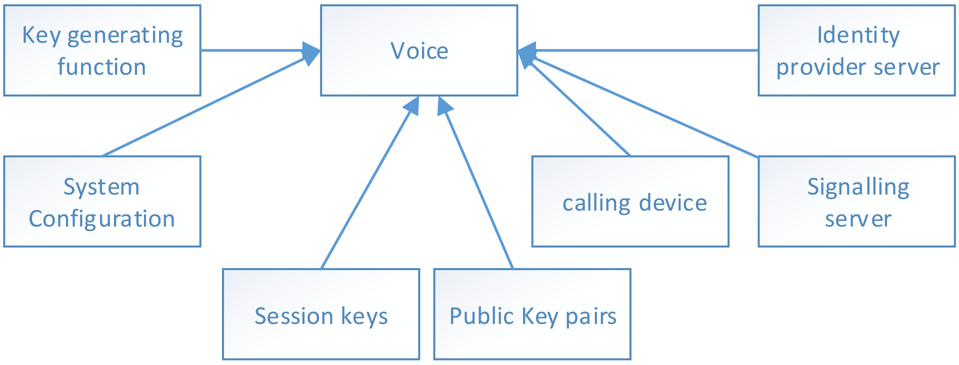

Figure 14: Asset dependency diagram for the asset "Voice"

5.3 Attack points

Assets identified in 5.2, are all targets for attack, which would aim to attack one or a combination of the following security aspects

- Availability
- Confidentiality
- Integrity
- Non-repudiation

Figure 15 shows potential attack points for a Public Protection and Disaster Relief (PPDR) case study that uses Web-RTC. The architecture shown is derived from [25]. There are 9 attack points identified in the figure. Attack points 1 to 7 are on communication links, between the actors in this use case. Attack point-8 and 9 are on the device hardware itself. Each of the identified attack points can be divided into several points in time and/or in location. For example, attack point-9 is on the device hardware, this may be divided into

- Device memory
- Hardware implementation

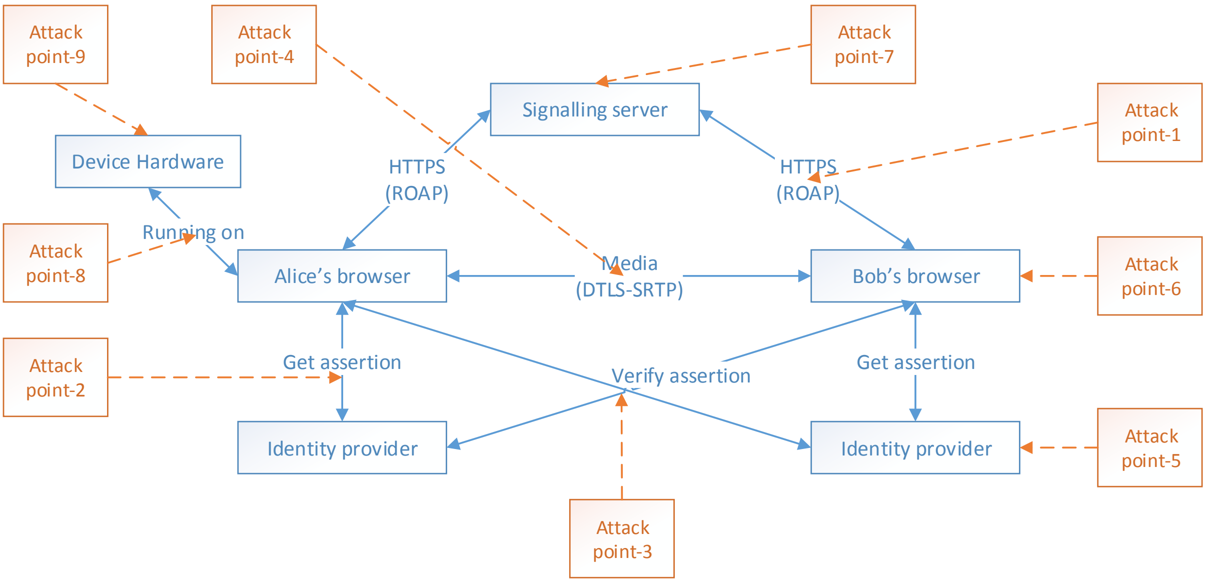

Figure 15: Potential attack points for a PPDR communication system based on Web-RTC

5.4 Threat list

5.4.1 Threat environment

The devices used in this use case are more accessible when compared to other use cases. The fact that communication is based on COTS devices using Web-RTC increases the exposure to a host of physical threats and logical threats more so than the other use cases. This increased exposure, can be of benefit because it allows for increased cross checking and better detection of threats and vulnerabilities, and consequently effective countermeasures (i.e. the increased exposure may lead to a mature and resilient secure architecture implementation).

5.4.2 Physical threats

Hardware and software COTS components used in public safety scenario are susceptible to physical attacks. Since the adversary can potentially have complete access to the device, power analysis (including EM analysis), fault and timing attacks could be, in principle, carried out successfully before and after the deployment of the devices. While for power analysis (and much likely also for EM) and fault attacks, the adversary must have the device in their possession, consideration must also be given to the possibility of successfully mounting a timing attack from a remote location. It is nevertheless important to mention that the availability of the device would increase the precision of timing attacks, since the adversary could easily get precise timing information.

5.4.3 Logical threats

Most of the logical attacks, identified in section 2.2, can be applied to the COTS in Public Safety Communications deployment of the Web-RTC. These are all listed in **Error! Reference source not found.**. However, these attacks are not targeting the crypto-lattice problem (i.e. they are independent of the type of public key system that is used), instead they target the protocols used and the implementation of the Web-RTC security architecture.

Most of the logical attacks try to circumvent the hardness of the LBC problem and its software or hardware implementations by attacking how it is deployed in the system. Attacks on the protocols and their implementations are more frequent and appealing and to a degree easier for the attacker.

Nevertheless, logical attacks can potentially be used against LBC implementations to reduce the scale of brute force attacks on the key.

For the scope of this deliverable, we focus on the logical threats that can be identified for LBC, which are:

- Software implementation errors in the crypto
    - Random number generation function
    - Memory safety violations
- Input validation errors
    - Cross site scripting
    - Code injection
- Hardware implementation errors

5.4.4 Human threats

Human threats are present in this use case. Social engineering, phishing attacks, or stealing devices can be used to mount logical attacks, physical side channel attacks or to bypass the cryptography and protocols. Though human threats are deemed outside of the scope of the LBC-crypto application, they are usually used in the context of an attack scenario to escalate an impact of the attack and/or as enablers for other attacks. For example a phishing email that may lead an authenticated calling party (the victim) to download key-logging malware which in turn steals the user's authentication credentials, in order to mount a cross-site scripting attack.

5.4.5 Summary of threats

The threats described above are summarised in the following table.

Threat ID	Threat	Assets at risk
CS2_T_1	Simple power analysis attack	CS2_A_6
CS2_T_2	Differential power analysis attack	CS2_A_6
CS2_T_3	Timing analysis attack	CS2_A_6
CS2_T_4	Electronic Eavesdropping	CS2_A_1, CS2_A_2, CS2_A_6
CS2_T_5	Voice Impersonating	CS2_A_1, CS2_A_2
CS2_T_6	Electromagnetic emanation	CS2_A_6
CS2_T_7	Side Channel: Fault Injection	CS2_A_6
CS2_T_8	Stealing the hardware and use it by the attacker	CS2_A_1, CS2_A_2, CS2_A_8 CS2_A_11, CS2_A_12, CS2_A_13
CS2_T_9	Brute force attack on session key	CS2_A_4
CS2_T_10	Breach of the underlying lattice hard problem	CS2_A_6
CS2_T_11	Breach of particular modifications in the lattice scheme	CS2_A_6
CS2_T_12	Incorrect selection of lattice parameters	CS2_A_6

CS2_T_13	Exploitation of vulnerabilities in the key management protocol	CS2_A_4, CS2_A_5, CS2_A_6, CS2_A_7
CS2_T_14	Implementation flaws	CS2_A_6
CS2_T_14.1	Software implementation errors: Memory safety Violations: Buffer overflows and over reads	CS2_A_1, CS2_A_2, CS2_A_8, CS2_9, CS2_A_10
CS2_T_14.2	Software implementation errors: Memory safety Violations: Dangling pointers	CS2_A_1, CS2_A_2, CS2_A_8, CS2_9, CS2_A_10
CS2_T_15.1	Input validation errors: Format string attacks	CS2_A_1, CS2_A_2, CS2_A_8, CS2_9, CS2_A_10
CS2_T_15.2	Input validation errors: Code injection	CS2_A_1, CS2_A_2, CS2_A_8, CS2_9, CS2_A_10
CS2_T_15.3	Input validation errors: Cross-site scripting	CS2_A_1, CS2_A_2, CS2_A_8, CS2_9, CS2_A_10
CS2_T_15.4	Input validation errors: HTTP Header injection	CS2_A_1, CS2_A_2, CS2_A_8, CS2_9, CS2_A_10

Table 11: Summary of Threats (COTS)

5.5 Risk analysis

Analysing the risk of potential threats in a case study, may start by identifying the impact score of a threat on the case study. The impact (also known as consequence) score can be measured by the help of the Impact Score in Table 12

Impact Score	Impact Description
100	Rescue operation is completely obstructed
90	Rescue operation is impeded by 90%
80	Rescue operation is impeded by 80%
70	Rescue operation is impeded by 70%
60	Rescue operation is impeded by 60%
50	Rescue operation is impeded by 30%
40	Rescue operation is impeded by 15%
30	Rescue operation is impeded by 10%
20	Rescue operation is slightly impeded
10	Rescue operation is not significantly impeded

Table 12 Impact Score Description for the COTS in public Safely Case studies

The success of a rescue operation is highly dependent on the availability, confidentiality and integrity of the communication systems in general and specifically the top level assets identified in 5.2. Threats to an LBC implementation that could affect these aspects of the communication system, and might jeopardise a rescue operation, would therefore have a higher impact and consequently higher risk value.

