Assessing Cryptography as a Service

Lucy Potter

Submitted as part of the requirements for the award of the
MSc in Information Security
at Royal Holloway University of London

Information Security Group
Royal Holloway, University of London
August 2024

Project Submission Declaration

Project Title	Assessing Cryptography as a Service
Student Number	100996199

I declare that this dissertation is all my own work, and that I have acknowledged all quotations from the published or unpublished works of other people.

I declare that I have also read the statement on plagiarism in the General Regulations for Awards at Graduate and Masters levels for the MSc in Information Security and in accordance with it I submit this project report as my own work.

Signed:

Dated: 27 August 2024

Abstract

Cloud Service Providers have in recent years started to offer services based in the use of cryptography. This notion of Cryptography as a Service (CaaS) has received some study however definitions and use cases are not fully agreed upon and do not always reflect what security services are actually being offered by providers. An organisation considering whether to use a CaaS offering may have limited guidance available depending upon what service they are buying, and no real consideration has given to how a service like this can be properly found fit for purpose by a potential CaaS customer.

This project carried out a search for CaaS providers and their offerings were reviewed and then characterised into a taxonomy. Three important criteria that a customer might use when assessing an outsourced service are those of adherence to standards, trust in an external provider and quality management of an outsourced service. These three criteria were analysed and the applicability of using them to assess a CaaS was discussed, with applicable methods and frameworks proposed. The results of the supplier search were then revisited with these three criteria in mind and any evidence of aligning with the proposed evaluation criteria was captured.

The findings of the project were that CaaS offerings could be far more varied in scope than that shown in academic literature, and that while some use cases (such as encrypted cloud storage) were given plenty of attention from the academic community, some others had not been well studied. Providers used a variety of terms and vaguely cryptography related language to describe offerings which have been found to be extremely flawed. Well known management standards and trust frameworks were followed by a large number of providers however a number obscured their designs, did not seek third party certification and used misleading language to convey trustworthiness as a provider. Quality management for CaaS is somewhat challenging, and current software service quality models lack metrics that can be properly validated by a customer.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Context

Cloud computing has allowed individuals and companies to make use of easily available, rapidly scalable computing and storage resource with little more needed on the part of the client than a reliable internet connection. Customers of cloud services pay for computing as needed, requiring no up-front costs, and often outsourcing the burden of procuring, building, configuring and managing computing resources entirely to the cloud services provider. This notion of a customer buying resources from a provider has been described in the ”as a Service” model, with the most commonly encountered cloud services being those of Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [73].

The increased scale of cloud service providers (CSPs) has allowed them to rapidly add new products and services to their offerings, as well as enabling third party providers to use CSP infrastructure to easily and cheaply roll out new products to new customer markets. This proliferation of rapid product innovation which is easily accessible over the internet has given rise to the ‘as a Service’ term being applied to many other offerings, with the moniker “X as a Service” (XaaS) being commonly used to describe how almost any product now can be made available as a managed service beyond the traditional descriptors of IaaS, PaaS and SaaS [46]. A managed service is one where the bulk of the design, operation, administration and maintenance of any technology is carried out by the service provider on behalf of the customer, thereby allowing the customer to concentrate on their core business.

Security products have also grown in complexity and variety, and increasingly have to cater towards a client base that conducts a large part of their business either connected to the internet or wholly hosted by CSP. Following other product offerings into the XaaS paradigm, security providers now offer Security as a Service (SECaaS) to potential customers. These SECaaS offerings are hugely varied in nature, covering products such as firewalls, end device management, vulnerability scanning, email security and many more. SECaaS providers claim a value proposition similar to that offered by the first cloud providers - namely that clients can outsource and easily scale the build, configuration and management of security tools, reducing the need to recruit or maintain security experts within their own organisation [89].

Within the SECaaS landscape there are many vendors offering managed cryptographic services to clients, either as part of a wider SECaaS product or standalone. These services are similarly varied in nature, including but not limited to encryption of data at rest, data in transit, data in use, authentication services, entropy services and key management services. The term “Cryptography as a Service” (CaaS) can be found in a whole host of marketing publications and industry white papers yet may describe slightly different things each time. The perceived value in this offering is that a customer can abstract away the complexity of managing a technology that is notoriously difficult to implement and manage well, while benefiting from the security services that a well designed and managed cryptosystem can deliver [102].

The danger of relying on a cryptosystem that might be flawed or that can be easily bypassed due to improper integration into the wider system that it is supposed to protect has often been pointed out by many influential thinkers on the topic [5]. Offering an outsourced cryptographic service that is generic (so not integrated into a specific system) and that cannot be properly evaluated at the design stage by a customer appears to play into these dangers. Indeed poorly designed cloud based CaaS offerings have existed and likely still do, but their implementation errors are often not brought to light until they are already serving millions of customers [11]. Yet there is a growing desire to adopt cryptographic technologies [36] and these services appear to fulfill a need.

1.2 Focus of Study

The focus of this study is the intersection between the proliferation of CaaS offerings and how they can be properly evaluated in light of current guidance and thoughts on cryptographic standards, service management models and assurance approaches. This

practical and CaaS specific overview and guidance is what the author finds to be lacking in published academic research and industry guidance.

The existing body of research on CaaS focuses mainly on technical implementations and solution architectures. Many papers assume a cloud service model, although some do look at implementations for internet connected devices (known as Internet of Things or IoT devices). The research proposals are often solution focused, attempting to solve a security question or understand how best to balance computing resource in order to deliver a particular cryptography related outcome. These papers are theoretical, and no accepted architectural patterns or standards have been taken from them and adopted into any standards or commercial service offerings. Although these studies do specifically state they are looking at XaaS models for cryptography, encryption or authentication, there is very little consideration of this aspect in the architecture which is put forward for study.

When considering what commercial offerings are available under this moniker the terms Cryptography as a Service and Encryption as a Service are often used interchangeably, in addition to a host of other cryptography adjacent XaaS terms which can be encountered. Often similar services will be offered under different names. There is a lack of commercial and academic consensus on what managed cryptographic services actually encompass.

Although some industry bodies have published guidance on the implementation and management of an Encryption as a Service offering, they do not consider other offerings which manage aspects of cryptography for a customer and whether the guidance from one might apply to the other. Acknowledgment is made of the need to apply proper service management approaches to these offerings, yet no consideration is given to what specific aspects of managing cryptography might require further or deeper consideration. Standards for cryptography and managed services are well established, yet there is little understanding of what a customer for a managed service might need to understand in order to verify compliance with standards.

1.3 Value of Research

The research will guide a security practitioner on what metrics are useful to properly assess a potential CaaS offering and whether it can achieve their security goals in the context of their business. When considering whether to purchase a particular service

offering, a potential customer must be able to judge whether that service offering can deliver value to them, and understand how that value is to be measured. This project will focus on how a commercial CaaS offering, delivered within the paradigms of cloud services and SECaaS, can be assessed, monitored and risk-managed from the viewpoint of a customer considering its procurement.

CaaS is marketed as a solution which absolves the customer of the need for expertise or resources in cryptography while still benefiting from the security properties it can deliver. The value of this research is that it can highlight the gaps in published guidance, standards and white papers while providing context to a potential customer so they can understand why those gaps still must be considered. This can then shape any discovery they may then undertake directly with service providers.

1.4 Research Aims and Objectives

The research aim is to provide an overview of what existing CaaS offerings are available to a potential client, understand what cryptographic services these offerings provide and likely use cases for them. Once this has been defined the aim is to consider relevant themes which have been identified in a literature review for the assessment, assurance and service management of a CaaS offering, to critically evaluate whether there might be gaps or shortcomings in these services when included as part of an organisation's security architecture and to suggest what metrics might be available for their assessment The research will be qualitative and exploratory in nature.

Specific research objectives are:

Definition and Taxonomy. To search for what current market offerings exist that could be considered as part of the CaaS family. Cryptography provides a number of security services as part of a larger security architecture, yet this specification is often lacking in marketed products. Once defined these products will be enumerated in the context of what service they offer, in what deployment model and a popular use case will be also be elaborated. These findings will be analysed in the context of any previous academic research. A taxonomy will be defined for CaaS, allowing a practitioner to understand the breadth of services which are offered in today's commercial landscape.

Standards Analysis. The current standards landscape will be reviewed, not only in cryptography but also in service management as well as other products such as hardware. Following the review the analysis will consider how, for an outsourced service, a

customer can recognise which standards are pertinent to them, how to ensure they are properly validated and what extra work or risk they might need to consider as a result. The results of the search will then be revisited in light of these findings.

Trust and Quality Analysis. In depth analysis of these themes will be conducted for CaaS. A thorough understanding of how much and whether a customer can trust a CaaS provider is needed, as well as a consideration on what extra measures a customer might need to take with their provider depending upon the service they procure and trust model that they decide upon. Quality definition for CaaS is an under researched topic, the approach here will be to take an existing piece of research using the ISO/IEC 25000 software quality model to evaluate CaaS and review this work for feasibility, while also considering what other quality models might also apply.

Evaluation of Current Market Landscape. Current vendors for CaaS will be evaluated for use of relevant standards, trust frameworks and quality models, using any metrics which may have been identified as appropriate during the analysis. This will be a qualitative evaluation based on publicly available internet sources.

1.5 Methodology, Scope and Limitations

One of the aims of the project is to understand the difference between what current academic literature states about CaaS and what is actually being offered as a product under that moniker. The project will compare the results of two searches, the first being a traditional literature review of academic sources for CaaS and related themes. The second will be a search of CaaS offerings on the internet. This will done by collecting online sources of information about these offerings such as searching suppliers websites for product information, supplier blogs, design papers and certification statements.

This project is heavily focused on cloud based CaaS offerings aimed at customers in enterprise/office based environments. There is very little consideration in this project of how CaaS might be deployed into other environments such as Operational Technology (OT) or the Industrial Internet of Things (IIoT). The findings may be applicable to some extent to these areas but they were not a deliberate target of the project. Similarly, the findings of Chapter 3 are that encrypted messaging can be deemed a CaaS offering and it is included in the taxonomy, however this is relatively mature technology compared to other offerings. Providers of encrypted messaging services have been the subject of much in-depth study, and this project would not be able to add anything in this area, so

does not focus on it.

The commercial landscape for cloud services, SECaaS offerings and CaaS offerings is very fluid and the findings for Chapter 3 and 8 apply to the research period of April-August 2024. This is not a mature market with understood norms around product specification, and the services highlighted in this report are likely to change and may cease to exist.

The research is entirely on secondary sources, based on open source findings, existing research, product websites and company white papers. No proprietary product information such as service designs, architectures or source code was obtained. As the aim is to try and characterise the entire CaaS market, this study has had to necessarily sacrifice depth for breadth. Service providers were not approached for clarifying questions, however the recommendations and findings from this project should be able to inform any questions directed to a provider when considering whether a CaaS offering is appropriate.

1.6 Structure of the Report

Chapter 1. Introduction to the project, context, methodology, research aims and value of research.

Chapter 2. Literature review. The literature review will cover works which specifically address CaaS or other cryptography related XaaS studies, as well as any works on outsourcing cryptography and some approaches to solving cloud security issues with cryptography. It will also look at adjacent themes in cryptography concerning key management, usability and trust, and will also review the body of work on SECaaS in so far as it is relevant to cryptography.

Chapter 3. Discovery and Taxonomy. This chapter will present what CaaS offerings are currently available commercially, based on the web search methods described earlier in this chapter.

Chapter 4. Standards. A chapter considering what standards are relevant to CaaS, covering organisational, cryptographic and other technical standards which are deemed relevant, and how these can be used when assessing a CaaS service.

Chapter 5. Trust. A consideration of the different trust models encountered in CaaS offerings, comparing privacy by design to privacy by policy and discussing commercial

trust frameworks.

Chapter 6. Quality. Consideration of how quality can be defined and monitored or measured in CaaS and a review of software quality models.

Chapter 7. Re-visit results from Chapter 3 and provide qualitative analysis of suppliers offerings against the metrics discussed in Chapters 4, 5 and 6. Attempt to characterise different CaaS market sectors and highlight areas where suppliers do not provide sufficient information.

Chapter 8. Summary of general findings, conclusions, suggestions for further research.

Chapter 2

Literature Review

2.1 Definitions

In order to place this work in its proper context we must establish some definitions. Many of the works reviewed for this search are either focused at the cloud environment or that of the Internet of Things (IoT). For cloud computing we will use the NIST definition which is “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources…that can be rapidly provisioned and released with minimal management effort or service provider interaction.” [73] Cloud also provides three service models which very roughly align to a TCP/IP stack (from bottom to top they are Infrastructure as a Service, Platform as a Service, Software as a Service) and it is these service models which have informed the concept of “X as a Service” (XaaS) in which the X denotes whichever computing resource is being applied to a service model.

The IoT has been defined by the ITU as “a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.” [58]. The common shorthand for understanding IoT is that of an ‘internet connected fridge/toaster/camera/thermostat’ - so a device that gains additional features over and above its original purpose, due to the fact that it now has an internet connection.

Somewhere between cloud and IoT are the concepts of ‘fog computing’ and ‘edge computing.’ These are simply attempts to define a halfway house between a resource and power intensive cloud data-centre with fast fibre connections and powerful servers and

a small, compute and connectivity constrained IoT toaster. A fog or edge computing node is a networked device often positioned between these two extremes in an attempt to better balance the trade-offs typically found between large, immobile computers and smaller more mobile computers such as weight, power, connectivity etc.

The term Cryptography as a Service (CaaS) is encountered as often as Encryption as a Service (EaaS) and Security as a Service (SECaaS). In order to understand CaaS and EaaS, we must first look at how SECaaS has been treated.

2.2 Security as a Service

SECaaS is the notion that security service provision can be outsourced to a third party, often as part of a cloud-based service offering. This definition is provided by the Cloud Security Alliance (CSA), with initial implementation guidance released in 2012 [30]. The attraction of SECaaS is that the increased scale and connectivity of cloud technologies can streamline the delivery of security to customers who would lack the necessary expertise to secure their IT [102]. The CSA definitions are split into several domains covering topics such as Web Security, Email Security and Network Security. One of the services so defined is Encryption as a Service.

Elsayed and Zulkernine [34] build on the domains defined by the CSA to provide a more complete taxonomy at Figure 2.1. They recognise that the security mechanisms defined by the CSA must also be modelled alongside other factors such as delivery models, deployment models, threats and attack surfaces. They also classify the controls into three main categories which are: protective, reactive and detective. Their literature survey noted that at the time of writing, EaaS was under-represented in the available research, with only 5% of published papers focussing on this.

A literature review on SECaaS [115] maps the CSA mechanisms against the security controls outlined in ISO/IEC 27002 to better understand the nuances of the controls implementation. They identify the advantages of SECaaS in cost, reduced customer burden (such as using a single interface with easier input options), ability to concentrate expertise in the provider and continued security updates. The drawbacks are identified in that the service itself can become a centralised target for attackers and there is huge reliance on and disparity between the quality of the service offered by the vendor.

The customer requirements when adopting SECaaS are an important factor to consider. Two papers consider the needs of a small business in adopting SECaaS. Pla et

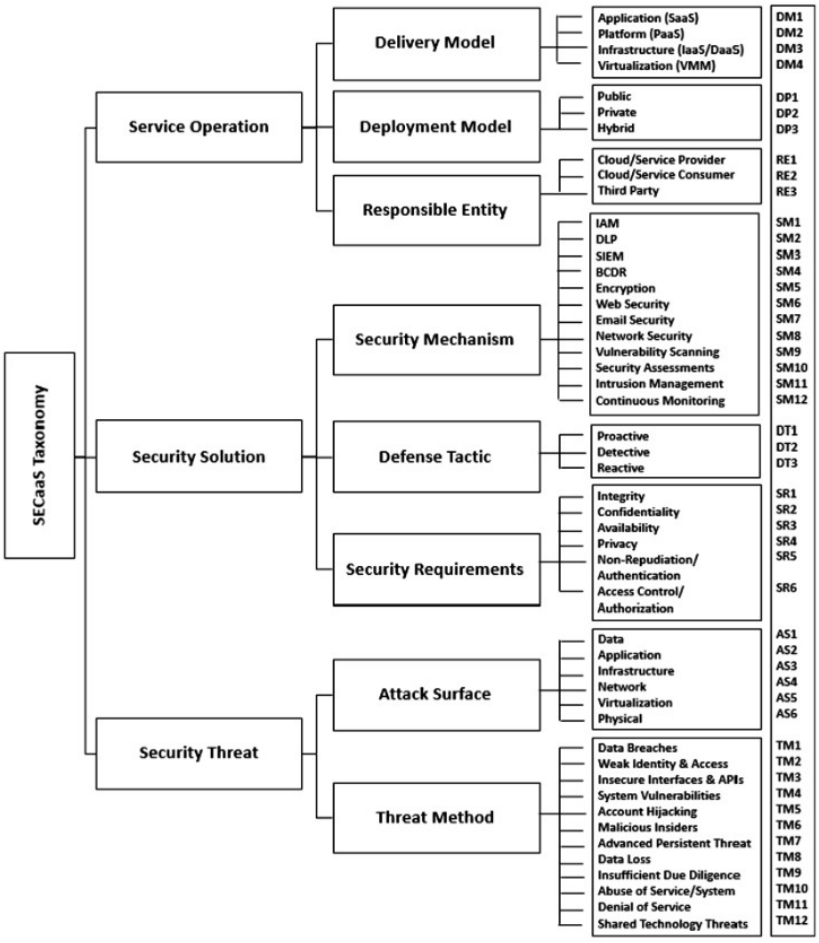

Figure 2.1: A taxonomy for SECaaS [34]

al [89] analyse the requirements of a small business for both on-prem and cloud solutions for SECaaS, while Nazareth et al [80] seek to understand what factors might drive the adoption of SECaaS by Small to Medium Enterprises (SMEs). Pla notes that SECaaS is attractive to a customer that prefers to focus on business outcomes and can take advantage of cloud properties such as the ability to rapidly, however they note that SECaaS offerings are often very generic, with fewer more fine grained offerings. They also highlight the issue that a customer cannot verify the capabilities of an outsourced toolset, that there is a need to trust the provider and that simply contracting for a SECaaS does not abdicate the customer of all responsibility in security management. Not noted in the study but worth also considering is that the customer also has to be a knowledgeable customer, able to properly articulate their requirements. Nazareth finds that

consumers appear to be price driven as opposed to threat driven, which may explain the relatively low take up of SECaaS offerings in the SME space.

SECaaS implementations are varied and consider it from several angles. As it is presumably a paid service, the notion of how to effectively bill for it is considered in [110]. The proposed model allows customers to tailor additional security functionalities above a standard baseline offering. They develop a security architecture based on Attack Detection Components which sit in the infrastructure layer of a cloud offering. These act as policy-based intrusion detection systems, with a service provider component delivering the baseline offering and a tenant specific component allowing for greater configurability - which then enables more fine grained billing from the provider.

A more general model [25] considers SECaaS in the Internet of Things (IoT). They develop a system whereby constrained security resources are allocated according to a contract-based game theory model. This paper brings in the cost component of SECaaS by using the model to determine where cost increases for a security service might be warranted due to the changing Quality of Service (QoS) offered by the provider. QoS is only modelled as an abstract concept so real world implementations will certainly differ to the findings of this paper. It also posits QoS as an always connected cloud service and does not consider what additional non connected mechanisms could be implemented to supplement the 'always on' nature of a service.

Boudi et al [22] explore the provision of edge/fog computing technologies to provide SECaaS to a mixed IoT estate. They propose a container based solution (built on Docker) for carrying out security functions, and acknowledges the importance of good orchestration across such a resource constrained environment. This paper does not propose exactly which security enforcing functions are implemented in this architecture. Kang et al [62] consider portability of SECaaS, describing a data at rest encryption service that can be easily integrated into existing cloud infrastructures (and presumably easily migrated to a new provider should the need arise), while Sharma [99] proposes a SaaS like service that can easily be bundled in with other SECaaS offerings into a unified console.

2.3 Reducing Complexity in Cryptography

One argument for the value of EaaS is that a service offering can reduce the complexity and difficulty for the customer of implementing and running encryption solutions and key

management solutions [26]. There is a wide body of research on security implementation failures and usability. Cryptographic implementation failures are still one of the most commonly found [85] and attacks on the implementation or key management of a cryptosystem often enable much larger attacks [77] [107] [41]. A well known argument is that without taking a whole system view of cryptography and its interdependencies with other mechanisms attacks will continue [98] [5]. Another one considers how a cryptographic system that is difficult to use will either not be used, or user error will introduce new vulnerabilities and attack surfaces [118].

Koien [64] revisits these arguments and finds them to still hold true, however also maintains that the growing trend for outsourcing security can never fully address an organisation's responsibility for security. He also calls out the need for trusted hardware and the importance of human factors when considering the security of a cryptosystem. On a wider theme but still applicable to our research question, Anderson [6] calls out the difficulty of implementing end to end encryption onto existing systems, and remarks upon the lack of motivation for IoT device manufacturers to adhere to deliver basic software patches and the poor implementation of authentication for these devices [6].

Cloud offerings such as Mego.io and SpiderOak promising encrypted data at rest with privacy of the data have been found not to be able to fulfil that promise after thorough analysis [3] [11]. Although the flaws were fixed by the providers after responsible disclosure, some of the flaws were due to poor design and implementation choices, such as using non-standard authentication protocols, following poor key management procedures or basic insecure cipher implementation (such as not using authenticated encryption to encrypt a client's public key prior to storing on a cloud server [11]). Considering that one of the purported attractions of CaaS/EaaS under the SECaaS model is the ability to outsource expertise, the actual cryptographic expertise employed in the initial design of these outsourced offerings appears to be insufficient.

2.4 Cryptography and Encryption as a Service

The concept of CaaS arose at a similar time as the notion of SECaaS, but it was not framed as a managed service, but more as a method of orchestrating distributed outsourced compute power to help achieve cryptographic outcomes.

Most of the literature agrees that the first public definition of CaaS was proposed at the 2014 RSA Europe Conference, with the core value proposition being to "perform

keyed cryptographic operations on behalf of end points via web services without exposing important cryptographic keys to the end points." [95]. Another use case was for the delivery of entropy of mobile devices for generation of better encryption keys. This use case was actually further developed in [111] proposing a cloud based deterministic random bit generator with seeds provided by a high-entropy source. The resulting random values can then be provided to client systems or devices over HTTPS.

Returning to the original 2014 proposition, CaaS was purported to be scalable, with a lower running cost and wider API support than more traditional customer owner hardware based approaches [95]. Challenges identified were those of endpoint authorisation, trust in the cloud server, network connectivity, and security of the provider [95].

One paper [92] examines the applicability of CaaS for mobile applications, and their analysis is very much through a web application lens. They view CaaS similarly to that of Software as a Service (SaaS), a marked difference to the 2003 paper which focussed mainly on infrastructure. This may indicate that CaaS can apply equally at all three of the recognised cloud service layers (Software as a Service, Platform as a Service and Infrastructure as a Service).

2.4.1 Implementations and Architectures

The CSA SECaaS Implementation Guidance for EaaS [30] gives guidance on best practice for cloud provided Encryption and Key Management Services. Notable observations from this document are:

- Architectures must consider encryption for data at rest, data in transit and data in use.
- Key management should not be carried out by the same provider hosting the data. They propose using a remote key management or client side key management service.
- Interoperability of encryption and key management technologies should be considered.
- Any cloud EaaS implementations must be supported with robust security policies that align to those of the customer. [30]

Many studies for Caas or EaaS are solution or architecture proposals. The studies fall into three broad patterns namely: Using the cloud to outsource cryptographic opera-

tions, storing encrypted data on the cloud and providing encryption services for data in transit.

2.4.2 Outsourcing Encryption

An early white paper from Palo Alto Labs [18] proposed a solution for offloading computationally expensive public key operations for network security onto a centralized server. They identified the main types of applications for their solution to be simple encryption and decryption as well as bulk encryption/decryption.

Rahmin et al [93] propose a hybrid private/public cloud to address the trust issues between a customer and provider while meeting the high compute costs of encryption through pooled resources. Pei [87] leverages the extra resources available in the cloud and cluster processing for an EaaS offering aimed at encrypting large video files.

While not explicitly presented in the XaaS model, the increased compute power provided by cloud offerings has also been studied in the context of implementing computationally intensive public key encryption schemes. ABE was developed by Sahai and Waters [96] as an extension of Identity Based Encryption (IBE) which aimed to address the issue of needing to issue and trust certificates in order to share a public key, with certificates binding the identity of a party to that public key value [60]. An Attribute based Encryption Scheme (ABE) is attractive for its ability to produce ciphertexts that can be decrypted by an individual that possesses the correct attributes needed to meet a *policy* that was defined at the time of encryption [60].

A drawback of ABE is that the ciphertext and decryption compute overheads grow as the access policy gets more complicated. This makes ABE implementations impractical for smaller mobile devices, which generally have less compute power and battery life. Green at al [44] built on the concept of proxy re-encryption by proposing an ABE scheme that could outsource the bulk of the resource intensive computation of decryption to a cloud based proxy. They use a technique called key blinding in which a single key derivation function produces both a private key K which is held by the user and a transformation key TK held by the cloud proxy. The proxy uses TK to transform the ABE ciphertext CT into an intermediate ciphertext CT' which can then be decrypted by the user's private key for far less computational effort. This is shown in Figure 2.2.

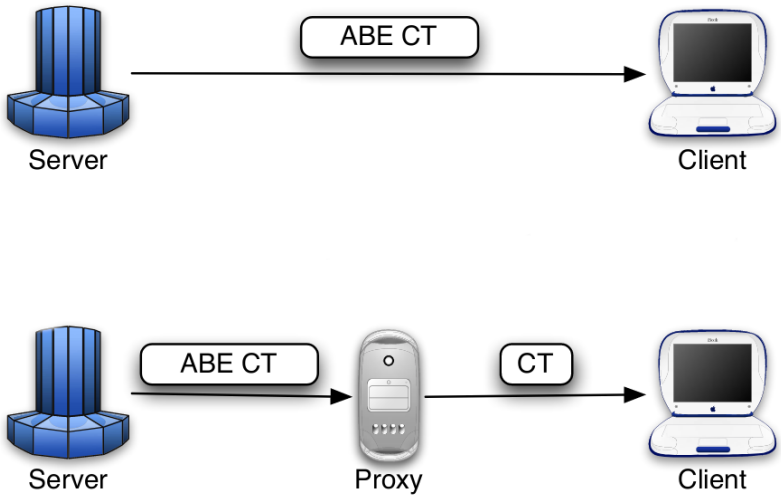

Figure 2.2: How ABE ciphertexts are traditionally retrieved vs how they could be using outsourcing [44]

2.4.3 Verifiable Outsourced Encryption

A challenge in the outsourcing of encryption or decryption is whether a client can be sure that an outsourced provider has actually carried out the operation as intended.. Lai [61] observed that that, “In the cloud computing setting, cloud service providers may have strong financial incentives to return incorrect answers, if such answers require less work and are unlikely to be detected by users.” [61] Numerous additional schemes for verifiable outsourced decryption have been proposed since [68] and [124]. Although these papers do consider the cloud as a place to outsource superior compute power, only [124] actually considers how to implement an architecture on the internet, across more than one cloud offering. Their use case is for mobile applications to perform ABE.

Ma et al [70] attempted to develop the concept of outsourced ABE further by proposing a complete encrypt/decrypt system and by considering how the proxy server can prove to the user that both the encryption and decryption operation was carried out correctly (known as verifiability) and how a server could be defended from a false accusation of an incorrect decryption (known as exculpability). Their batch verification method for encryption was proven incorrect [123] in 2017. To date there is no study that has proven verifiable outsourced encryption.