In risk analysis, however, the relationship between a threat to the LBC and the impact on the case study might not be easily identified. We claim that the risk analysis methodology used in this section, would be able to measure the *Impact*, and therefore the *Risk*, of an attack and/or a threat to the LBC in the case study.

5.5.1 Risk analysis

A risk analysis may start by identifying the potential threat sources. Next, we rank each threat source's capability to carry each possible threat. Table 2 is used when assigning a capability score to a threat source who poses a threat and may carry out an attack. Table 13 shows the capability score for two Threat sources (Terrorists and Hackers) to carry out a number of different attacks. The table does not list all potential threats but illustrates how different threat sources may have different capabilities scores.

Threat Source	Threat ID	Threat	Capability score, C
Terrorist	CS2_T_1	Simple power analysis attack	4
Hacker	CS2_T_1	Simple power analysis attack	2
Terrorist	CS2_T_2	Differential power analysis attack	4
Hacker	CS2_T_2	Differential power analysis attack	2
Terrorist	CS2_T_3	Timing analysis attack	4
Hacker	CS2_T_3	Timing analysis attack	2
Terrorist	CS2_T_4	Electronic Eavesdropping	5
Hacker	CS2_T_4	Electronic Eavesdropping	4
Terrorist	CS2_T_5	Voice Impersonating	3
Hacker	CS2_T_5	Voice Impersonating	1

Table 13: T.S. capability score to carry out threats

After assigning a Capability score for an attacker to carry out an attack, the next step in the risk calculation is to assign a vulnerability score for each threat.

5.5.2 Impact and Risk calculations

Figure 16 shows an attack scenario carried out by a threat source. The threat source (TS) is assumed to be a Terrorist organisation with vast resources. Hence, the capability score for each attack is relatively high. The scenario in the figure starts with a power analysis attack targeting the secret key. If the attack succeeds, the TS may target the "session keys" asset either by compromising its confidentiality, or its availability. Compromising the session keys confidentiality is easily done if the secret key of public key crypto pair is already compromised. Hence the vulnerability to threat T3 “Eavesdrop attack” in Figure 16 , is assumed to be maximum $V_3 = V_{max}= 5$, and the capability of the T.S. to carry out that attack is also maximum $C_3 = C_{max} = 5$. After compromising the confidentiality of the session key, the TS may targets the confidentiality of the “Data over the air” asset by carrying out an eavesdrop attack (Threat T5 in Figure 16). Alternatively the TS may target the integrity of the “Data over the air”, by carrying out an injecting data attack (threat T7), or by doing voice impersonating and pretend to be a legitimate caller (threat T6). Both attacks, T6 and T7 lead to endangering the rescue operation. Voice impersonating attack (T6) is expected to be difficult for the TS to carry out, hence, it is given low capability and vulnerability scores $C_6 = 3$, and $V_6 = 2$.

The threat scenarios shown in the figure can lead to the following eventual impacts:

- Decrypted data over the air (depicted in Figure 16 as Impact K_E). i.e. the confidentiality of the top-level CS2_A_1 asset (see Table 10) is threatened.
- Impeding the rescue operation (Impact K_H) through
    - Communication unavailable between legitimate parties. The availability of CS2_A_1 asset (Impact K_C)
    - Impersonating (Impact K_G)
    - Modifying the data over the air (Impact K_F) i.e. the integrity of CS2_A_1

It has been judged that the first impact may have an impact score of $K_E = 40$. The second impact has been judged to have an impact score K=100 (see Table 12). To calculate the risk of this threat scenario, the impact score of each intermediate stage of the scenarios should be determined first. The impact is measured with respect to the eventual impacts.

Figure 17 shows the calculation of the impact at the different intermediate stages of a threat scenario.

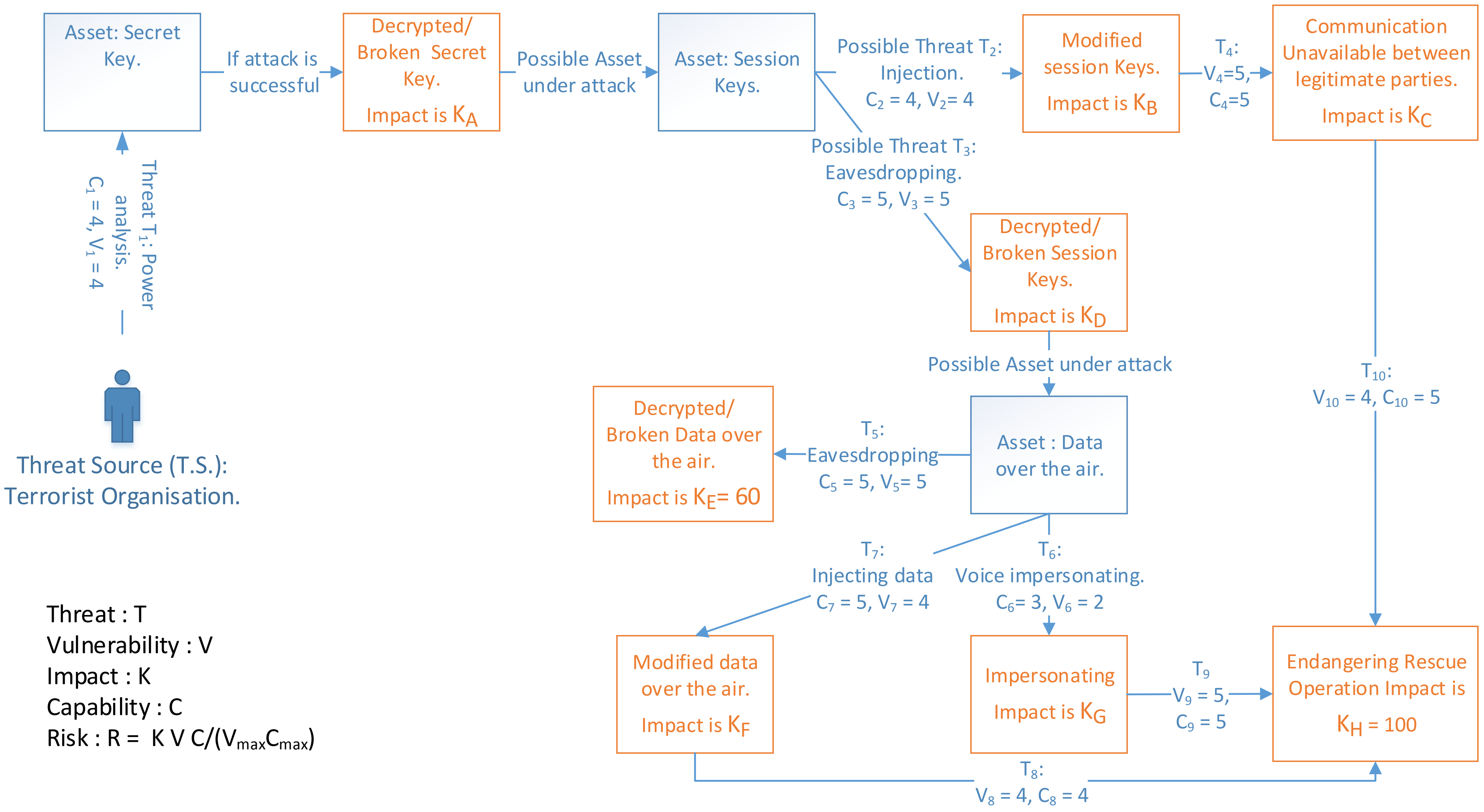

Figure 16: Attack scenario carried out by a Threat source

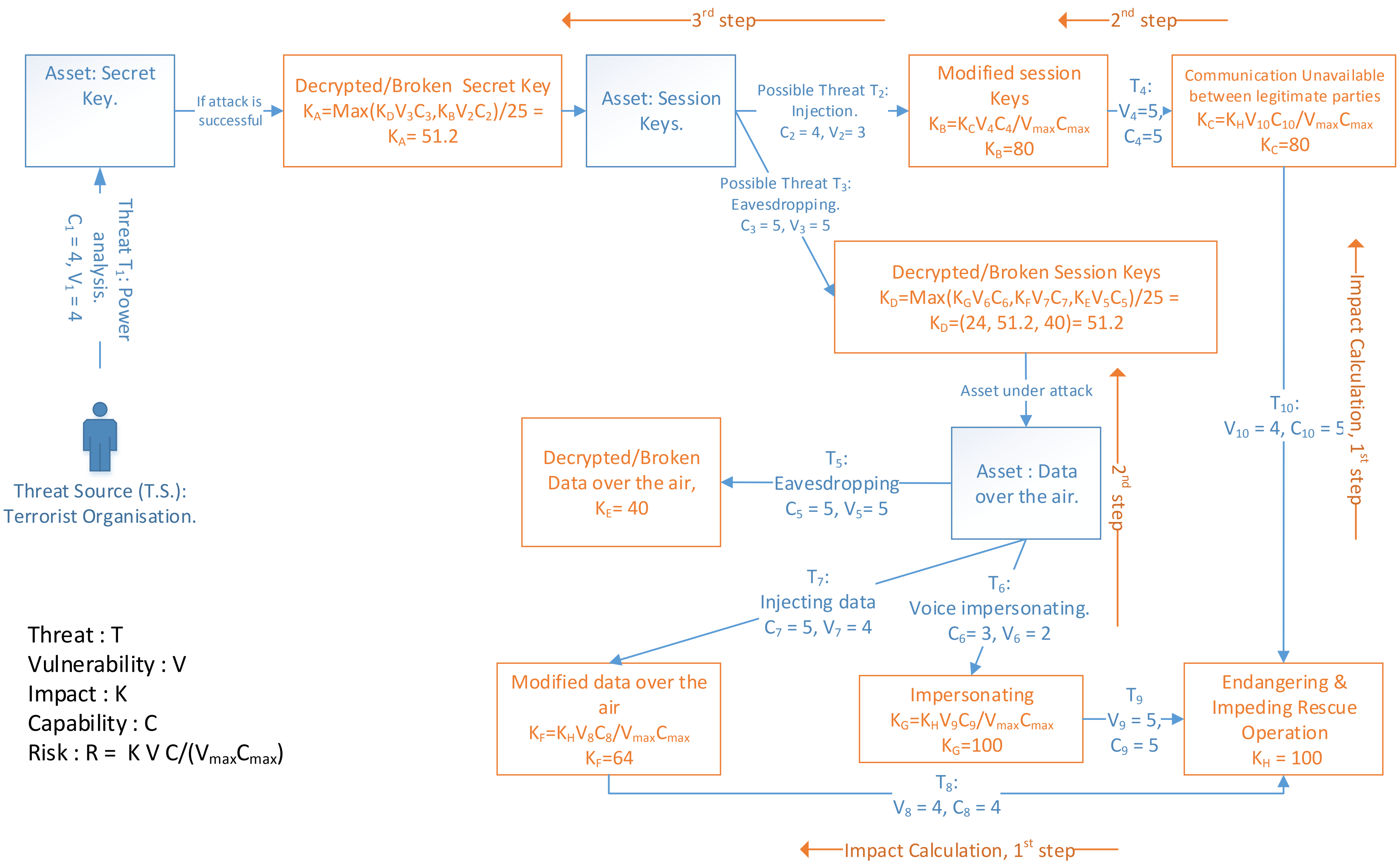
3rd step
2nd step
Asset: Secret Key.
If attack is successful
Decrypted/Broken Secret Key
K_A=Max($K_D V_3 C_3$,$K_B V_2 C_2$)/25 =
K_A= 51.2
Asset: Session Keys.
Possible Threat T_2: Injection.
C_2 = 4, V_2= 3
Modified session Keys
K_B=$K_C V_4 C_4$/$V_{max} C_{max}$
K_B=80
T_4:
V_4=5,
C_4=5
Communication Unavailable between legitimate parties
K_C=$K_H V_{10} C_{10}$/$V_{max} C_{max}$
K_C=80
Possible Threat T_3: Eavesdropping.
C_3 = 5, V_3 = 5
Decrypted/Broken Session Keys
K_D=Max($K_G V_6 C_6$,$K_F V_7 C_7$,$K_E V_5 C_5$)/25 =
K_D=(24, 51.2, 40)= 51.2
Threat T_1: Power analysis.
C_1 = 4, V_1 = 4
Threat Source (T.S.): Terrorist Organisation.
Asset under attack
Impact Calculation, 1st step
2nd step
T_{10}:
V_{10} = 4, C_{10} = 5
Decrypted/Broken Data over the air,
K_E= 40
T_5:
Eavesdropping
C_5 = 5, V_5= 5
Asset : Data over the air.
T_7:
Injecting data
C_7 = 5, V_7 = 4
T_6:
Voice impersonating.
C_6= 3, V_6 = 2
Threat : T
Vulnerability : V
Impact : K
Capability : C
Risk : R = K V C/($V_{max} C_{max}$)
Modified data over the air
K_F=$K_H V_8 C_8$/$V_{max} C_{max}$
K_F=64
Impersonating
K_G=$K_H V_9 C_9$/$V_{max} C_{max}$
K_G=100
T_9
V_9 = 5,
C_9 = 5
Endangering & Impeding Rescue Operation
K_H = 100
T_8:
V_8 = 4, C_8 = 4
Impact Calculation, 1st step

Figure 17: Calculating the impact

Impact Stage	Threats leading to the next stage	Impact Calculation	Impact Score
H	-	-	K_H= 100
G	T_9	$K_G = K_H V_9 C_9/(V_{max} C_{max})$	K_G = 100
F	T_8	$K_F = K_H V_8 C_8/(V_{max} C_{max})$	K_F = 64
E	-	-	K_E = 40
D	T_5	$K_{D\ of\ E} = K_E V_5 C_5/(V_{max} C_{max})$	$K_{D\ of\ E}$ = 40
	T_7	$K_{D\ of\ F} = K_F V_7 C_7/(V_{max} C_{max})$	$K_{D\ of\ F}$ = 51
	T_6	$K_{D\ of\ G} = K_G V_6 C_6/(V_{max} C_{max})$	$K_{D\ of\ G}$ = 24
	T_5, T_6, T_7	$K_D = Max(K_{D\ of\ E}, K_{D\ of\ F}, K_{D\ of\ G})$	K_D = 51
C	T_{10}	$K_C = K_H V_{10} C_{10}/(V_{max} C_{max})$	K_C = 80
B	T_4	$K_B = K_C V_4 C_4/(V_{max} C_{max})$	K_B = 80
A	T_2	$K_{A\ of\ B} = K_B V_2 C_2/(V_{max} C_{max})$	$K_{A\ of\ B}$ = 38
	T_3	$K_{A\ of\ D} = K_D V_3 C_3/(V_{max} C_{max})$	$K_{A\ of\ B}$ = 51
	T_2, T_3	$K_A = Max(K_{A\ of\ B}, K_{A\ of\ D})$	K_A = 51

Table 14: Impact Calculation Table for the threat scenario

To calculate the risk of the threat scenario given in Figure 16 and Figure 17, we calculate the risk for each threat scenario based on the following rule;

- Risk of threat scenario that goes from A to H through impact stages (A->B->C->H) is given by

$$R(A, B, C, H) = \frac{K_H V_1 C_1 V_2 C_2 V_4 C_4 V_{10} C_{10}}{(V_{max} C_{max})^4} = 25$$

- Risk of threat scenario that goes from A to H through impact stages (A->D->F->H) is given by

$$R(A, D, F, H) = K_A V_{10} C_{10}/(V_{max} C_{max})$$

$$R(A, D, F, H) = K_H V_1 C_1 V_3 C_3 V_7 C_7 V_8 C_8/(V_{max} C_{max})^4$$

- The maximum Risk of the threat scenario is given by

$$R_{max} = \frac{K_A V_1 C_1}{(V_{max} C_{max})} = 33$$

Impact stages of the threat scenario	Threats	Risk Equation	Value
A, B, C, H	T_1, T_2, T_4, T_{10}	$R(A, B, C, H) = \frac{K_H V_1 C_1 V_2 C_2 V_4 C_4 V_{10} C_{10}}{(V_{max} C_{max})^4}$	24.567
A, D, F, H	T_1, T_3, T_7, T_8	$R(A, D, F, H) = \frac{K_H V_1 C_1 V_3 C_3 V_7 C_7 V_8 C_8}{(V_{max} C_{max})^4}$	32.768

A, D, G, H	T_1, T_3, T_6, T_9	$R(A,D,G,H) = \frac{K_H V_1 C_1 V_3 C_3 V_6 C_6 V_9 C_9}{(V_{max} C_{max})^4}$	15.36
A, D, E	T_1, T_3, T_5	$R(A,D,E) = \frac{K_E V_1 C_1 V_3 C_3 V_5 C_5}{(V_{max} C_{max})^3}$	25.6

Table 15: Risk calculation for possible threat scenarios

Table 15 calculates the risk for each threat scenario in Figure 16. The maximum risk is found to have a value of 33.

Threat ID	**Threat**	**Vulnerability**	**Capability**	**Impact**	Risk
CS2_T_1	Simple power analysis attack	4	4	51.2	0.33 (Medium)
CS2_T_2	Differential power analysis attack	4	4	51.2	0.33 (Medium)
CS2_T_3	Timing analysis attack	3	4	51.2	0.25 (LOW)
CS2_T_4	Electronic Eavesdropping	5	5	51.2	0.51(Medium)
CS2_T_5	Voice Impersonating	2	3	100	0.24 (LOW)
CS2_T_6	Electromagnetic emanation	3	3	51.2	0.18 (Low)
CS2_T_7	Fault Injection	4	4	51.2	0.33 (Medium)
CS2_T_8	Stealing the hardware and use it by the attacker	3	4	60	0.28 (LOW)
CS2_T_9	brute force attack on session key	4	1	51.2	0.08 (LOW)
CS2_T_10	breach of the underlying lattice hard problem	3	3	100	0.36 (MEDIUM)
CS2_T_11	breach of particular tweaks in the lattice scheme	3	4	100	0.48 (MEDIUM)
CS2_T_12	incorrect selection of lattice parameters	4	3	100	0.48(MEDIUM)
CS2_T_13	Exploitation of vulnerabilities in the key management protocol	4	3	100	0.48 (MEDIUM)
CS2_T_14	Implementation flaws	4	4	80	0.51 (MEDIUM)
CS2_T_15	Input validation errors	4	4	80	0.51 (MEDIUM)

Table 16: Risk assessment for the CoTS Public Safety Communication scenario

5.6 Countermeasures

Highest risk attacks are expected to be attacks on the implementation flaws and protocols.