2.4.4 Homomorphic Encryption

Homomorphic encryption has attracted great interest from cloud providers due to its ability to maintain confidentiality of customer data while still enabling processing of that data by a provider. The current problem is that a Fully Homomorphic Encryption (FHE) scheme (one that can perform the full range of computations on data as opposed to a partial scheme which cannot) is still prohibitively demanding both in terms of computation and key size. Zkik et al [127] have proposed a fully homomorphic encryption scheme that is delivered as a service to paying clients - they explicitly base it on the SECaaS model. Their cloud based architecture uses a private core cloud to manage key management, authentication and encryption and decryption. Any homomorphic calculations are then outsourced to a public cloud service provider. However in order to offer this as a service, the provider managing the private core cloud still needs to be trusted by the client not to view encrypted data, so the guarantee of user privacy is doubtful. [50] proposes a very similar architecture, the main difference being that the public cloud is simply used for the storage of encrypted images. They aim their solution at the mobile market, with a similar aim of utilising superior cloud computing resources to perform the encryption.

2.4.5 Threshold Cryptography

Threshold cryptography relies on the collaboration of more than one party to jointly compute a secret, in such a way that no single individual con reconstruct that secret on their own, and the secret is computed when a threshold of $(t + 1)$ from n parties carry out the operation [33]. The use of cloud services to achieve threshold cryptographic operations has been considered in [2] where an implementation of threshold cryptography is offered as a distributed service, provided on demand through an API. Their envisaged application is to run on a blockchain. [16] provide a similar architecture for a similar use case, however their solution provides an ability to scale to much larger numbers of participants in a threshold scheme.

2.4.6 Non-Scheme Specific

Other factors that make EaaS attractive to the user are the ability to offload costs to the provider (who can presumably scale up to reduce the per implementation costs) as well as offloading key management [102]. Some studies propose EaasS or Caas architectures without defining any specific cryptographic scheme. An early architecture

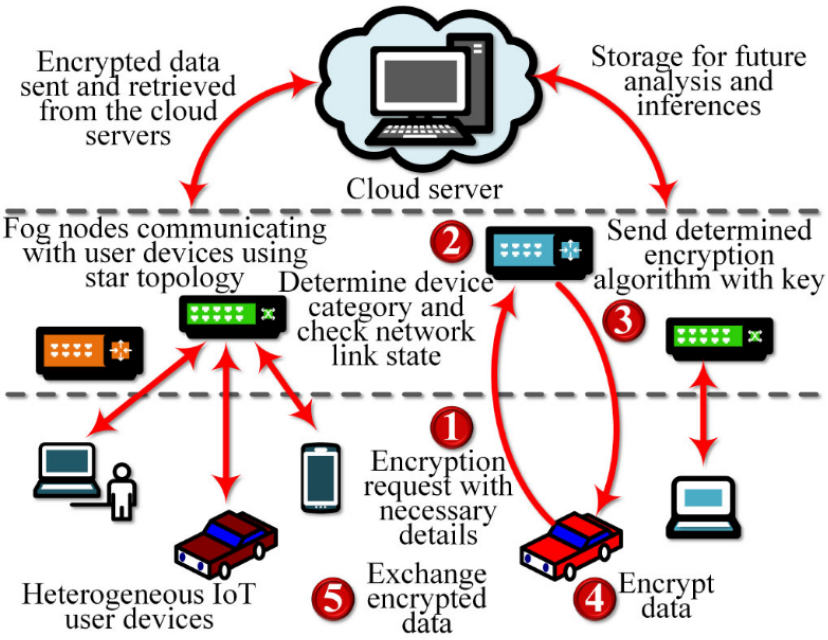

Figure 2.3: A Network Architecture and Data Flow for EaaS in IoT [32]

([21] addresses the issue of not fully being being able to trust the cloud provider with sensitive credentials and cryptographic primitives and introduces the concept of a client controlled domain where cryptographic operations take place. This solution still resides fully on a cloud provider's infrastructure, and while it provides an architecture that a client can trust, it is not clear that any cryptographic operations are being carried out on the client's behalf. Xue et al [125] propose a very generic architecture that reflects many other cloud compute services in that hardware resources are pooled to create a virtualised cryptographic operations stack, with load balancing and resources allocation managed by the cloud provider, and a generic service layer which interacts with client requests. Colombo [27] describes a Data Protection as a Service offering which uses the same encrypt through API model as [95], with keys management controlled by the customer.

Deb at al [32] devise an architecture (Figure 2.3) which combines fog computing nodes combined with a EaaS offering that tackles the issue of heterogenous devices and variable network links. Their contribution is to develop a platform that "dynamically decides the appropriate encryption scheme concerning the device category and network link state" [32]. This approach is also covered by Zhang [126] however they follow a

slightly different architecture and use different criteria to select an appropriate encryption scheme, focussing instead on security and latency requirements of a proxy node.

2.5 Key Management

The management of cryptographic keys on cryptosystems is in its own right an area of substantial study. A recent survey [94] has provided a taxonomy of Key Management Systems (KMS) (Figure 2.4which this review will handrail where appropriate. The concepts which are most discussed from this taxonomy when looking at KMS in the CaaS or SECaaS paradigm tend to be in key sharing techniques and location of cryptography operations.

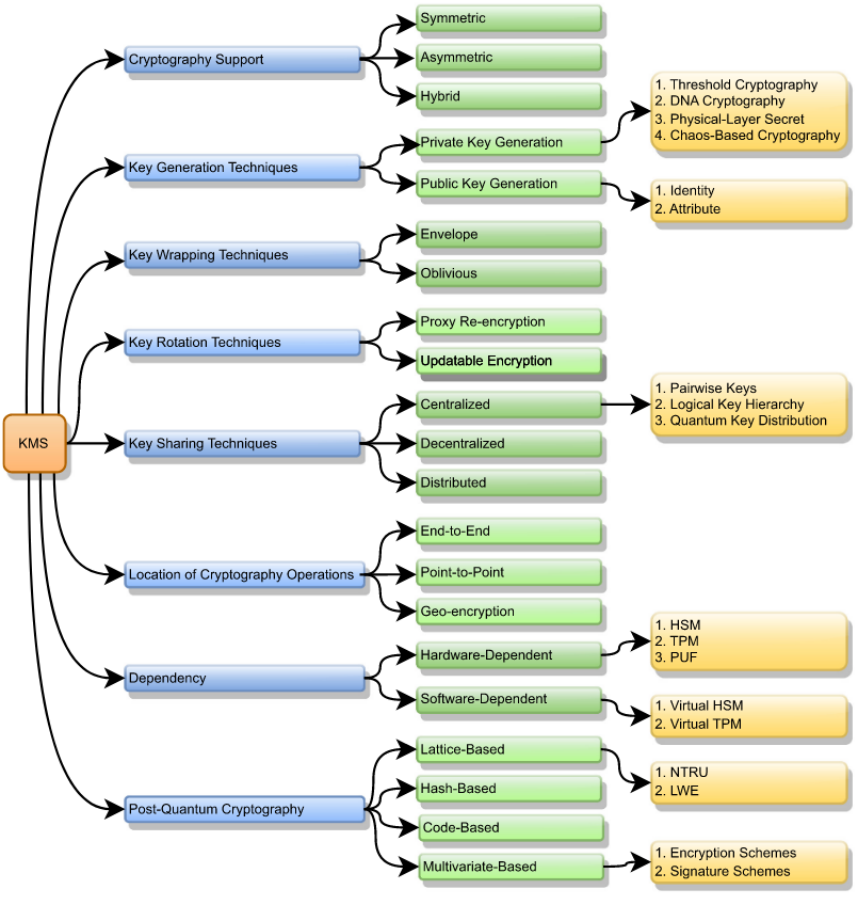

Figure 2.4: KMS Taxonomy [94]

KMS in cloud deployments are a well known difficult problem, with specific challenges

found in all three main cloud service types (Infrastructure as a Service, Platform as a Service, Software as a Service) [24]. Both IaaS and PaaS key management is a challenge due to the difficulty in bootstrapping authentication, while SaaS key management suffers from complexity as well as the need to fully trust the provider [24].

Key Management as a Service (KMaaS) is a commercial offering where all aspects of the key management lifecycle are outsourced to a provider [65]. KMaaS offerings are extremely widespread, either bundled in with a cloud service provider's other offerings (AWS, Cloudflare) or available as a separate service which can work across more than one cloud offering. Factors such as data residency requirements and levels of client control over the key lifecycle will likely influence a client's selection of a particular offering [69]. Trust in the provider, or the system, is fundamental to many of these schemes, which is often underpinned through the use of dedicated Hardware Security Modules (HSMs) for the storage of keys. If a client maintains their own keys in their own managed security architecture, then added key management complexity is placed on the client, and other performance issues must be considered such as latency in retrieving keys from the customer infrastructure to that of the provider [49].

If key management is needed to support multiple cloud environments or multiple solutions there is a risk that each of these already come with their own dedicated KMS. The management of multiple differing KMS will be complex, whether done by a client or provider. The Key Management Interoperability Protocol (KMIP) defines a single information model for all actors in a cryptosystem which require a key, as well as the keys themselves, with common objects, operations and attributes [84].

Key sharing techniques draw on earlier works which considered how to build scalable KMS for group communications. A survey of these KMS done in 2003 [91] identified three main approaches, which were centralised management and control of group keys, decentralised management across several actors and distributed protocols for key management across all group participants [91]. Their findings are that while all models have trade-offs in terms of complexity and ease of use, fully distributed protocols cannot scale.

In terms of studies on KMaaS in cloud offerings, [100] considers how a centralised group key management approach could be made more reliable and fault tolerant. They note that a single manager is a significant target and single point of failure, their model distributes the compute power required for a single controller across several nodes yet retains the centralised authority of a group key controller. Their solution, built using the Apache ZooKeeper service, achieves this, however they note that in order to build a

service that can scale care must be taken to avoid designed-in performance bottlenecks.

Lerman et al [67] propose an approach towards the management of long term keys using threshold cryptography. Their solution relies on the cooperation between different cloud providers to generate encryption keys, with the threshold scheme preventing a single provider from being able to generate the entire key and then access a user's data. Their claim is that this removes the requirement for the user to store the key or needing to fully trust a single cloud provider. It is accepted that many barriers remain before this could be implementable.

Fehis et al [39] model how a KMS could be deployed to support more than one paying client, and ensures no conflict of interest for a provider by defining Chinese Wall based security policies to separate access to keys.

The Advanced Metering Infrastructure (AMI) is provided by utilities companies to enable the networking of smart meters placed in customer homes [117]. It is a useful lens through which to analyse the opportunities and challenges of implementing CaaS and several papers have used it as a case study for both Eaas and KMaaS. Early guidance on key management for AMI [66] stresses the need to design a system that can be operated by a user with no security background and that due to the sheer number of keys in scope, automation of key management must be pursued as "manually managing cryptographic keys is not an option." [66]

2.6 Trust Management

The ability of a customer to trust a cloud service offering has received much study. A survey of trust management approaches used in cloud environments [83] identifies four main categories which are termed: "Policy" "Recommendation" "Reputation" and "Prediction." The approaches themselves are a mixture of technical, procedural and reputation or relationship-based. A more detailed proposal [47] identifies a set of attributes with which to score and identify a 'trustworthy cloud'. Their architecture is that of a cloud based broker, where service providers can register, and users can assess services which have been modelled based on a range of inputs such as questionnaire responses, expert assessments, certifications and so on. The resulting trust assessment is modelled as a probability in recognition of the fact that some of the sources may be unreliable.

The Cloud Security Alliance [31] has published guidance on expected roles and respon-

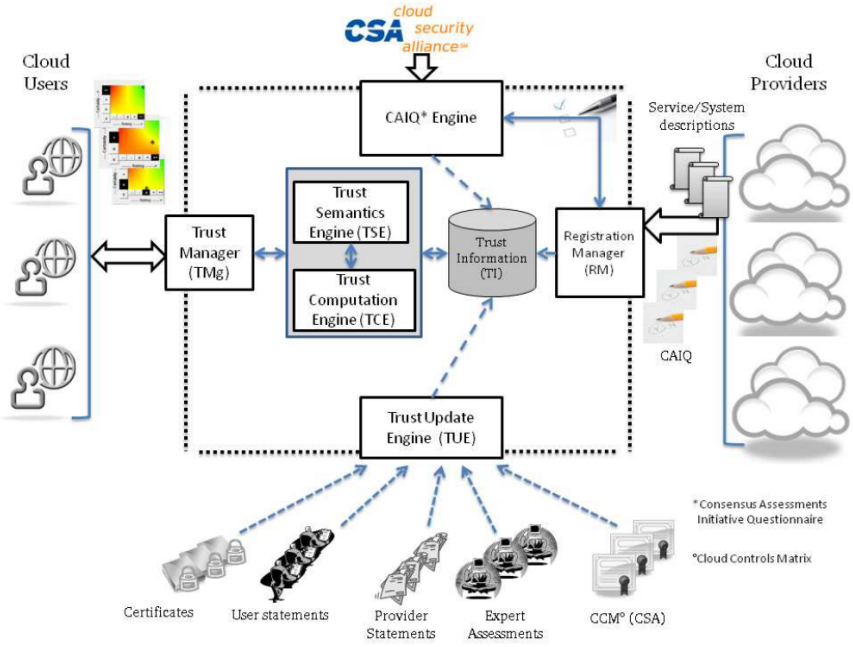

Figure 2.5: Trust Management System Architecture [47]

sibilities for providers of third party security services. Some common desired characteristics include providing well-defined Service Level Agreements (SLA) and employing qualified and skilled professionals. There is very little specific guidance on using encryption, just that encrypted data must be accessible by the customer [31], with no guidance on how to measure that.

One study combines a number of cryptographic security services with other security controls into a wider service offering they term “Trust Management as a Service” [45]. This is probably the closest offering that maps into the PaaS service model, its intended use case is for deployment into an untrusted environment like a public cloud. The customer is still expected to hold their own private key separately in order to use it.