Threat ID	Threat	Countermeasure
CS2_T_1	Simple power analysis attack	Remove the dependency between the data processed by the device and the secret data, and remove the dependency between the data processed by the device and its power consumption.
CS2_T_2	Differential power analysis attack	
CS2_T_3	Timing analysis attack	Achieve a constant computational time independent of the value of the secret key, and avoid conditional branches dependent on the secret key.
CS2_T_4	Electronic Eavesdropping	Use secure parameters of the LBC and cross check used protocols and implementation.
CS2_T_5	Voice Impersonation	This attack is out of scope
CS2_T_6	Electromagnetic emanation	Apply masking and randomisation of execution times. Mitigation techniques include applying physical shielding, metallisation layers on the device core or encapsulation of the device.
CS2_T_7	Fault Injection	Careful design of error handling modules and interrupt processing so that intermediate results are not leaked from the system.
CS2_T_8	Stealing the hardware and use it by the attacker	Out of scope
CS2_T_9	Brute force attack on session key	Choose key lengths carefully to provide adequate protection for the maximum lifetime of the session keys.
CS2_T_10	Breach of the underlying lattice hard problem	Perform thorough cryptanalysis of the scheme prior to deployment. Ensure that all assumptions are clearly communicated to end users.
CS2_T_11	Breach of particular modifications in the lattice scheme	Publish the scheme to allow cross checking by the cryptography community experts prior to deployment.
CS2_T_12	Incorrect selection of lattice parameters	Use the parameters recommended for higher levels of security than thought to be needed. Use parameters recommended by multiple independent sources.
CS2_T_13	Exploitation of vulnerabilities in the key management protocol	Strict adherence to well defined coding practices. Subject the protocol to cross checking to detect vulnerabilities and suggest remedies.
CS2_T_14	Implementation flaws	
CS2_T_15	Input validation errors	

Table 17: Countermeasures against the identified risks in COTS in Public Safety Communications

6 Privacy Preserving Municipal Data Analytics

6.1 High level view

6.1.1 System view

The architectural model of the municipal data analytics use case is illustrated in Figure 18. The Municipality is the primary actor within the model. Data is collected by the municipality through a variety of data sources, such as sensor networks, CCTV feeds, patient data, etc. The exact nature of the data and the methods used to collect it is dependent on the target of the investigation or research. A typical example might be a project researching the effectiveness of traffic management within a municipal environment. This can use a broad range of information from smart cars, video cameras, smart sensors and other sources.

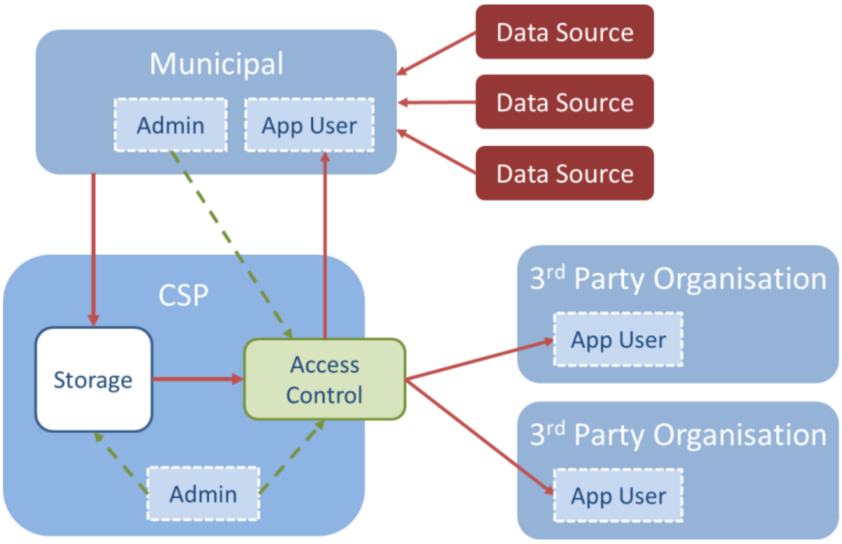

Figure 18 Municipal Data Analytics Architecture

The data collected is invaluable to the municipality investigating economic and social trends, public health, disease prevention and treatment, impact and effectiveness of technology, and many other areas. However, the ability to engage with external research institutes, academic and industry bodies, increase the potential to derive more comprehensive insights from larger data sets.

As Figure 18 shows, this engagement with 3rd party organisations is achieved by leveraging the resources of a Cloud Service Provider (CSP). There are slight variations to this model according to the specific implementations, but for the purposes of this document, it is sufficient to view the CSP as the actor responsible for the storage and access control to the municipal data. Administration of these components of the architecture can be performed either remotely from the municipality or from within the CSP infrastructure itself. Access control can be performed entirely by the municipality admin or delegated (in part) to CSP admin staff depending on the requirements of the scenario. Authorised 3rd party organisations can access the data via the CSP in order to perform analysis on the data sets. This analysis can either be performed on the CSP, so that the data does not leave the boundaries of the CSP, or locally, having retrieved the data from the CSP. For more detail on the specifics of the use case, refer to D9.1.

6.1.2 Components

The primary components in the municipal data analytics use case are:

- The Municipality – This is the entity that owns the data set for analysis. The municipality administrator is in charge of managing the data set, setting the access control policies and managing the keys (either locally or from the CSP).
- The CSP – This is the cloud provider hosting the municipal data. The following sub-components exist inside the CSP environment
    - Access Control Mechanism – Mechanism to provide the authentication and authorization processes for users wishing to gain access to the municipal data set.
    - Storage – The virtual storage (pooled from physical resources) on the CSP used to store and access the municipal data.
    - Analytics Application (optional) - Depending on the specific scenario, the analytics application may be hosted and run on the CSP.

The secondary components of the use case are:

- The 3rd Party Organisations - This is the research institute (industry or academic) participating in the analysis of the municipal data. Within these entities are two classification of user
    - Administrator: May be responsible for authorization of access by individuals in that organization to the municipal data set. This is dependent on the contractual relationships with the municipality and the CSP.
    - Application Users: These are end users that carry out the analytics process on the municipal data. They can be members of the municipality itself, a 3rd party industry partner or an academic institute.
- Data sources – Data can be recorded from a wide variety of sources, as outlined in in D9.1, for analysis. These sources range from video/camera feeds, utilities monitoring, traffic monitoring systems and distributed wireless sensor networks.
- Identity Federation – In order to achieve access to a central service (the data analytics application and/or municipal data access portal) across multiple organisations existing in separate trust domains, some form of identity federation is required so that credentials issued locally to each affiliated organization can be used to establish an access token for the service. The implementation of an identity federation system can exist with the Municipality or on the CSP, managed by the municipality.

6.2 Critical system assets

6.2.1 Assets description

1. Application layer data: This is the actual data-content communicated between application layers. This includes
    a. Personally Identifiable Information (PII) such as social security numbers, postal addresses, vehicle registration numbers, insurance numbers, etc.
    b. Secondary/Incidental data that may leak information about a person or organisation (e.g. GPS/location data, timetables/calendar information, schedules, etc.)
    c. Confidential Municipal data (e.g. sensor readings, video/image feeds, medical records, etc.)

2. Symmetric (session) keys: Used in encrypting the application layer data
3. Symmetric (session) key generation function and inputs: This includes the function used in generating the key and its inputs. Whilst the key generating function should not be a secret, if knowledge of the key generating function and its input values are wholly or partially acquired, it is possible to generate the key, or at least decrease the number of possibly generated keys.
4. Private keys (of public key pair): Private keys within a PKI, used for establishing session. There is no risk to the corresponding public key.
5. Public/Private key pair generation function and inputs: This includes the function used in generating the key and its inputs. Whilst the key generating function should not be a secret, if knowledge of the key generating function and its input values are wholly or partially acquired, it is possible to generate the key, or at least decrease the number of possibly generated keys.
6. Broadcast/Multicast keys: Keys used for broadcast of multicast encryption schemes
7. Broadcast/Multicast key generation function and inputs: This includes the function used in generating the key and its inputs. As with the generation of public/private key pairs, if knowledge of the key generating function and its input values are wholly or partially acquired, it is possible to generate the key, or at least decrease the number of possibly generated keys.
8. Random Number Generator: As part of the key generation process, a RNG may be used which must be adequately secured and generate suitably random numbers.
9. KMS: The protocol, processes and operations in place to manage the lifecycle of keys within the environment.
10. Authentication mechanism: This is dependent on the policies in place for Authentication, Authorisation and Accounting (AAA). Typically, at a base security level, passwords would be required to authenticate authorized users. Stronger authentication mechanism such as smartcards, tokens and biometrics can be used to provide better security. A compromise of these mechanisms would lead directly to a compromise of key material.
11. CSP infrastructure: The infrastructure of the CSP hosting the municipal data (and in certain scenarios managing the keys). This includes physical security, network security, access & control policies, storage security (backups, encryption, failover capability, etc.).
12. Municipality infrastructure: Servers, workstations and networking infrastructure within the municipality itself. This includes physical security, network security, access & control policies, storage security (backups, encryption, failover capability, etc.).
13. 3rd Party Organisation Infrastructures: Servers, workstations and networking infrastructure within the municipality itself. This includes physical security, network security, access & control policies, storage security (backups, encryption, failover capability, etc.).
14. Sensor network: One of the sources of municipal data may be through the use of a distributed network of sensors collecting data (e.g. traffic or pollution monitoring).

6.2.2 Assets summary

Reference	Asset

CS3_A_1	Application layer data
CS3_A_2	Symmetric (session) Keys
CS3_A_3	Symmetric (session) Keys Generation Function and inputs
CS3_A_4	Private keys (of public key pair)
CS3_A_5	Public/Private key pair generation function and inputs
CS3_A_6	Broadcast/Multicast keys
CS3_A_7	Broadcast/Multicast key generation function and inputs
CS3_A_8	Random Number Generator
CS3_A_9	Key Management System
CS3_A_10	Authentication mechanism
CS3_A_11	CSP infrastructure
CS3_A_12	Municipality infrastructure
CS3_A_13	3rd Party Organisation Infrastructures
CS3_A_14	Sensor Network

Table 18: Summary of Case Study 3 assets

6.2.3 Asset dependency diagrams

Figure 19 shows the assets dependency diagram for the top level assets (application layer data – CS3_A_1) in the municipal data analytics use case. Here, the hierarchy of assets are illustrated such that it shows the dependency that each asset has on related assets. A vulnerability in a subordinate asset can lead to a vulnerability in, or point of attack against, an asset that depends on it. The relationship between assets in the assets dependency diagram, will be further analysed in subsequent sections.

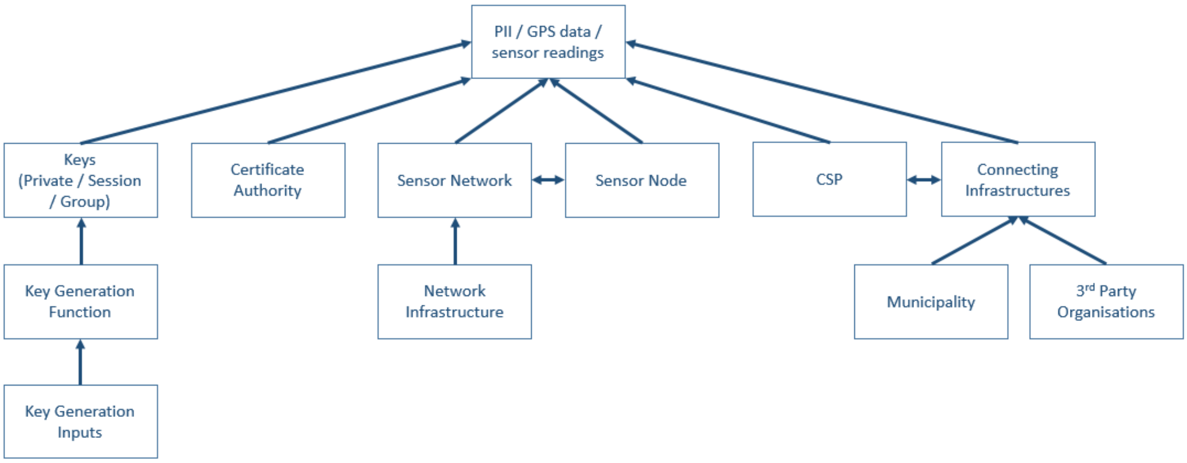

Figure 19: Asset dependency diagram for the municipal data assets

6.3 Attack points

3rd Party Organisations: Authorised users associated with a 3rd party organisation can be granted access (partial or full depending on the scenario) to the municipal data. Key management protocols are implemented to maintain a level of access control to the data. Analysis of the data can be exclusively performed on the CSP, preventing any confidential data from leaving the boundary of the system. However, it is possible for a user to simply take screenshots on their workstation of any displayed data. There may also be situations that allow users in 3rd party organisations to download a local copy of the data for analysis.

There must exist, as a base level, a degree of trust between the municipality and the 3rd party organisation that proper processes are in place for the management of personnel (recruitment processes, code of conduct, etc.), network security and physical security so that a breach may not occur from this location. 3rd Party Organisations should be aware of and make provisions for the prevention of social engineering attacks and physical attacks.

CSP: Municipal data stored and processed using the resources of a CSP are exposed to new vulnerabilities not present in private databases stored at a municipals local facilities. There are additional considerations with respect to data protection that need to be addressed. When data is transferred to a CSP, the owner of the data loses control over where the data is physically stored and how it is protected.

Ensuring data is stored on servers within specific regions can be critical due to differing legislation or regulations with regard to how data can be processed and accessed. The data owner must also have assurances that on termination of the contract with the CSP, the data (and any backups) be permanently destroyed. Policies for access control, encryption, backups, etc. can also be agreed on to ensure the CSP meets the security requirements of the data owner. A service level agreement (SLA) between the data owner and the CSP can help to mitigate any concerns over the policies implemented with respect to the data.

Sensor Nodes: Data can be collected for the municipality through the use of a spatially distributed network of wireless sensor nodes. These nodes are application specific, are typically resource constrained and are potentially vulnerable to physical tampering. An attack on a sensor node could lead to manipulated data being transmitted to the central database of municipal data or an attacker could even retrieve/intercept actual recorded data from the node and other connected nodes.

Sensor Network: A wireless sensor network is composed of a collection of spatially distributed autonomous sensor nodes. Typically, there is a sink node to which all collected data by the other nodes is forwarded. For example, should an attacker compromise the sink node in a wireless sensor network (WSN), this would enable them to intercept data from other nodes.

Key Generation & Distribution: The selection of key generation functions and distribution protocols is critical to provide strong resistance against attacks on the keys. Cryptographically secure pseudo-random number generators should be used in the generation of any seed values or other random input. Sufficiently large keys should also be generated such that attacks on the key are not feasible by attackers.

Certificate Authority: A trusted CA should be used in a PKI to provide confidence in any certificates issued within the system.

Communication channel: The service is intended to be accessible over public network (i.e. the Internet) and therefore all such public communication channels should be secured with TLS 1.2.

6.4 Threat list

6.4.1 Threat environment

In the municipal data analytics scenario, the confidentiality and integrity of the data is exposed to a much larger attack surface than if it were stored and processed within traditional, in-house, data centres. By the very nature of cloud computing and CSPs, the data is on an Internet facing service. The level of security provided by public cloud providers such as Amazon or Azure can vary to the point where responsibility for security of Software as a Service (SaaS) is almost entirely with the data owner. While Amazon Web Services (AWS), for example, offers many security features within their catalogue of services, these must be selected and implemented by the cloud consumer (in this scenario, the municipality). As well as issues around the correct configuration of security features, storing and processing sensitive data within a CSP exposes the data to actors (CSP staff, attacker exposing weakness in CSP, attacker monitoring traffic to/from the CSP) that otherwise, would have no attack vector to exploit.

With respect to the protection of the confidentiality of the municipal data, a vital tool used to prevent a compromise is encryption. Encryption, when properly used, prevents unauthorised actors from accessing the data or modifying the data in such a way that it remains undetected to the data owner. Current recommended encryption algorithms and key sizes make attacks on encrypted data infeasible. As the move is made towards algorithms and keys resistant to post quantum computing attacks, so too will appropriate protocols and key sizes be selected to prevent direct attacks on the encrypted data. However, due to the difficultly in performing such attacks, a more favourable route for attackers is to target the keys themselves. This is of particular concern in the municipal data scenario as the keys are in a potentially compromised situation. With the exception of use case 1 in D9.1, the other two use cases require that the keys be processed and optionally stored at the CSP. This presents a key focus area for attackers to exploit a weakness in the CSP (whether physical, logical or human) in order to gain access to the keys.

6.4.2 Physical threats

Risks related with physical attacks, for the case of privacy preserving municipality data, seems to be limited to timing attacks. It is in fact extremely unlikely that an adversary can have access to the details needed to mount a power analysis attack on a cloud based system. Similarly, it seems extremely difficult for an adversary to inject a fault during the computation. Timing attacks instead were previously carried out successfully across networks and in cloud systems, thus this type of physical attack appears to be more realistic in a cloud based system.

Keystroke timing attacks have been shown in traditional computing environments to be effective in attempting to steal a target's credential information (i.e. login passwords) [15] . The success of this class of attack is based on the application of advanced statistical analysis techniques on timing information monitored over a network. The attack proposed in [15] demonstrated the effectiveness of this by gathering significant information on the targets keystrokes in a Secure Socket Shell (SSH) session.

Moving to the cloud environment, the objective of the attack is the same, to record the keystroke timing information while the target user is authenticating. This timing information can then be used to recover the password. In [16], Ristenpart et al. demonstrated an evolution of this idea by launching co-resident Virtual machines (VM) to the target VM in order to measure the cache-based loads while the target enters their credential information. Using simple password authentication mechanisms in such an environment can leave systems and service potentially vulnerable to attack.

The possibility of an attacker launching VM instances that are co-resident to the target VM opens the potential other class of side channel attacks. Side channels such as the CPU cache can leak information due to the shared nature of the physical resources of the server. Zhang et. al. successfully extracted the ElGamal decryption key from the target VM (running on a Xen hypervisor) [17].