2.7 Quality and Performance

Almost all of the studies and proposed architectures surveyed consider the performance of their solutions, against a variety of metrics. Typical performance characteristics that are often tested in distributed systems are latency [18], while many cloud based architectures look at use of computing resources, either through time to perform certain

operations or CPU or memory usage during those operations [44] [62] [87] [93]. One study [32] which is IoT specific, also looks at energy consumption, showing how an EaaS can reduce energy consumption requirements for resource constrained devices.

Two papers consider the applicability of EaaS through characteristics that are not purely performance based. [99] uses a range of criteria to perform a qualitative evaluation of their proposed architecture. The analysis doesn't quite work, some criteria which could conceivably be quantified (such as reliability and cost of ownership) are only briefly discussed. Other criteria evaluations, such as security, would be better served with a threat based or risk based analysis. However the paper is quite short and such depth of analysis would have been unbalanced.

Wu [122] looks to develop a quality model which allows for numerical comparison between different EaaS offerings, based mainly on technical performance. They contend that many of the known cloud XaaS quality models do not consider security, which is a failing when evaluating the quality of EaaS. Their quality model devises numerical values for selected characteristics (such as security and reliability) taken from the ISO 25010 product quality standard then applies a weighting coefficient to give an overall score for an EaaS solution. Their evaluation of the model is theoretical and mainly proves that the weighting coefficient provides sufficient distinction between scores. While it provides a useful way to start thinking about whether an EaaS offering is fit for a given purpose, the actual scores are derived from analysis tools that a customer couldn't expect to have access to, so is not practical from a real world perspective.

2.8 Case Studies

Several works devise architectures or frameworks and then apply them to real world case studies in order to compare their performance against a more traditional solution. One early study [48] develops an EaaS offering for data in transit between smart meters in the AMI. Their offering follows a traditional cloud model, with an EaaS server in the cloud performing encryption and decryption, as well as other functions such as key management and identity and access management. They compare this with running a more traditional in-house Public Key Infrastructure (PKI) and claim a substantial reduction in compute overheads. Their argument that EaaS can reduce management overheads doesn't quite hold water, as the same argument could apply for outsourcing the PKI management to an external provider.

Zhang [126] applies their fog based architecture to smart electrical substation communications, with nodes placed acting as proxies for substations. Their three layer architecture has a cloud layer to perform management, provisioning, orchestration and monitoring while the fog layer nodes hold keys and cryptographic configurations for end-points. End-point nodes act as proxies for the actual resource constrained devices and perform encryption and authentication services for them. This case study seems more appropriate, as the architecture genuinely provides a security service that wasn't achievable with traditional Operational Technology (OT).

Butun et al [23] consider authentication as a service for public safety networks, building on a previous schemes devised for wireless sensor networks. They propose a solution which can be deployed onto IoT enabled body sensors which are worn by first responders or firefighters. Their architecture is almost entirely cloud based, with a two-level authentication scheme. They make use of the cloud as well as more locally placed management nodes, with the cloud solution managing the relatively more computationally intensive PKI that issues certificates to users who then use these to register end devices. This scheme has quite a simple architecture, however there appears to be a relatively high user burden in the initial registration phases compared to other solutions that have been reviewed.

2.9 Conclusion

The high level findings of this review are as follows:

- The main identified service offerings in the literature are Cryptography as a Service, Encryption as a Service and Key Management as a Service. The overarching concept is that of Security as a Service. Other "X" as a Service offerings which incorporate cryptography do exist but are not as well studied.
- The main use cases are for outsourcing intensive cryptographic operations to another compute source in the cloud, encryption of data stored in the cloud and as a proxy for encrypted communications between IoT devices.
- The issue of trust in the provider is paramount, with increased complexity for a user the less trust there is. Trust in a provider is bound up with assessing the quality of a provider's offering.
- Defining whether a particular CaaS offering is fit for purpose is very complex, with

a variety of technical, non-technical and financial drivers to consider. Simple performance metrics are easy to capture in a test rig, yet difficult to balance against wider concerns such as billing, ease of use and interoperability.

Chapter 3

Overview of Cryptography as a Service

The aim of this chapter is to give an overview of CaaS and its concepts by examining the most common use cases in which it is encountered. Use cases will be considered in the context of relevant academic research, any products which are currently offered, likely drivers and business models as well as architectures. At the end of this chapter a taxonomy for CaaS will be proposed within which the chapter's findings can be understood. This should not be understood as a complete survey of every single use case for CaaS, rather an attempt to understand what differentiates offerings from each other when they are outsourced. Two well known use cases of encrypting data in transit, that of End to End (E2E) secure messaging and Transport Layer Security, didn't present with XaaS offerings so were not included in this analysis.

3.1 Encrypted Cloud Storage

The most common security service provided by CaaS is that of confidentiality through the use of encryption. Encryption as a Service (EaaS) could be defined as the encryption of an entity's data by another actor possessing either the skills, technology and resources to do so. This is done as part of a managed service. There is an increase in the adoption of encryption technologies in recent years [36]. When it comes to providing confidentiality of data, the three most common use cases are for encryption of data at rest, data in transit and data processing [30]. The requirement to encrypt stored data

has largely arisen from the increased use of third party Cloud Service Providers (CSPs) [30], regulatory and legislative [37] drivers as well as high profile stories of data loss from unencrypted cloud storage [113].

Product offerings in this space are the most commonly encountered CaaS offerings. All major CSPs offer encrypted storage with both client side and server side encryption offerings [8] [75] [42] with encrypt-by-default being a standard pattern. The level of trust between a client and their provider is managed through selecting options for key management, with both the provider's KMS as well as client supplied KMS made available. Recent years have given rise to Zero Knowledge Encryption offerings from some cloud providers [72]. These providers claim true E2E encryption, with the provider having no way to access a client's data despite encryption taking place on the server side.

A survey of both the academic literature as well as current service offerings has shown that a variety of data flow architectures can be pursued for EaaS for data at rest, with varying degrees of trust in the provider and complexity for the user depending on where encryption takes place. These architectures are shown in Figure 3.1, and attempt to show the organisational boundaries across which some level of trust will have to be assumed. The full trust model has all encryption taking place within the CSP, while the true hybrid allows for some encryption on the client side. The third model allows for a trusted third party to carry out encryption on behalf of the client prior to storing with the CSP. The multi-cloud model has a single client managing encryption on more than one CSP. Finally the third party orchestrated model has encryption taking place on client-owned infrastructure, but with orchestration of this encryption carried out by a third party prior to storage with a CSP.

These data flow and trust models are necessarily oversimplified at this stage. They only consider client, CSP and 3rd party controlled organisations as location/hardware based. As we will see in later sections, client controlled virtualisation does not simply fit into any of these models. Additionally, control of KMS is not considered in these models, this will be analysed in later sections.

3.2 Secrets Management

Secrets management is a popular concept in software development and DevSecOps adoption. There is a growing recognition for the need to control the management of keys, credentials, permission and certificates that may be used by a team of developers,

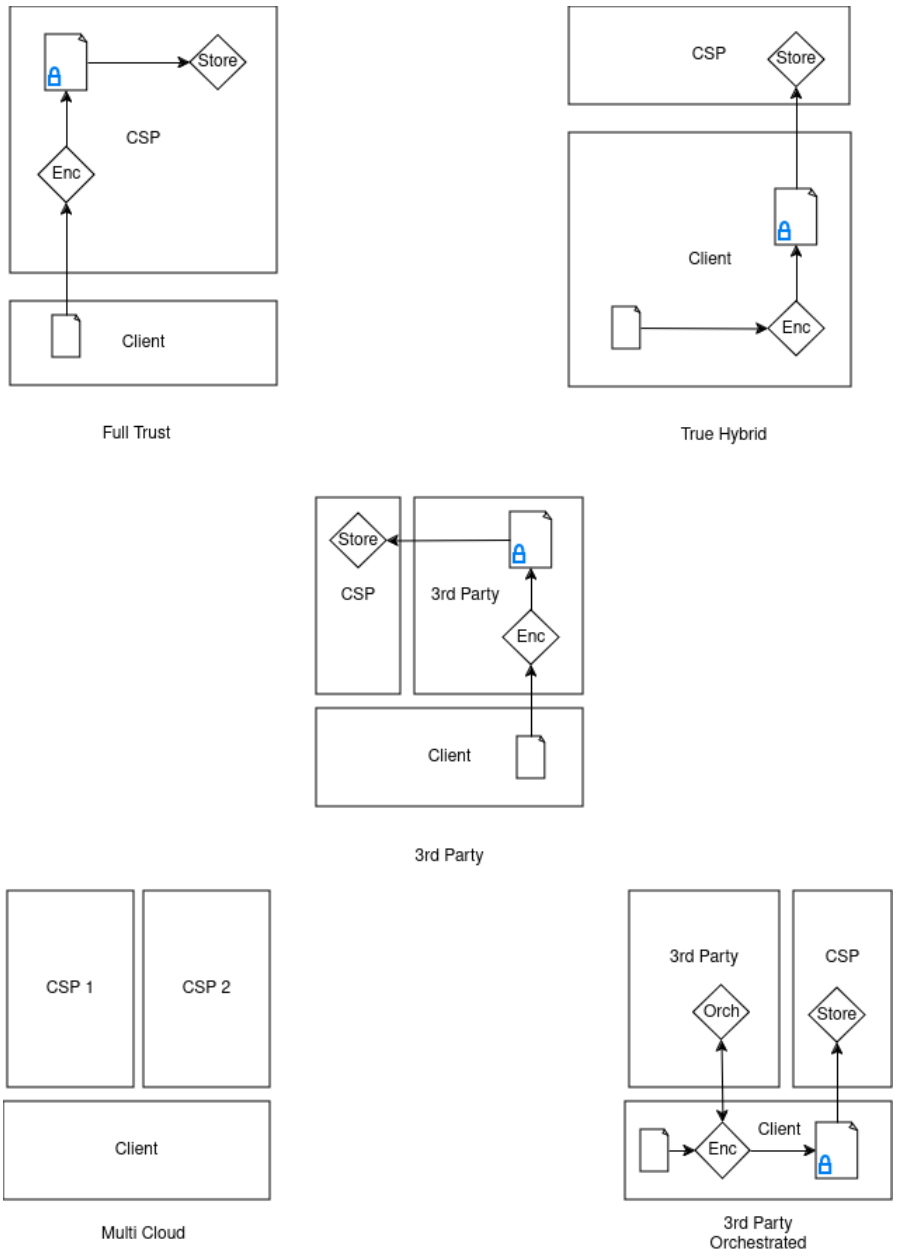

Figure 3.1: Overview of encrypted storage architectures

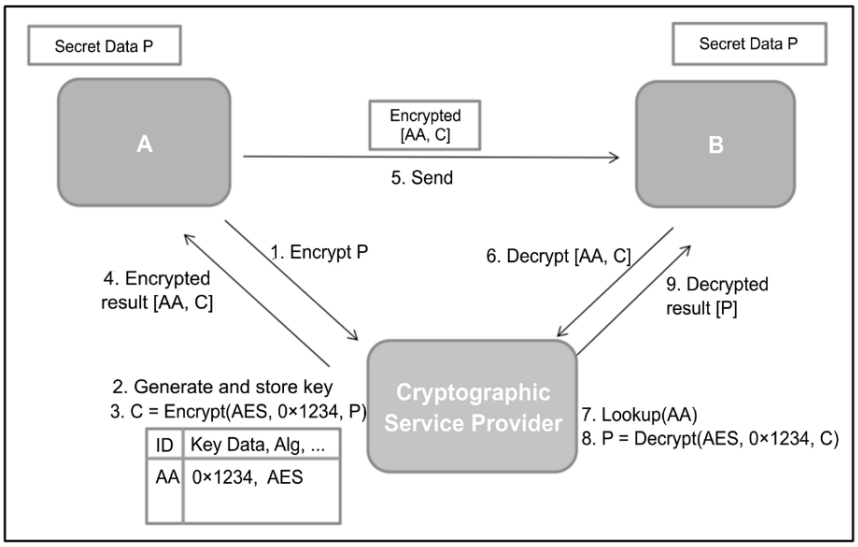

Figure 3.2: Simplified architecture for CaaS showing A passing data to a CSP for encryption then passing the data onto B who uses the CSP for decryption [92]

more than one application, or during the DevOps pipeline. Using secrets engines should reduce the need to hard-code credentials into a code base and is recognised as good practice for application development [86]. Some providers offer Encryption as a Service for applications with applications making API calls on a provided encryption function.

This offering is attractive primarily for its ability to reduce complexity for the client. Deployment offerings are both cloud based as well as client-hosted, with service offerings covering both Software as a Service and Platform as a Service depending upon the tool configuration bought by the customer. Marketing materials for these providers term this as EaaS for data in transit, as encrypted data can be stored on client infrastructure. The business model is a monthly subscription which scales according to client requirements for number of secrets managed and quality metrics such as availability.

3.3 Digital Rights Management

A common use case for the encryption of data in transit is that of video streaming, with the most common implementation of this encryption achieved through Digital Rights Management (DRM). DRM as a term has covered a variety of technologies since its inception however in this scenario it is the encryption of content in order to ensure that a data owner can control the expected use of a digital asset in line with a set of policies

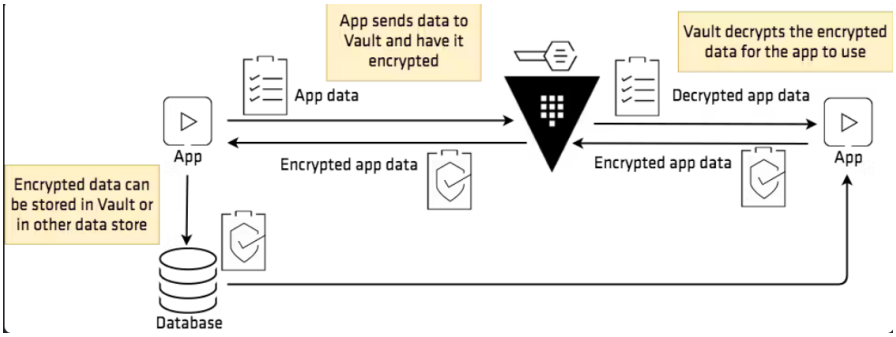

Figure 3.3: HashiCorps Vault EaaS architecture, a database is shown instead of apps explicitly passing data to each other however the concept of a provider carrying out encrypt and decrypt mirrors the 2014 architecture

once that asset has been distributed. In this light DRM provides a form of confidentiality and access control, which is enabled through cryptography.

DRM was originally devised to enforce copyright protection on digital products such as DVDs, eBooks and computer games. For video streaming DRM is used to ensure that only the correct (paying) users can view a streamed video, at the correct time (for example during a rental period or while their subscription is active) while preventing that user from recording or tampering with the stream in any way.

The basic architecture for a DRM enabled streaming service is shown in Figure 3.4. A video stream is encoded then encrypted by the server prior to streaming to a user over a Content Delivery Network (CDN). The user will have a DRM client, either in their web browser or on a dedicated application, which provides authentication details to the streamer and requests a DRM license to view the content. The stream sends the DRM license to the client, contained within it is the key required to decrypt the stream.

In reality a modern streaming service will have to ensure interoperability with a variety of DRM systems in order to ensure the widest possible reach [120]. DRM has a number of well known solutions such as Widevine [63], PlayReady [77] and FairPlay [7]. However the need to ensure maximum coverage for video streaming has resulted in DRM streaming encryption becoming highly standardised, with the Common Encryption Format (CENC [57] and the Content Protection Information Exchange (CPIX) [28] specification ensuring widespread interoperability between DRM solutions.

Service offerings for video encryption and streaming will not only encrypt a client's uploaded video they will also include DRM license management [112], with a customer

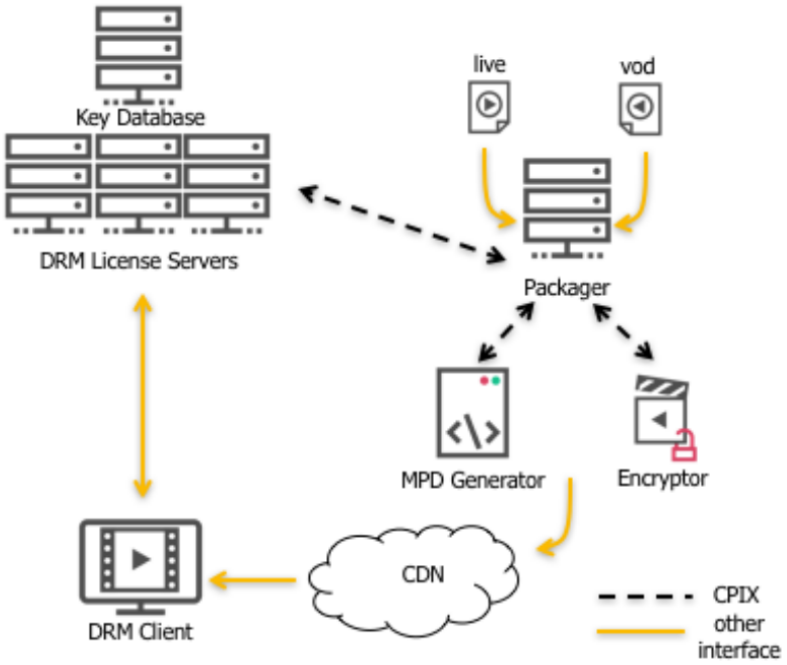

Figure 3.4: DRM Architecture [88]

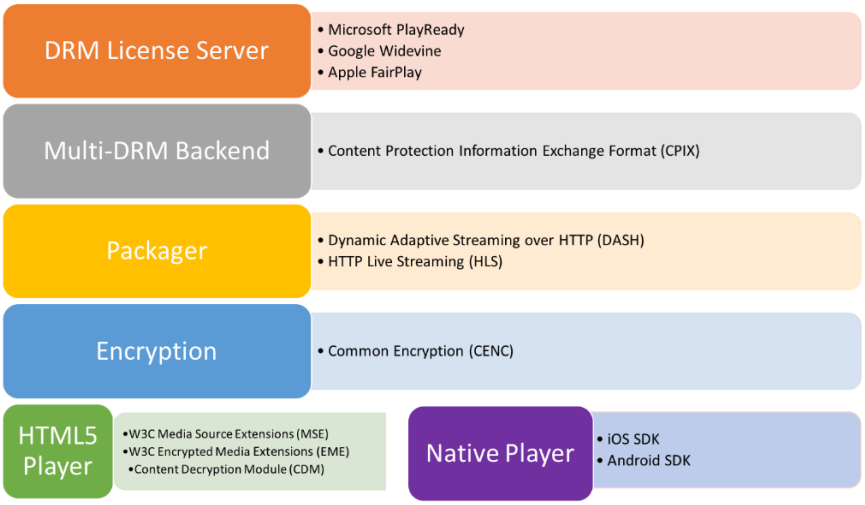

Figure 3.5: A representative DRM stack with CENC, CPIX and supporting a variety of DRM license solutions [88]

paying for full license lifecycle management on the cloud, accessed through an API [35]. While major cloud providers such as AWS and Google do offer this service, the architecture enables the use of a single management service which can work with a range of content delivery networks and interoperate with their specific license servers. The business model is subscription based, with monthly prices based on the number of streaming licenses required, as well as interoperability with major DRM systems.

3.4 Key Management

Key Management as a Service (KMaaS) is a commercial offering where all aspects of the key management lifecycle are outsourced to a provider [65]. KMaaS offerings are extremely widespread, either bundled in with a cloud service provider's other offerings [9] or available as a separate service which can work across more than one cloud offering. Factors such as data residency requirements and levels of client control over the key lifecycle will likely influence a client's selection of a particular offering [69]. Trust in the provider, or the system, is fundamental to many of these schemes, which is often underpinned through the use of dedicated Hardware Security Modules (HSMs) for the storage of keys. If a client maintains their own keys in their own managed security architecture, then added key management complexity is placed on the client, and other performance issues must be considered such as latency in retrieving keys from the customer in

If key management is needed to support multiple cloud environments or multiple solutions there is a risk that each of these already come with their own dedicated KMS. The management of multiple differing KMS will be complex, whether done by a client or provider. The Key Management Interoperability Protocol (KMIP) defines a single information model for all actors in a cryptosystem which require a key, as well as the keys themselves, with common objects, operations and attributes [84].

KMS in cloud deployments are a well known difficult problem, with specific challenges found in all three main cloud service types (IaaS, PaaS, SaaS) [24]. Both IaaS and PaaS key management is a challenge due to the difficulty in bootstrapping authentication, while SaaS key management suffers from complexity as well as the trust boundary between provider and client [24].

KMaaS is a cryptographically enabling service, which, as well as standing alone as a service offering in its own right, can also be combined with all the use cases described

in this chapter. Commercial offerings include both symmetric key and certificate management, therefore a managed Public Key Infrastructure (PKI) service would also fall under this use case.

3.5 Trusted Compute

A Trusted Execution Environment (TEE) provides confidentiality for data in use, it is also referred to as confidential computing. The underpinning technology for TEE is that of a Trusted Platform Module (TPM), providing a physically isolated root of trust for a computer. A root of trust will compute a hash over its current state or code which is then passed onto the next step of a boot process to be verified, with this verification taking place at each stage of the boot process. The end state of these operations is a fully trusted compute environment [104].

The challenge arises when attempting confidential computing in a cloud environment, when a client does not control the physical infrastructure. Although a TPM could conceivably be used by the CSP, the client will require assurance that this has actually been done and that the integrity of the compute environment is maintained throughout processing. This is done through cryptographic attestation, which is essentially a scaled up and distributed version of the root of trust. A draft architecture for a TEE using remote attestation has been defined by the Internet Engineering Task Force and follows two patterns, which are the passport model and the background check model. The conceptual data flow is shown at Figure 3.6 with the background check model shown at Figure 3.7.

Products for remote attestation are bundled in with major CSPs such as the AWS Nitro Enclave [10] which provides the service at no extra cost to an existing customer and full integration with its bundled Key Management Service (KMS) offering. Microsoft Azure also provides an attestation service at no extra cost [76]. There are some third party providers offering a confidential compute service built on top of these existing solutions, offering ease of configurability for policies and integration with a client's existing software development workflow [38].

Outsourced yet trusted computing is a cryptographically enabled service, which is often found in a cloud deployment. Its service model is that of Platform as a Service (PaaS). Client trust in a trusted compute offering is dependent upon the key management implementation as well as robust policy configuration.

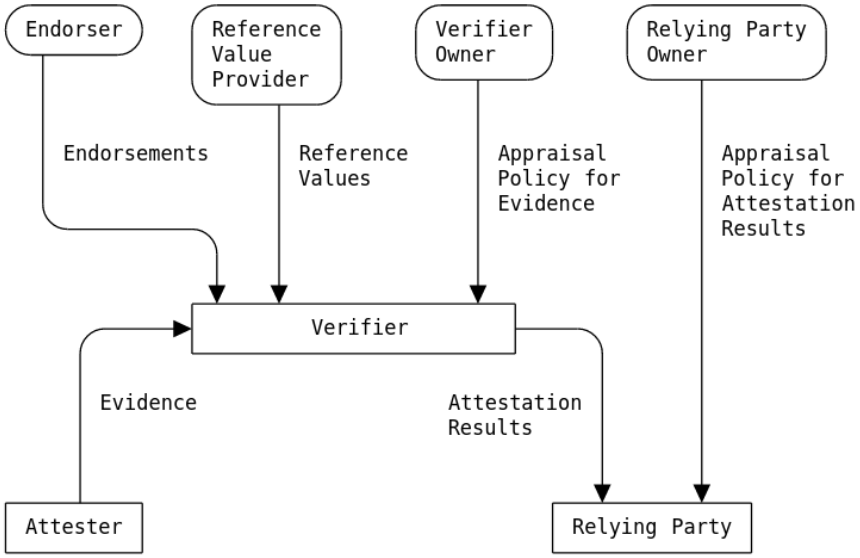

Figure 3.6: IETF Remote Attestation Conceptual Data Flow [20]

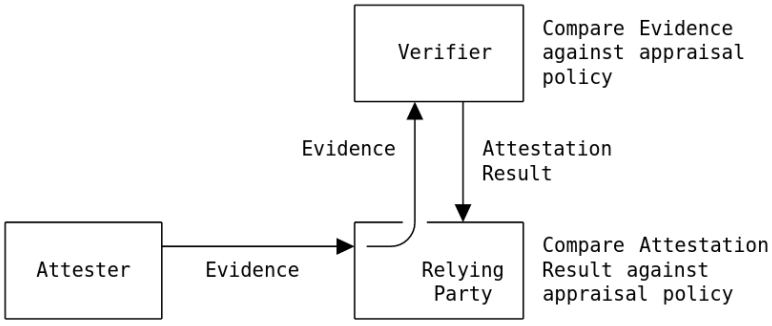

Figure 3.7: The IETF RATS Background Check Model [20]

3.6 Entropy

Entropy as a Service (EnaaS) seeks to make use of the always connected nature of the internet to allow greater leverage of previously hard to access technology. More specifically it seeks to allow the rapid demand and delivery of good quality non-deterministic random numbers to clients that do not possess the funds or technology to generate them in-house. EnaaS could be described as a cryptography-enabling service, in that it provides the primitives required to perform key generation and encryption but does not actually carry out any cryptographic functions itself. In this respect it is aimed at cloud architectures and IoT devices. Cloud provisioned VMs are often from a single golden image which leaves individual instances with very little sources of randomness and IoT devices lack the necessary processing power [111].

NIST has developed a reference architecture for EnaaS along a fairly simple client server pattern [111]. A client sends a timestamped request to an EnaaS server over HTTP, along with their public key. The EnaaS server accesses a high quality source of randomness (such as a quantum random number generator) and returns a time stamped random bit string, encrypted with the client's public key and digitally signed by the server. Current guidelines for using EnaaS are that multiple EnaaS values must be obtained from more than one source, which are then combined with local sources of randomness which then becomes a seed for a non-deterministic random bit generator.

There are numerous examples of EnaaS offerings, some offering a very simple transactional business model whereby a client pays per amount and length of random numbers. Some product offerings are disingenuously termed as 'keys' [90] and other vendors currently offer this service for free. It is still incumbent on the client to ensure these sources are appropriately combined with others using a recognised and standardised cryptographic key generation technique. The majority of the products offered highlight the use of a quantum random number generator, although some research has proposed solutions which use disparate sensor data as a source [109].

3.7 Bundled Services

The "XaaS" term is rooted in the NIST definitions of cloud computing deployments [73] however, as is shown in this chapter, it is also a popular marketing tactic. There are a wide array of vendors offering "Cryptography as a Service" however the offered services vary widely. The types of services offered fall into the following broad categories:

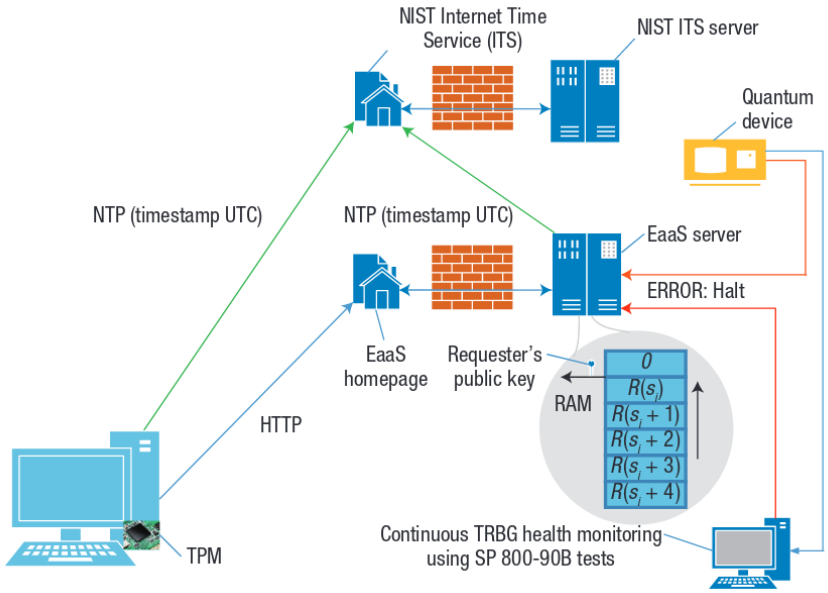

Figure 3.8: NIST Entropy as a Service reference architecture [111]

Key Management and Managed PKI. These offerings are for key management and PKI management, often with HSMs for key storage management included. KMaaS has already been discussed earlier in this chapter

Orchestration and Management. These services offer software based solutions for the management of not only keys but also HSMs which are configured either as key stores, or to carry out cryptographic operations (such as key generation, or encryption/decryption) in dedicated and trusted hardware. Many also offer dashboard, event management and logging monitoring solutions for both keys and HSMs.