6.4.3 Logical threats

The immediate risk associated with a weakening of the security of the lattice based scheme in the context of the municipal data analytics use is low. Such a development would contribute more towards performance degradation as a remediation process is put into effect to replace and distribute affected keys. If the break in the scheme is significant enough that the symmetric keys used to encrypt the municipal data at rest are vulnerable, there is the risk that the data could be accessible by unauthorised users and would need to be re-encrypted. In the longer term, a significant breakthrough in attacking or breaking lattice based schemes would require that users of the schemes move away to more secure schemes.

A more tangible threat that needs to be considered is to the crypto period selected for the lattice based keys. This is the length of time during which a key is in use. Intuitively, the longer the crypto period is, the more vulnerable the data is to a sustained attack (e.g. brute force). Therefore, limits should be set on the keys for how long they can be used to protect the data. The lifetime of municipal data stored on a CSP for retrieval/analysis by authorised collaborating parties should be clearly defined and appropriate crypto periods and key sizes selected to protect the data for the duration of the lifetime.

Closely linked to the crypto period for the keys, is the issue of key management. As keys reach their end of life, new keys must be generated and distributed, re-encrypting the protected data on the CSP. Outside of expiration of keys, key revocation process are needed to prevent invalid keys (due to compromise from attacker, accidental distribution, employee reassigned or made redundant, etc.) from acting as a valid key. Improperly implemented key management processes can lead to vulnerabilities in the overall scheme. Therefore, care should be taken when defining key management protocols for lattice based schemes, many of which will likely be based on traditional key management approaches. A simple mapping from traditional protocols to use lattice based algorithms and keys in place of, for example, algorithms and keys for RSA may result in potential vulnerabilities being overlooked.

Relative to RSA, the keys used for lattice based schemes are larger. While bandwidth and storage resources are not considered to be restricted in a CSP environment, there is historically a drive to use the smallest key sizes possible while providing adequate security for the data/communications. Finding the appropriate balance between the crypto period, key size and key management processes is a crucial area in determining the effectiveness and feasibility of any lattice based scheme.

6.4.4 Human threats

The threat to lattice based schemes employed in the municipal data analytics use case is a particular concern due to the number of users (municipal employees, 3rd party employees, CSP employees) with access to the system and due to the scope of access available to those users. Each of these users are a potential target for attackers who may attempt phishing attacks, social engineering, or otherwise hack an account. While the majority of these attacks extend beyond lattice based schemes, the threat still exists and established processes for mitigating these threats apply (adequate employee training, email firewalls, IPS/IDS, etc.).

Within the context of this use case, the most effective means to mitigate the risk associated with human threats is to ensure appropriate access control mechanisms are in place to prevent any unauthorised user from gaining access to the system. This approach such be applied from top to bottom as even breach of a low level account can be used to attack the system as an attacker will attempt to escalate their privileges in order to gain access to the critical servers in the infrastructure (including the key server).

6.4.5 Summary of threats

The threats described above are summarised in the following table.

Threat ID	Threat	Assets at risk
CS3_T_1	Timing attack from co-tenant VM on CSP	CS3_A_10, CS3_A_11
CS3_T_2	Brute force attack on session key	CS3_A_2
CS3_T_3	Breach of the underlying lattice hard problem	CS3_A_2, CS3_A_3, CS3_A_4, CS3_A_5, CS3_A_6, CS3_A_7
CS3_T_4	Targeted weakness of random number generator	CS3_A_3, CS3_A_5, CS3_A_7, CS3_A_8
CS3_T_5	Incorrect selection of lattice parameters	CS3_A_3, CS3_A_5, CS3_A_7
CS3_T_6	Vulnerabilities in the key management system	CS3_A_9
CS3_T_7	Vulnerabilities in the PKI infrastructure	CS3_A_9
CS3_T_8	Physical attack on CSP infrastructure	CS3_A_11
CS3_T_9	Physical attack on municipality and/or 3rd party organisation infrastructures	CS3_A_12, CS3_A_13
CS3_T_10	Attack on network communications	
CS3_T_11	Attack on distributed sensor nodes and/or network	

Table 19: Summary of Threats for Municipal Data Analytics Case Study

6.5 Risk analysis and Countermeasures

This section will attempt to quantify the threats listed in Section 6.4 using the methodology described in Section 3.3.

6.5.1 Risk analysis

The primary objective of the threats summarised in the previous section is the compromise of the keys used to protect the municipal data (through encryption and/or digital signatures). This includes the symmetric keys used to perform the encryption/decryption of the data and the asymmetric keys used in protocols for establishing shared symmetric keys, KEKs, DEKs, session keys, etc. The consequence of a successful attack on the keys or on the key management system responsible for their generation and management would leave the municipal data vulnerable to manipulation and theft.

Figure 20 & Figure 21 illustrate the calculation of the risk associated with attacks on the private keys and on the key management system respectively. Both attack vectors highlight the criticality of the security provided by the keys. A compromise or weakness in the generation of the keys or the system responsible for the management of the keys through their lifecycle has the direct consequence of a complete failure of the protection of the data.

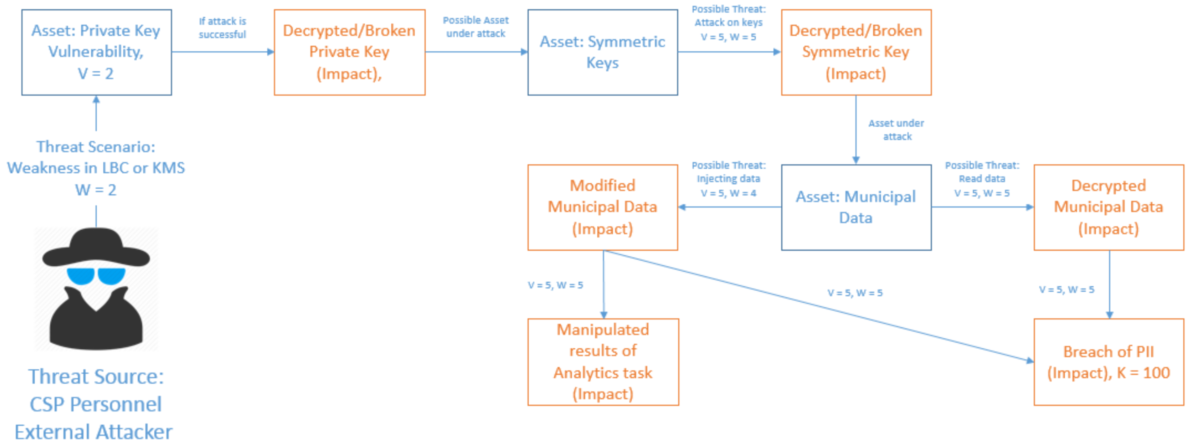
Asset: Private Key Vulnerability, V = 2
If attack is successful
Decrypted/Broken Private Key (Impact),
Possible Asset under attack
Asset: Symmetric Keys
Possible Threat: Attack on keys V = 5, W = 5
Decrypted/Broken Symmetric Key (Impact)
Asset under attack
Threat Scenario: Weakness in LBC or KMS W = 2
Modified Municipal Data (Impact)
Possible Threat: Injecting data V = 5, W = 4
Asset: Municipal Data
Possible Threat: Read data V = 5, W = 5
Decrypted Municipal Data (Impact)
V = 5, W = 5
V = 5, W = 5
V = 5, W = 5
Manipulated results of Analytics task (Impact)
Breach of PII (Impact), K = 100
Threat Source: CSP Personnel External Attacker

Figure 20: Municipal Data Analytics: Risk Calculation 1

Figure 20 considers a fundamental failure in the generation of the public/private key pair, leading to the compromise of the private component which can subsequently be used to recover the symmetric keys protecting the data. The level of risk of this threat occurring is low and as further research is applied to the problem, confidence in the strength of lattice based cryptography will grow. By comparison, Figure 21 deals with a common threat that transcends the underlying basis for the cryptography. When direct attacks on the cryptographic algorithms are computationally infeasible, it is therefore more appealing to attackers to target the keys themselves. A robust key management process should be deployed to provide sufficient protection against attacks while also offering the ease of use required of the scenarios described in D9.1 for the municipal data analytics use case.

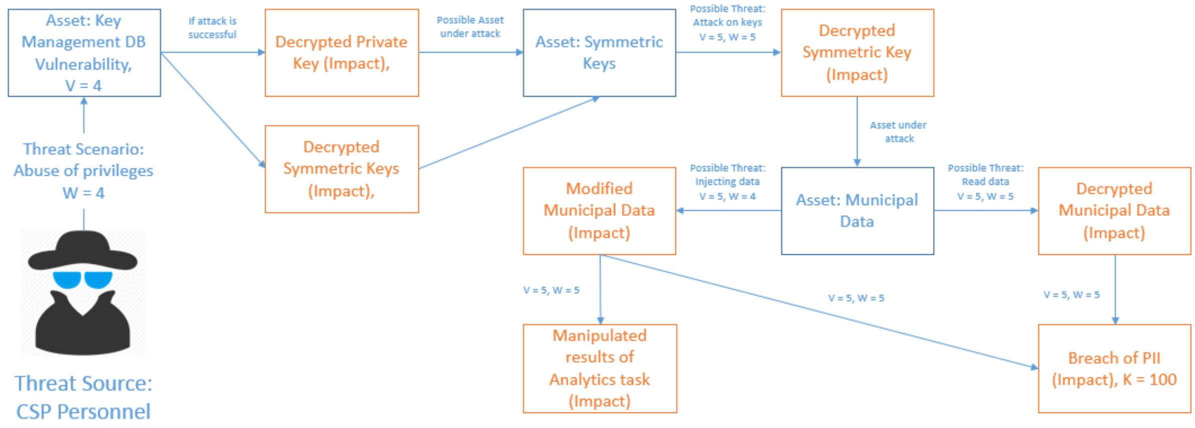
Asset: Key Management DB Vulnerability, V = 4
If attack is successful
Decrypted Private Key (Impact),
Possible Asset under attack
Asset: Symmetric Keys
Possible Threat: Attack on keys V = 5, W = 5
Decrypted Symmetric Key (Impact)
Decrypted Symmetric Keys (Impact),
Asset under attack
Threat Scenario: Abuse of privileges W = 4
Modified Municipal Data (Impact)
Possible Threat: Injecting data V = 5, W = 4
Asset: Municipal Data
Possible Threat: Read data V = 5, W = 5
Decrypted Municipal Data (Impact)
V = 5, W = 5
V = 5, W = 5
V = 5, W = 5
Manipulated results of Analytics task (Impact)
Breach of PII (Impact), K = 100
Threat Source: CSP Personnel