Cryptographic/Security Platform. These offerings integrate cryptography applications (such as KMS or secrets management) with other security such as Identity Management, Access Control and Email Security.

Consulting. Many of these offerings claim specialised knowledge in the design, implementation and management of cryptosystems and their integration into other solutions. Some services are for vendor agnostic consulting services, which offer design and implementation services for other products.

Due to the variety of bundled services on offer they cover both cryptographically enabled and enabling solutions. Deployment models can also cover hybrid and private cloud

solutions as well as public cloud options. Business models are similarly varied, with subscription, license based and fee based models all offered.

3.8 Taxonomy

From the above mentioned use cases a taxonomy for CaaS as it is used today is proposed a Fig 3.9. In addition, taxonomies for SECaaS and Key Management were also consulted.

One notable observation from this taxonomy is how much the current service and product offering for CaaS differs from the academic research. Two areas where this is obvious are in encryption of data at rest and encryption of data in use. The academic research for EaaS solutions are heavily based in exploring ABE implementations and FHE schemes, however actual product based implementations of these schemes are nowhere in evidence. This taxonomy is aimed at understanding services, deployment and real world implementations therefore novel schemes like ABE and FHE are not included. In order to stand the test of time, the taxonomy should be able to accommodate those schemes if and when they become actual products.

In a similar vein, although there is a wide body of research on using CaaS for the internet of things, the industrial internet of things and more disconnected computers, no actual products or real world implementations could be found for this at the time of writing. IoT is still included in the taxonomy as it is a possible deployment model and IoT is much in use today.

A second observation is that taxonomy generally holds up to the view of cryptography that covers a set of technologies that can either enable another security requirement such as confidentiality through the use of encryption or that the use of cryptography itself is enabled through the use and management of systems and primitives such as key management and entropy. This is important when considering CaaS as it is the outcome of what the service enables that should be assessed, however this outcome may itself contribute to second order or third order outcomes. For example, a well run and configured KMaaS offering may contribute towards the successful use of cryptography in an organisation, but doesn't achieve this end on its own. Following on from this is the point, which has been made before by other researchers considering cryptosystems, that the use of cryptography is best considered as part of a wider security architecture, and the interaction between all elements of that architecture must be thought through

Figure 3.9: Proposed taxonomy for CaaS

carefully [97].

It is this difficulty with language when looking at CaaS that may explain why it is not explicitly called out in the CSA SECaaS offerings of technology, which proposes Encryption as a Service (EaaS) and nothing more [30]. By separating out this single use of cryptography, it arguably is made easier to publish guidance that is coherent and not contradictory. Other SECaaS categories such as Data Loss Prevention and Network Security make use of cryptography, while the importance of enabling systems such as a KMS are highlighted by the CSA but not considered a service in their own right - despite much evidence that these services do exist and are for sale.

The final observation is that the deployment models follow cloud deployment models and cloud service models very closely. This recognises that the rise CaaS and the other XaaS models appear to have largely been driven by the adoption of cloud technologies. Later stages in this report will consider whether the variety of service models and deployments introduce different considerations and trust implications for clients considering adopting CaaS.

Chapter 4

Analysis of Relevant Standards

This chapter will review the current standards based landscape in which CaaS offerings may operate. The aim is to discuss those aspects of well known legislation or standards that are pertinent to CaaS and what might be a factor or worth considering when adopting it. As well as published standards and legislation the chapter will also survey any industry specific certifications, review pertinent protocols, discuss sector specific guidance and government guidance that might apply.

4.1 Management Framework Standards

Much like legislation, there are a huge number of standards and specifications for cryptography, cloud provision and cloud security as well as for security management and service management. The standards most relevant to CaaS will be reviewed in this section, covering management frameworks, relevant technical implementation standards and the standards validation process.

ISO/IEC 27000 Series. The ISO/IEC 27000 series gives an organisation a holistic approach to establishing and reviewing a security management system. It covers a range of practices and controls, both technical, procedural and people-based. At the heart of the ISO 27000 is the risk management process which is used to inform the implementation of specific security controls for the organisation. As ISO/IEC 27000 is a management framework, aiming to cover as many sectors as possible, it veers away from being overly prescriptive in any form. When it comes to cryptography, the series only states that an organisation should have an approach to the use, implementation and manage-

ment of cryptography (including key management) but does not specify anything further. The series contains a large number of publications giving advice on process, controls and sector specific guidance, an exhaustive review of all of these publications would be impractical and of little value. However it is worth highlighting the guidance found in ISO/IEC 27017.

ISO/IEC 27017 is a part of the series that gives specific guidance on security management of cloud provided services, and there are three considerations here worth highlighting. First is that there may be gaps between the information security requirements of a customer and the security services provided by a CSP, and that a customer may have to implement additional controls to mitigate risks brought on by using a CSP [52]. On the implementation of cryptography, guidance to the customer is to ensure that any cryptography used by the CSP is compliant with the customer's own policies [52]. Finally on the implementation of key management the guidance to the customer is to ensure the CSP provides plenty of detail of key types used and the procedures of the key management life cycle [52]. This assumes the customer is knowledgeable enough to understand what the CSP is stating about their KMaaS offering.

Certification for an organisation is against ISO/IEC 27001, which contains the actual requirements for implementing and running an information management system. Guidance from the other publications in the family can and should be followed, but ultimately is not mandatory if an organisation can justify following another form of practice.

ISO/IEC 20000. Like the ISO/IEC 27000 series, the ISO/IEC 20000 standard sets out how an organisation can deliver a service management system. Again, the standard is not prescriptive in its controls, it simply states that a managed service should use them. Pertinent aspects of the standard for this study centre around scope definition [54], clear definition of accountability on the part of the service provider, mutual understanding between a provider and customer on what is to be delivered and the importance of service monitoring, regular reporting on performance and communication [53] - with specific requirements for the outcomes delivered by the service to be measurable and monitored. These requirements will be explored in more depth in later chapters.

Due to both standards similarity and overlap in many areas, ISO 27000 also provides guidance for an organisation wishing to implement both 27000 and 20000 (ISO 27013). ISO 20000 is also certified, with many providers offering both ISO 27001 and ISO 20000 certification. ISO/IEC 20000 is aligned to other popular service management frameworks such as ITIL.

ISO/IEC Guidance Publications. Many ISO/IEC publications are not standards, rather they are guidance publications. The most relevant to CaaS is ISO/IEC 11770 which provides guidance on how to develop and maintain a key management framework. Unlike ISO 27001 and ISO 20000-1 it does not have a list of requirements for key management, meaning that compliance against it cannot be assessed or certified. ISO 27002 calls out to this standard for guidance in developing a KMS, and this standard in turn refers to other cryptographic implementation standards in the ISO series. Relevant findings from this standard are in its overlap with the ISO 27000 family and ISO 20000 in that it acknowledges a KMS may be outsourced and the need for robust audit and compromise detection. The standard also includes a number of recommended key management and key distribution protocols for both symmetric and asymmetric key types, with specific definitions on the security service provided by each protocol, such as mutual authentication or forward secrecy. A 2016 study found that the definitions for several protocols were flawed, after a finding and updates to a similar standard (ISO/IEC 9798 Entity Authentication) were not also applied to the entity authentication protocols described in ISO/IEC 11770. The latest version of the standard has not altered the protocols but has changed the security service assertions for each to reflect the challenges made in the study [29].

NIST SP 800-57 Key Management. This publication, much like ISO 11770, provides guidance and recommendations for key management. The two have areas of similarity and overlap however also have areas of advice that differ. The NIST publication contains application specific key management guidance for popular use cases. Data at rest is considered with guidance for file encryption use cases [14]. Often the two publications complement each other, for example SP 800-57 will recommend where to use keys derived from passwords however ISO/IEC 11770 contains actual management protocols for key derivation from weak secrets such as a password.

4.2 Cryptographic Implementation Standards

There is a host of technical standards for cryptographic implementations, sponsored by a variety of different standards organisations. Much like the legislation section, it would not be helpful to provide an exhaustive list of every possible standard, the relevant organisations and the standards are listed in 4.1.

This table is not intended to be definitive. It does show however the variety of organisations that maintain standards. It is important to understand which organisation sponsor

Organisation or Company	Standards and Guidance
ISO	Numerous, of note are ISO/IEC 19790 Requirements for Cryptographic Modules as well as ISO/IEC 25759 and ISO/IEC 20540 which provides standards for the testing of those modules. Many standards aligned to FIPS.
NIST	Provides standards for symmetric ciphers, digital signatures and hash functions. Actively working on standardisation for Lightweight Cryptography (LWC) algorithms as well as Post Quantum Cryptography (PQC) algorithms. Numerous SPs on implementations for cryptographic primitives.
RSA	Developed the Public Key Cryptography Standards (PKCS) family. The standards still in use are now managed by the IETF and OASIS.
IETF	Maintains standards for implementing popular stream ciphers such as ChaCha20-Poly1305 as well as implementing protocols such as TLS 1.3 and IKE. Also took over those PKCS standards still in use from RSA.
OASIS	Took over the maintenance of PKCS11 Cryptographic Interface Specification from RSA. An important interface requirement for HSMs. Maintains Key Management Interoperability Protocol (KMIP) in use in many KMS offerings.

Table 4.1: The main organisations maintaining cryptogrpahic implementation standards with a brief discussion of which implementations they support

which implementations as it has implications for which can be validated and how. The later section on validation will go into this in greater detail.

4.3 Secure Hardware

As described in the previous chapter, Hardware Security Modules (HSM) or Trusted Platform Modules (TPM) are present in numerous use cases for CaaS offerings such as Key Management or Secure Compute. Unlike the cryptographic implementations standards in the previous section however, alignment to secure hardware standards is often featured in the marketing for CaaS offerings, therefore merits a more in-depth

analysis and discussion on what this means for a CaaS customer.

Trusted Computing Group. The Trusted Computing Group has developed a library of specifications for a TPM, listing numerous features. Vendors then select the desired features to incorporate into their product design.

FIPS 140-3. NIST has developed standards for cryptographic modules which could be in software or hardware. The Federal Information Processing Standard (FIPS) 140-3[1] is aligned to ISO/IEC 19790 and is aimed at US government agencies, however products with this certification are sold to other markets and possessing a FIPS 140-3 certificate is often used as a marketing feature. FIPS 140-3 allows for four different security levels. The cryptographic implementations do not vary from level to level, rather the difference between them can be found in the other controls that are implemented such as requiring access control, multi-factor authentication or, in the case of hardware products, protections such as anti-tamper mechanisms. For example a level 1 certified module would only require role based access control to be implemented, while a level 4 module requires access control based on multi-factor authentication.

4.4 Validation and Certification

Reference to various standards is useful however without any validation of whether that standard has been implemented properly the customer is no more knowledgeable.

ISO Certification. Organisations can be audited against and certified for the ISO 27001 and ISO 20000 standards. This is a paid service delivered by an accredited auditor who will operate under a country specific certification body. Validating ISO certification is slightly more effort on the part of a customer as, unlike with NIST, there is no central register of certified ISO 27001 compliant organisations. Rather the customer is reliant upon a vendor providing their actual ISO 27001 certificate, then cross referencing that with the certifying body.

One huge attraction for a CaaS customer in seeing whether a provider is ISO certified is that it provides a degree of trust in the competence of that provider, therefore it is an important factor to consider. However a customer should never just rely on seeing an ISO badge on a provider's website. There is still a requirement for a customer to

[1] FIPS 140-3 was published in 2019 and formal submissions began in 2020. Due to the lengthy validation timeline, vendors could submit modules for validation against the previous version FIPS 140-2 until late 2022, with FIPS 140-2 certifications remaining valid until 2026.

Figure 4.1: Use of the FIPS 140 logo is very carefully controlled by NIST. Vendors must follow specific guidelines regarding its use on product website and may only use specific phrases in conjunction with it. Vendors found to be misusing it may be subject to legal action.

be knowledgeable on their requirement, and able to foresee where outsourcing some security functions may lead to additional risks and controls.

NIST Validation Programmes. NIST has two dedicated programs aimed at validating the build of a cryptographic processing module as well as the implementation of those cryptographic algorithms that it has standardised. The cryptographic module validation programme is aimed at both hardware and software products for FIPS 140-3.

The current approach for validation is for a vendor to use an approved testing facility, however the proliferation in products seeking validation in recent years has led to long backlogs and delays at testing facilities. Another complication is that the module is assessed against one of the aforementioned security levels and therefore only validated for correct operation if used within the surrounding security architecture defined at each level [55]. A supplier can market a validated module, but an improper integration into a wider security or cryptographic architecture may mean it does not perform as stated.

Finally a module is only certified when placed in a specific operational environment, which could cover supporting technology such as operating systems and firmware. If any change is made to the operational environment the module must be resubmitted for validation. NIST is now seeking to overhaul this process with the Cryptographic Module Validation Program, bringing down waiting times through the increased use of automation in testing and reporting tools. Furthermore, better standardisation of tests will allow suppliers to carry out some testing in house and then report these results to NIST, further accelerating the testing cycle [101].

The NIST Cryptographic Algorithm Validation Programme (CAVP) will verify algorithm implementations as well as entropy sources [82]. It follows a similar model in that approved third party testing facilities can perform this service for a vendor. NIST provides

Figure 4.2: Although no new modules are being submitted for validation, FIPS 140-2 certification is still valid until 2026

a searchable list for all approved implementations and most popular cryptographic libraries can be found here. It is a prerequisite for vendors seeking FIPS 140-3 certification for their algorithm implementations to receive this validation, so does not hold certification in its own right.

The NIST validation program and the FIPS publications are undoubtedly excellent resources however it must be borne in mind that while they are widely referenced and used by many organisations and vendors, they are aimed first and foremost at assisting US Government agencies. Although aligned to international standards, they only operate within their own ecosystem - for example no FIPS approved module will have implementations of non NIST approved algorithms which means that they might not be appropriate for every encryption use case.

There is much discussion on the value of FIPS 140 certification, and it has been argued that it is holding back any innovation in cryptographic implementations [114] and that certification only shows that the modules function correctly. Certainly the CAVP is very deterministic in its testing, in-depth code review which could catch serious design flaws is not in scope [43]. FIPS 140-3 is still considered to be immature, with many gaps and questions about how to practically design for it [59].

Trusted Computing Group. The TCG will issue vendors certification of functional compliance against the standard and provide a certified products list on their website. Compliance testing is carried out by the vendor, against a TCG approved test plan, with an automated TCG developed test suite. This certification only proves the product behaves

as required, as the TCG adds an extra requirement on vendors for their products to also demonstrate Common Criteria Security Assurance Level 4 [103].

4.5 Guidance and Architectures

The Cloud Security Alliance. The CSA has defined specific Security as a Service (SECaaS) offerings as well as architectural and implementation guidance for each offering. The most relevant publication is its guidance for Encryption as a Service. Since the original SECaaS list was drafted in 2016, more and more offerings in the XaaS vein have come to the fore, and in 2024 it also published guidance for HSM as a Service. The guidance found in both publications can provide a reference architecture for an entire EaaS or HSMaaS solution, and makes regular reference to cryptographic standards outlined above.

- EaaS. This publication provides considerations for both providers and customers of EaaS. Architecture patterns and implementation guidance is given for EaaS for data at rest, data in transit and data processing. The importance of key management is stressed at several points in the publication, with three possible key management models outlined, ranging from fully under customer control to fully under a CSP control. Some relevant technical standards are mentioned, interestingly no management frameworks are referred to [30].
- HSMaaS. This publication acknowledges the growing use of HSMs across a variety of applications in the cloud and provides guidance for service providers. It specifies the split of responsibilities between customer and provider according to the different cloud service layers, giving suggestions for splits at the Infrastructure as a Service Level, Platform as a Service level and Software as a Service level. Consideration is given to relevant standards, applicable security controls, management and monitoring, key management and vendor selection [19].

These guidance documents are included for consideration in this chapter, despite not being standardised, because they form extremely useful advice and guidance for both a customer and a vendor of a CaaS offering. They are useful sources of requirements for a customer and provide valuable architectural patterns for a vendor.

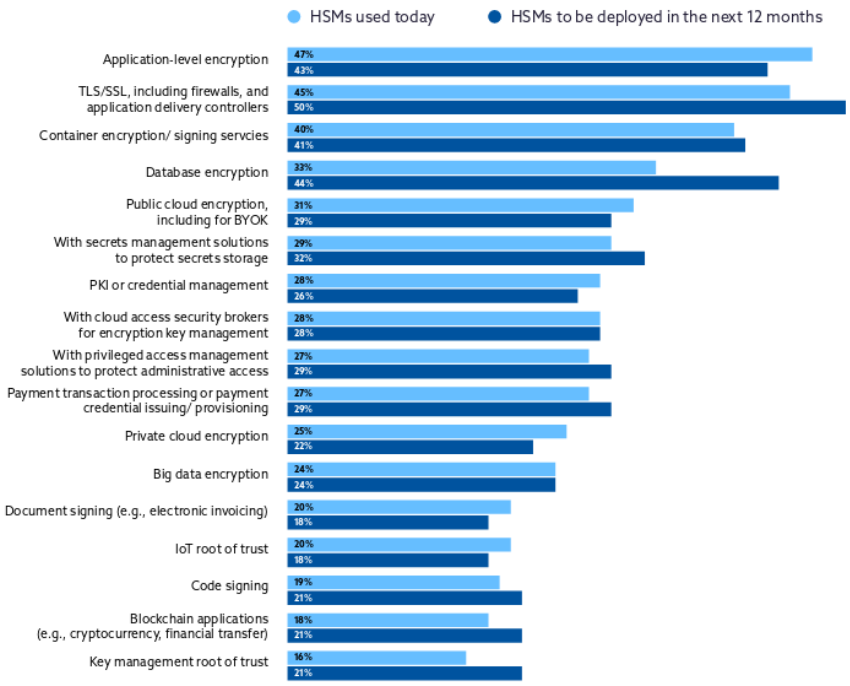

Figure 4.3: A comparison of HSM use cases, deployments in 2022 vs planned deployments for 2023. [36]

4.6 Discussion of Findings

When considering the findings of this chapter, we must bear in mind that this is not intended to be an exhaustive guide to every pertinent standard for CaaS. The aim of this study is to determine whether a CaaS offering can actually be properly assessed by a prospective customer. Therefore when looking at standards and certification, the question a customer should be asking them self is *"When a vendor states that they are compliant with a certain standard, what is that telling me?"*. Another useful question a customer could ask them self is *"What standards could inform the requirements I place on the vendor?"*.

As is seen here, standards could be used as an indicator of some quality (such as FIPS) or as a source of advice on which to design a system (such as ISO/IEC 11770). Not all standards are supported with certification, and those that are will mean extra expense to the vendor in achieving this. Not all testing regimes are the same, and a customer may find themselves with a service that has a multitude of certificates which still don't support their requirements. The pros and cons of FIPS 140-3 certification have been discussed elsewhere in this chapter, and it is an excellent example of vendors using a standard aimed at government agencies as a marketing tool. However, it could also be viewed as a useful benchmark of quality, and a measurable aspect of a managed service, as required in ISO/IEC 20000.

The findings of this chapter are partly affected by the scope of the topic under study - a more focused search against a specific use case such as Encrypted Cloud Storage or Trusted Computing would have allowed for more in-depth analysis. However the complexity itself also supports the argument in support for adopting CaaS offerings, for example in a customer not needing to be knowledgeable about certain technical implementations. The management frameworks of ISO/IEC 27001 and ISO/IEC 20000 are likely of most relevance to CaaS customer, and in the case of ISO/IEC 27001 they might need to consider not only the vendor's but their own adherence to this standard.

The requirements for the use of specific components such as secure hardware might not be something that is articulated by an insufficiently knowledgeable CaaS customer. A recent study on the implementation of secure hardware in systems found overwhelmingly that compliance to standards and industry specific regulation was the main driver for adoption, with interviewees stating that "unless it's compliance related, it's a nice to have." [106]. The issue with the CaaS offerings which have been discussed as use cases is that none of them is specifically subject to any regulation that mandates the

use of secure hardware, despite there being a clear need for it in at least two of the use cases. There is a risk therefore that unscrupulous providers would not implement them.

No standards for the correct and secure functioning of whole systems were found, therefore a CaaS customer should note that there is a gap between existing standards for individual components and the standards for management frameworks. The trustworthiness and security of systems rather than their individual components is still an area of concern in security discourse [15]. A customer might expect a management framework to fill those security gaps with additional controls, however identifying where the gaps might be remains a challenge.

All standards are necessarily slow to evolve. The finding against ISO/IEC 11770 was published in 2016, yet the new version was not issued until 2018. FIPS 140-3 addressed several perceived failings of FIPS 140-2, notably addressing side channel attacks[121], yet FIPS 140-2 still remained in place for several years after these possible attacks were identified. Furthermore many standards appear to 'nest' within each other with complex management frameworks such as ISO/IEC 27002 referencing out to other publications such as ISO/IEC 11770 or NIST standards referencing out to SPs. The ripple effect of updating a standard which might be referenced by others should not be underestimated, nor the burden in keeping up to date with which standards are applicable to a particular use case.

Guidance publications for some use cases are available, and in many cases they reference pertinent standards. They are generally more accessible than standards publications and should be consulted by both customer and vendors when considering CaaS offerings. Unfortunately they do not cover all the use cases identified in the previous chapter.

Chapter 5

Trust

This chapter will consider to what extent a CaaS customer can trust a provider and what mechanisms and controls exist to allow them to make this decision in an informed manner. It will discuss the choices a customer may have to make around using end-to-end encryption solutions, the rise of the term 'zero knowledge' in cloud encryption, relying on design or organisation controls for trust and existing commercial trust frameworks.

5.1 Previous Work

The issues of trust in cryptosystems are outlined by Balsa et al in their 2022 paper titled, "Cryptography, Trust and Privacy: it's complicated." [13] In this they consider whether the security and trust assumptions made at the design stage about a cryptosystem can actually ever be relied upon in reality. The paper is worth further examination because it specifically deals with outsourced solutions which claim to offer cryptographic services (although it does not term them as such).

The paper compares two approaches to trust in confidentiality protection. The first views the service provider as a trusted party that provides a service based on a contract or assertion, the paper terms this privacy by policy or process oriented privacy. The second views the service provider to be an untrusted party, and confidentiality is delivered through the cryptosystem that the service provider offers. This is termed privacy by design. The paper follows the use cases of a secure web browser and a secure messaging app to compare the two approaches. The analysis offered in this chapter is to build on the arguments offered in the paper by considering encrypted cloud storage and trusted

execution environments through the same lens as it considers two other use cases.

5.2 Trust in ECS

For the use of encrypted cloud storage (ECS), different architectures are available to the customer and have been outlined in Chapter 3. The two main patterns are that of client-side encryption (CSE) and server side encryption (SSE). There are numerous commercial offerings, for CSE which claim “zero knowledge” ECS, ie that the service provider is incapable of breaching the confidentiality or integrity of client data. It should be stressed that use of the term zero knowledge in this context is not a recognised cryptographic term and is used purely for marketing purposes. A more accepted term is End-to-End Encryption (E2EE) however many E2EE implementations will require a customer to use a key management system.

The generic design for all of these zero knowledge offerings is based on a password based key derivation function (PBKDF) which not only generates a key with which the user can authorise to the server, but generally also generates a set of symmetric encryption keys and asymmetric key pairs at the same time. This set of keys is associated with the user and is normally encrypted with their master key - often derived from the first PBKDF process. Example key hierarchies from two providers are shown at Figure 5.1 and 5.2.

For CSE the expectation is that documents are encrypted by the client prior to being sent to the cloud. If this is to fit the SECaaS model, which aims to reduce management burden, and expertise from the client, then this is implemented by the client installing an app from the service provider and the app handles encryption and key management. In the SSE model, the document is sent to the CSP and encryption takes place there, using a key controlled by the CSP.

Almost always these zero knowledge offerings include both the encryption function and the KMS as part of the offering. The KMS is based on the customer knowing a secret (a password) that inputs into a Key Derivation Function but is not known by the provider, leading to the claim that the provider is not capable of seeing the customer’s data.

Along with the rise of zero knowledge encryption offerings has come a corresponding rise in the number of security analyses of these offerings, showing many of them to be flawed [11] [119] [4] [71]. These analyses often make two types of findings. The first is that many of the designs of CSE ECS cannot combine cryptographic primitives into a

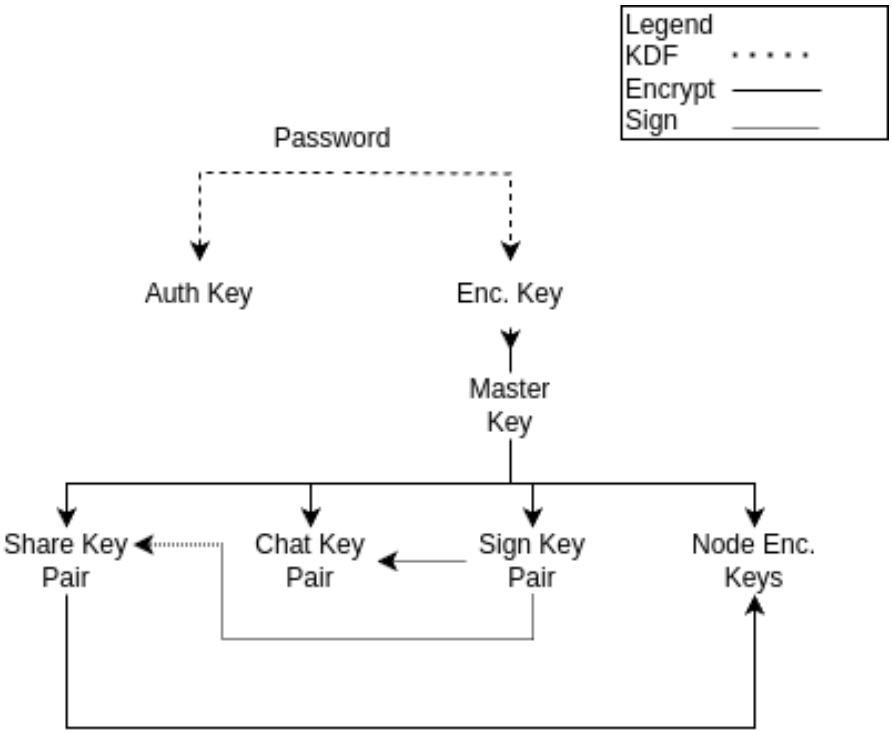

Figure 5.1: A key hierarchy for the MEGA CSE offering [11].