Figure 21: Municipal Data Analytics: Risk Calculation 2

As in the other use cases, in order of the most severe impact to the least severe, the possible outcomes affecting the municipal data in the event of a successful attack are:

- Compromising the confidentiality – This would allow an attacker to steal municipal data
- Compromising the integrity – This would allow an attacker to manipulate municipal data
- Compromising the availability – This would allow an attacker to disrupt access to the data hosted on a CSP

Table 20 below examines the threats summarised in the previous section with respect to the overall calculated risk (as a function of the vulnerability, capability and impact described in Section 3.3.

While the impact measurement for the majority of the threats is quite high, the overall level of risk are classified as MEDIUM or LOW due to either the inability of attackers to carry out such attacks or the hardness of the problem, reducing the chance of a vulnerability existing.

The threats with the highest risk measurement registered for the municipal data analytics case study are CS3_T_2, CS3_T_5, CS3_T_8 and CS3_T_9. A brute force attack on the session keys has a very high impact and with the resources available through cloud computing, attackers have the means to carry out such an attack. However, the vulnerability here is quite low unless some significant reduction in the strength of lattice based cryptography can be made. An attack vector with more potential for success in recovering key data is to attack the KMS as noted by CS3_T_6. The proper implementation of a KMS hardened against attacks is crucial for the protection of the keys. Failure here would have a detrimental impact on the overall security.

CS3_T_8 and CS3_T_9 both related to physical attacks on the datacentres, whether the datacentre is located at the CSP or at one of the participating organisations. CSPs can be selected to offer significant security benefits over traditional datacenters and therefore the vulnerability score here is slightly lower. However, CSPs are not bullet proof and introduce their own set of attack vectors as discussed in Section 6.4. It should be noted for CS3_T_9, that a weakness is any one of the participating organisations would affect the overall risk rating.

Threat ID	Threat	Vulnerability	Capability	Impact	Risk
CS3_T_1	Launching a timing attack from a coresident VM	3	2	3	18/125 = 0.14 (LOW)
CS3_T_2	Brute force attack on session key	2	4	5	40/125 = 0.32 (MEDIUM)
CS3_T_3	Breach of the underlying lattice hard problem	3	1	5	15/125 = 0.12 (LOW)
CS3_T_4	Compromise of the random number generator	2	1	4	8/125 = 0.06 (LOW)
CS3_T_5	Incorrect selection of lattice parameters	4	2	4	32/125 = 0.26 (LOW)
CS3_T_6	Vulnerabilities in the key management system	4	3	5	60/125 = 0.48 (MEDIUM)
CS3_T_7	Vulnerabilities in the PKI infrastructure	2	3	4	24/125 = 0.19 (LOW)
CS3_T_8	Physical attack on CSP infrastructure	3	4	4	48/125 = 0.38 (MEDIUM)
CS3_T_9	Physical attack on municipality and/or 3rd party organisation infrastructures	4	4	4	64/125 = 0.51 (MEDIUM)
CS3_T_10	Attack on network	2	2	3	12/125 = 0.10

	communications				(LOW)
CS3_T_11	Attack on distributed sensor nodes and/or network	4	3	2	24/125 = 0.19 (LOW)

Table 20: Risk Calculation for Threats against Municipal Data Analytics Case Study

6.5.2 Countermeasures

In this section, we propose some security strategies and preventative measures that should be adopted in order to mitigate the risks posed by the threats listed in the previous sections.

Two key areas have been identified as risks for the municipal data analytics case study: physical attack on the datacentres; attack on the KMS. To protect the CSP datacentre, appropriate security policies for access control (physical and logical), segregation of duties, personnel hiring, configuration/patch management, etc. should be in place. Such policies are typically agreed upon between the CSC and the CSP through the signing of a Service Level Agreement that can include security features. These controls are beyond the scope of what SAFEcrypto can achieve but are prerequisite for assuring the security of the system. Areas that SAFEcrypto can address however is the efficient and effective use of encryption and key management to limit the access an attacker may have to the municipal data in the event of a compromise. For example, there is scope within the various environment configurations described in D9.1 to manage how much access the CSP has to the KMS.

The risks present at a CSP are potentially greater at the 3rd party organisations where the dedicated resources for security available at a CSP may not be in place. In a scenario with multiple participating organisations with access to keys to decrypt the data or to decrypt the KEK, a vulnerability in just one of those organisation's datacentres would potentially result in a data leak. Therefore, it is important that all participating organisations adhere to security best practices. By restricting the processing of data and management of the keys to the CSP only, the potential for such weaknesses can be reduced as neither keys nor unencrypted data ever enters their datacentres.

The previous section highlighted the risks associated with attacks on the strength of the lattice based cryptography and on the keys. A compromise of either will allow an attacker to access the municipal data. The underlying security of the lattice based cryptography is considered secure, but further investigation into the strength of the schemes should be conducted through academic and industrial research. This continued scrutiny of the lattice based schemes will increase the confidence in the underlying strength and resilience of the cryptography to attack.

A KMS should provide the efficient management of keys through their lifecycle. This includes stages for the generation of the keys, their distribution, usage, active period and destruction. While one of the objectives of this is to efficiently use keys to minimise communication overhead relaying key data and to manage complex interactions of multiple entities interacting with shared data sets, the KMS is also responsible for setting the parameters (key sizes, crypto periods, protocols, etc.) that affect the security of the system. Careful selection of these parameters can ensure the keys (and the data they protect) remain protected. D8.1 examines in more detail, the selection of protocols and other parameters for a KMS with a particular focus on the requirements of the case studies. An additional consideration here is the control of access to the KMS so that unauthorised users cannot gain any control over the keys.

The countermeasures are summarised in Table .

Threat ID	Threat	Countermeasure
CS3_T_1	Launching a timing attack from a coresident VM	Enforce policies at CSP to separate resource usage such that attackers cannot launch coresident VMs.
CS3_T_2	Brute force attack on session key	Careful selection of parameters to ensure high security per bit.
CS3_T_3	Breach of the underlying lattice hard problem	Publish the scheme openly for cryptanalysis by a wide range of experts. Maybe offer a prize for breaking the scheme, before it is deployed, to encourage wide-spread cryptanalysis.
CS3_T_4	Compromise of the random number generator	Careful selection of entropy source and random number generator implementation.
CS3_T_5	Incorrect selection of lattice parameters	Use the parameters recommended for higher levels of security than thought to be needed. Use parameters recommended by multiple independent sources.
CS3_T_6	Vulnerabilities in the key management system	Use well established protocols wherever possible. Submit modifications to cryptographic community for expert analysis.
CS3_T_7	Vulnerabilities in the PKI infrastructure	Use well established protocols wherever possible. Submit modifications to cryptographic community for expert analysis.
CS3_T_8	Physical attack on CSP infrastructure	Physical security and policy enforcement. Bind security requirements to SLA with CSP.
CS3_T_9	Physical attack on municipality and/or 3rd party organisation infrastructures	Physical security and policy enforcement. Reduce impact of such attacks by heavier reliance on CSP for handling data processing and the KMS.
CS3_T_10	Attack on network communications	Physical security and use of standard communications protocols.
CS3_T_11	Attack on distributed sensor nodes and/or network	Harden sensors against tampering or physical attacks. Lightweight communication protocols may be more susceptible to attack, use recommended protocols for communications.

Table 21: Summary of Countermeasures for Municipal Data Analytics Case Study

7 Summary and Conclusions

This deliverable analysed the risk of potential threats and vulnerabilities associated with LBC architectures for the three case studies described in D9.1; Satellite Key Management (SKM-CS), CoTS in Public Safety Communications (CPSC-CS), and Privacy Preserving Municipal Data Analytics (PPMDA-CS). The deliverable produced an overview of three families of threats; Physical Side Channel (PSC), Logical, and Human. These attacks were investigated in more detail for each case study.

For each case study, the critical assets were identified and the dependencies between the assets were highlighted. Assets are classified into goal assets and key (root) assets, where a compromise of the latter will make the former more susceptible to attacks. After identifying assets, attack points were highlighted for each case study.

The threat environment is different for each case study, which affects the risk analysis factors, especially threat exposure, and countermeasures. In SKM-CS, the physical accessibility is limited prior to launching the satellite and impossible after launching. For example, while a PSC attack based on power analysis could be possible prior to launch, such an attack would be impossible to conduct once launched. However, timing and fault attacks should still be considered even after the Satellite deployment. In the SKM-CS, the ability to control the level of impact of a successful attack is also very limited, with a worst case scenario resulting in the loss of control over the Satellite for an extended period of time during which the Satellite could be irrevocably destroyed. The limited deployment numbers for the SKM-CS leads to a less cross-checking of the protocols implemented and consequently less opportunity to detect faults and errors.

The environment in the CPSC-CS is characterised by more accessible hardware, software components and communication channels. This allows a wider exposure to a wider number of threats and attacks. On the other hand, this also allows a wider community to scrutinize and cross check the system security, which leads to a better detection of faults and vulnerabilities, and consequently to the development of effective countermeasures and a more resilient secure architecture implementation. The impact of a successful attack in the CPSC-CS is easier to contain than in the SKM-CS as patches, new protocols and/or new keys can be deployed with relative ease. Therefore it is more critical to get the design implementation right in the SKM-CS prior to deployment and launching. In the CPSC-CS it is important to maintain a very active security monitoring and response team before and during deployment.

Finally, the PPMDA-CS environment operates in the Cloud, utilises the benefits of virtualised resources to facilitate collaborative analysis on municipal data. Within a datacentre setting the threat of physical attacks is limited primarily to side channel attacks launched from co-tenant virtual machines on the same physical resource. Other threats relevant to this use case involve the logical threats to the underlying strength of LBC that is common across all use cases. In particular, due to the collaborative nature of the use case operating on sensitive data, there is a particular emphasis on the management of keys to restrict access to the data. This use case faces a particular challenge in ensuring that access to the data set and the KMS within the various datacentres is restricted such that the CSP (or partner organisation) employees may not access the data.

Threats for each case study were identified and a risk analysis was carried out. The risk calculation method used is explained in Chapter 3. For the scope of the deliverable, the risk is calculated as a function of the capability of the attacker to carry out a particular threat, the vulnerability of the system under attack to the threat, and the impact of threat if it is mounted successfully. The risk calculation is applied for single attack threats and scenario threats, where a sequence of attacks may be carried out.

As part of the risk calculation, many assumptions have been made when assigning vulnerability, capability and impact scores. These scores were determined as a 'best effort' endeavour due to the early stage of the project. However, the risk analysis highlighted concerns mainly regarding logical threats such as key management and in particular modifications of the LBC implementation which may lead to further vulnerabilities. From a case study perspective, it is noted that Human attacks are likely to be a particular concern, whether as a standalone threat or as a part of a threat scenario. For the scope of this project, human attacks are not dealt with directly. However, through effective key management to protect data at-rest and in-transit the impact of malicious human actions can be reduced. The risks scores of threats are found to be between Low to Medium. However, as explained before the scores are based on assumptions that need to be reviewed when the design and implementation of the LBC system is at a more advanced stage.

In conclusion, this document delivers a deeper understanding of the risks and threat environment for each case study. This information will feed into the other work packages responsible for designing and implementing hardware and software solutions and will influence their technical decision making. By understanding and prioritising threats in each case study, a more secure and robust LBC based solution can be realised at project end.

8 References

[1] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. Power Analysis Attacks: Revealing the Secrets of Smart Cards. Advances in Information Security. Springer, New York, 2007. Xvii

[2] Paul Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Neal I. Koblitz, editor, Advances in Cryptology—CRYPTO '96, volume 1109 of Lecture Notes in Computer Science, pages 104–13. Springer, Berlin, September 1996.

[3] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, Cryptographic Hardware and Embedded Systems—CHES 2002,volume 1965 of Lecture Notes in Computer Science, Berlin, August 2002. Springer.

[4] Kai Schramm, Thomas J. Wollinger, and Christof Paar. A new class of collision attacks and its application to des. In Thomas Johansson, editor, FSE, volume 2887 of Lecture Notes in Computer Science, pages 206–222. Springer, 2003.

[5] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael Wiener, editor, Advances in Cryptology—CRYPTO '99, volume 1666 of Lecture Notes in Computer Science, pages 398–412. Springer, Berlin, August 1999.

[6] Dakshi Agrawal, Josyula R. Rao, and Pankaj Rohatgi. Multi-channel Attacks. In Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors, Cryptographic Hardware and Embedded Systems—CHES 2003, volume 2779 of Lecture Notes in Computer Science, pages 2–16, Cologne, Germany, September 2003. Springer

[7] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, Cryptographic Hardware and Embedded Systems—CHES 2004, volume 3156 of Lecture Notes in Computer Science, pages 16–29. Springer, Berlin, September 2004.

[8] Thomas S. Messerges. Using second-order power analysis to attack DPA resistant software. In Çetin Kaya Koç and Christof Paar, editors, Cryptographic Hardware and Embedded Systems—CHES 2000,volume 1965 of Lecture Notes in Computer Science, pages 231–37, Berlin, August 2000. Springer.

[9] Jason Waddle and David Wagner. Towards efficient second-order power analysis. In Marc Joye and Jean-Jacques Quisquater, editors, Cryptographic Hardware and Embedded Systems—CHES 2004, pages 1–15, Cambridge, MA, USA, August 2009.

[10]Elisabeth Oswald, Stefan Mangard, Christoph Herbst, and Stefan Tillich. Practical Second-Order DPA Attacks for Masked Smart Card Implementations of Block Ciphers. In David Pointcheval, editor,Topics in Cryptology — CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006, volume 3860 of Lecture Notes in Computer Science, pages 192–207, San Jose, CA, USA, February 2006. Springer.

[11]Hagai Bar-El, Hamid Choukri, David Naccache, Michael Tunstall, and Claire Whelan. The sorcerer's apprentice guide to fault attacks. Proceedings of the IEEE, 94(2):370–382, 2006.

[12]Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of eliminating errors in cryptographic computations. Journal of Cryptology, 14(2):101–119, 2001. 25

[13]Gilles Piret and Jean-Jacques Quisquater. A Differential Fault Attack Technique against SPN Structures, with Application to the AES and KHAZAD. In Cryptographic hardware and embedded system, CHES 2003: 5th international workshop, Cologne, Germany, September 8-10, 2003: proceedings, page 77. Springer Verlag, 2003.

[14]Chong Hee Kim and Jean-Jacques Quisquate. New Differential Fault Analysis on AES Key Schedule: Two Faults Are Enough. In Smart Card Research and Advanced Applications: 8th Ifip Wg 8.8/11.2 International Conference, Cardis 2008, London, UK, September 8-11, 2008, Proceedings, Lecture Notes in Computer Science, page 48. Springer, 2008

[15]Dawn Xiaodong Song, David Wagner and Xuqing Tian. Timing Analysis of Keystrokes and Timing Attacks on SSH. Proceedings of the 10th Conference on USENIX Security Symposium - Volume 10. 2001.

[16]Thomas Ristenpart, Eran Tromer, Hovav Shacham and Stefan Savage. Hey, You, Get off of My Cloud: Exploring Information Leakage in Third-party Compute Clouds. Proceedings of the 16th ACM Conference on Computer and Communications Security, pages 199-212. 2009.

[17]Yinqian Zhang, Ari Juels, and Michael K. Reiter and Thomas Ristenpart. Cross-VM Side Channels and Their Use to Extract Private Keys. Proceedings of the 2012 ACM Conference on Computer and Communications Security, pages 305-316. 2012.

[18]Committee on National Security Systems (CNSS) Glossary. CNSSI No. 4009, April 2015

[19]Minimum Security Requirements for Federal Information and Information Systems. Federal Information Processing Standards Publication, FIPS-PUB 200, March 2006.

[20]Guide for Applying the Risk Management Framework to Federal Information Systems NIST SP 800-37 Revision 1, February 2010.

[21]IETF RFC 6347, "Datagram Transport Layer Security Version 1.2".

[22]IETF RFC 5238, "Datagram Transport Layer Security (DTLS) over the Datagram Congestion Control Protocol (DCCP)".

[23]IETF RFC 6083, "Datagram Transport Layer Security (DTLS) for Stream Control Transmission Protocol (SCTP)".

[24]IETF RFC 5764, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)".

[25]IETF RTCWEB "WebRTC Security Architecture", March 2015.

[26]Fundamentals of Capabilities-based Attack Tree Analysis. Amenaza Technologies Limited

[27]Risk Management, Information Security. Hans Georg Schaathun, http://www.computing.surrey.ac.uk/teaching/2010-11/comm037/05threat/handout.pdf

[28]The Open Web Application Security Project https://www.owasp.org/index.php/Main_Page, accessed on 16th February 2016.

[29]Introduction to Public Key Technology and the Federal PKI Infratsucture, NIST SP 800-32

[30]https://www.himmo-scheme.com/ accessed on 17th February 2016

[31]European Union Ageny for Network and Information Security, ENISA https://www.enisa.europa.eu/activities/risk-management/current-risk/risk-management-inventory/glossary , accessed 18th February 2016

[32]Guide for Mapping Types of Information and Information Systems to Security Categories. NIST 800-600 Volume I, August 2008http://csrc.nist.gov/publications/nistpubs/800-60-rev1/SP800-60_Vol1-Rev1.pdf , accessed on 18th February 2016

[33]http://www.WebRTC.org/architecture, accessed on 18th February 2016 inventory/glossary , accessed 18th February 2016

[34]Joye, Marc, "Elliptic curves and side-channel analysis", ST Joournal of System Research, pages 17-21, 2003

[35]Lattice-Based Cryptographic Architectures, SAFEcrypto Deliverable 6.1

[36]State-of-the-Art in Physical Side-Channel Attacks and Resistant Technologies, SAFEcrypto Deliverable 7.1. Available from project website www.safecrypto.eu

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 644729

End of Document