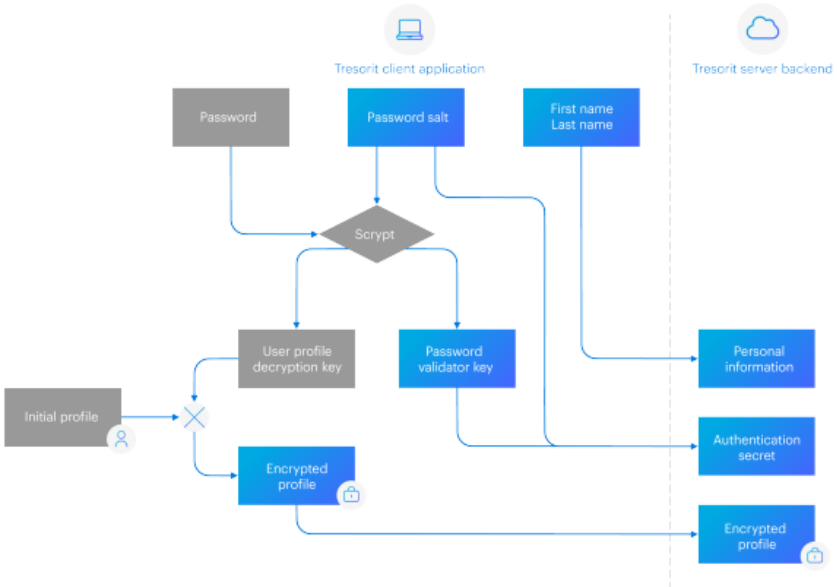

Figure 5.2: Another implementation this time from Tresorit, showing an authentication key and encryption key generated from a password [108]

secure cryptographic system. The second is that poor implementation of some software (such as the client side encryption application) allowed for attacks. A summary of these findings is shown at Table 5.1.

Provider	Findings	Date	Outcome
Tresorit	Possible that integrity protection of file share cannot be promised due to provider acting as its own CA. Could not be proved as source code could not be obtained. [119]	2014	Provider still in operation.
SpiderOak One	Numerous password recovery attacks from the CSE app were possible due to poor implementation of KDF functions, authentication protocol, RPC procedure to client app, key rotation policy. [4]	2018	Responsibly disclosed. ECS offering withdrawn in 2020.
SpiderOak One	Integrity protection of file share cannot be sustained due to provider acting as own CA. [119]	2014	As above
Mega	Key recovery attack and integrity attacks possible. No integrity protection for key ciphertexts, leading to key recovery attack. No origin authentication when encrypting keys. Poor key management. [11]	2022	Responsible disclosure and security flaws allegedly fixed, still offered.
NextCloud	No origin authentication for keys allowing a malicious server to conduct a key insertion attack. IV reuse on AES-GCM implementation enabling known attacks. [71][81]	2024	Responsible disclosure and security flaws allegedly fixed, still offered.
Proton	No data origin authentication on public keys could allow a key insertion attack (no proof given). Unwieldy key hierarchy. [74]	2020	Responsible disclosure and security flaws allegedly fixed, still offered.

Table 5.1: Summary of recent security analyses on major CSE offerings claiming ”zero knowledge encryption”

It is difficult to determine whether these studies actually made the service offering better. While the findings are published in academic journals and responsibly disclosed to the provider, very few are available open source. Some CSPs simply changed the claim they were making in regards to zero knowledge encryption, replacing it with a

weaker claim [4]. These examples reinforce Balsa's argument that privacy by design still requires some verification that the design is fit for purpose.

However the fact that these providers can even be meaningfully analysed is worthy of mention. There are still numerous providers offering zero knowledge encryption that obscure their design such that it can't be analysed [51] [105]. The customer therefore has no choice but to trust the implementation of a service provider that claims there is no need to trust them at all.

In more than one security analysis the observation is made that although the use case of end to end encrypted messaging has been the topic of much study in the cryptographic community, the same effort has not been put into deriving secure and robust systems for end to end encrypted / zero knowledge cloud storage [12] [81]. The lack of accepted standards or architectures for this type of offering therefore results in CSPs providing their own cryptosystem design, while not necessarily possessing the correct cryptographic expertise.

5.3 Trust in Confidential Computing

The TEE was described in Chapter 3. For this analysis we will consider the Remote Attestation (RATS) architecture described in RFC 9334. TEE is covered as a use case for RATS in the document and we must consider the relationship between the Attester, the Verifier and the Relying Party. The basic flow of information is that the Attester provides evidence of a claim of fitness to a Relying Party. A Verifier also checks this evidence against a pre-defined policy and it is the combination of the evidence and the Verifier's response that enables a Relying Party to accept an Attester's fitness for use. In the RFC, the TEE model has a Trusted Application Manager (TAM) which can act as the Relying Party.

The RFC considers the trust relationships as follows:

Relying Party. The Relying Party must trust the Verifier to honestly appraise the evidence provided by an Attester. The RFC achieves this by having the Verifier show a 'trust anchor', which can be a certificate, public key with additional data or symmetric key. The Relying Party therefore requires a trust anchor store in which to keep and reference trust anchors from the Verifier. This is the most basic level of trust allowed for the in the RFC, stronger proofs require the Verifier to initially attest itself to the Relying Party.

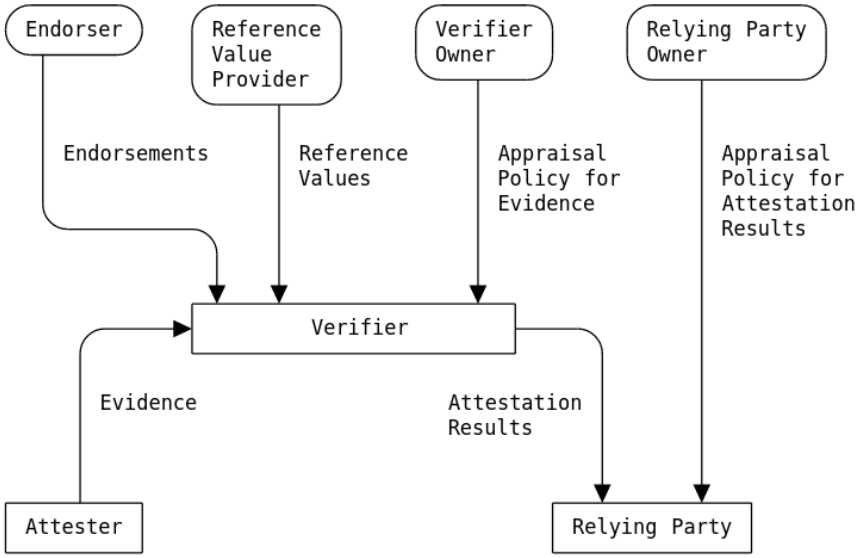

Figure 5.3: Conceptual data flow for RATS [20]

Attester. The RFC posits that some of the evidence that the Attester sends to the Verifier may be sensitive in nature and that therefore it will need to trust the Verifier not to leak this data.

Verifier. The Verifier must trust that the Attester is providing accurate evidence of its state and trustworthiness. This is often indirectly provided though an Endorser, often the manufacturer, provide an attestation key for the Attester which is then stored in the Verifier's trust anchor store. Additional checks must be undertaken the first time the key is accepted into the trust anchor store, either by security analysis of the device, or by checking the key with the Endorser.

In CaaS the trade-off for the customer is where and how much of these trust models to outsource to the CSP. In the CaaS model, the CSP must assure the customer that their TEE is fit for purpose and has not been modified. Therefore it could be that organisationally the customer is the Relying Party while the CSP plays the role of both Attester and Verifier. If this is the trust boundary then the customer will have to maintain their own trust anchor store, as well as their own appraisal policy store. Maintaining a trust anchor store invariably brings in the issue of key management, which must in this case be owned by the customer. Additionally the customer will need control and knowledge of their appraisal policies in order to compare the Verifier's results.

If the issue is that of privacy, then the attestation data and claim made by the Attester to the Verifier may have to be formulated in such a way that no sensitive data is included in this claim (as outlined in the RFC). This is due to an Attester and Verifier likely both being controlled by the CSP. Another consideration is that the customer cannot verify whether appropriate trust relationships have been established between the Verifier and Attester, due to them both belonging to the CSP. The customer is very reliant on the supplier having good supply chain management practices when enrolling devices prior to attestation.

The cloud based TEE use case is in some ways more straightforward than the ECS use case, as it does not assume a privacy by design paradigm, and products do not claim zero knowledge. However RATS is a relatively new standard and there are numerous TEE products offering remote attestation which do not make it clear whether the customer does indeed have control of the elements identified above. Additionally a recent survey has identified that many TEE designs are closed source, which prevents proper security analysis [79]. There are several open source attestation technologies available (with source code published allowing for analysis) but these are not offered in the XaaS model.

The cloud based TEE use case analysis more closely follows the *privacy by policy* model outlined in Balsa, and the need to re-direct trust towards external experts and suppliers, as well as the customer's own policies and controls, is a lot clearer.

5.4 Organisation Controls for a Trusted Provider

Balsa also proposes some technical and organisational controls which could be followed for the privacy by policy approach. These are now discussed for CaaS and in light of findings here and from previous chapters.

Design. The argument here is that the more simple a design is, the more reliably it works and can be more predictably secured. It also supports the observation made above that there is very little thought given to design and protocols for cryptosystems promising zero knowledge, resulting in poorly thought out solutions.

Product Information. The advice is to publish the specifications of the cryptosystem and source code to allow for analysis. Some providers already do this, although the quality of what is published varies a great deal. A stronger control would then be to publish the results of any security analysis alongside these aforementioned documents.

This is reliant upon external experts wanting to / being engaged to undertake this type of analysis for this particular product. Many TEE solutions are not open source and cannot be analysed in this manner.

Encourage or Commission Independent Code Audits. Although a reasonable control, the scope and frequency of these audits will matter. Depending upon the frequency of updates, code audits may be out of date, and frequent code audits may be prohibitively expensive to undertake.

Open Standards. Adoption of open standards and interoperability technology is also suggested. As discussed in chapter 3, using well known standards is a sensible option, yet the challenge remains in properly assessing conformance to those standards. Some of the methods mentioned above may help with this, but these are not ubiquitous and tend to be expensive.

Controls and Policies. Access control policies, logging, audit and incident reporting are all mentioned as controls to support trust. These are also covered in the service management and security standards discussed in the previous chapter. Policies for whistle blower protection, bug bounties, data use and access are also covered in these standards.

5.5 SOC certification - Trust Services Criteria

The American Institute of Certified Public Accountants (AICPA) sponsors certification against a set of technical and organisational controls aimed at suppliers working in high trust industries such as financial services. Suppliers can be assessed against these criteria by an independent assessor in order to receive certification. The controls listed overlap with those suggested by Balsa in some instances, notably in implementing access control, maintaining activity logs, incident reporting and having policies on data access and privacy. They also go beyond Balsa's suggestions by addressing supply chain management (something which Balsa actually remarks upon but then does not suggest a remediation for) and workforce management.

SOC awards are at three levels, Level 1 is aimed at financial services while SOC 2 and 3 are aimed at providers of trusted services. Level 2 is an assessment of a provider's controls at a the time of assessment and Level 3 is a review of performance against those controls over a period of time. It must be noted that achieving SOC 2 shows that the assessor is satisfied that the provider's proposed controls and design fit the use

case [1].

One area where Balsa's suggested controls and the SOC controls differ is in the technical controls for publishing specifications, providing source code and commissioning code audits. One control does exist which is to assess the security of new designs and architectures [1] however this is a fairly weak link. The SOC certification is broadly aimed at a general set of service providers who want to provide trusted services, whereas Balsa's suggestion was specifically aimed at cryptosystem providers and is rooted in the principles of cryptography development.

Chapter 6

Quality

ISO/IEC 20000 and ISO/IEC 27001 mentioned the need for regular monitoring and reporting of a managed and secure service and it can be argued that this concept is an important defining characteristic of CaaS. In order for a service to be monitored it must be given a set of performance metrics against which to be assessed. This section will consider the notion of service quality, what metrics might be applicable to CaaS offerings, how they might be measured and monitored and any challenges that may arise when doing so.

Measuring a quality of service is not a new concept in computer networking or cloud service provision. In networking, Quality of Service (QoS) is often associated with managing and balancing limited network resources to ensure priority for critical or high-performance applications. Factors that may affect performance such as latency, throughput and error rates are considered at the design stage and carefully monitored during operation.

Many studies have proposed quality models for cloud offerings, however only one has specifically proposed a quality model for cloud encryption. This model specifically considers how the quality of Encryption as a Service could be defined. In their 2017 paper, "A Quality Model for Evaluating Encryption as a Service," Wu et al follow the ISO/IEC 25099 product quality model to characterise an EaaS offering for encrypted cloud storage. The architecture they model against is a server side encryption offering, and the chosen quality characteristics are a subset of the eight quality characteristics that make up the ISO/IEC 25099 model.

The Wu paper is an important first step in defining quality criteria however it is an incom-

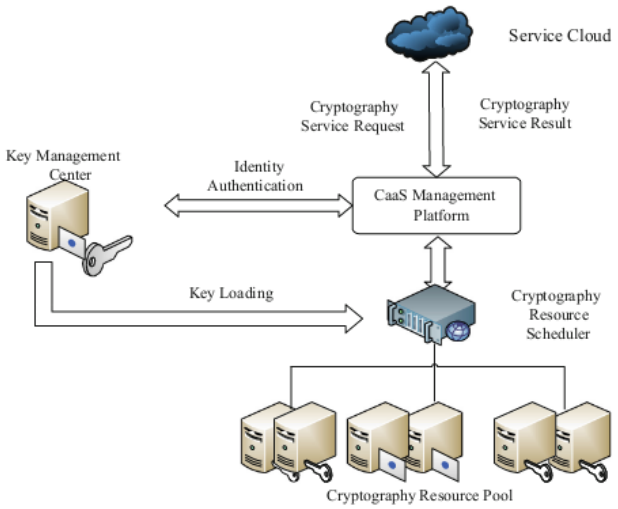

Figure 6.1: The EaaS architecture used by Wu [122]

plete piece of work in that it is not made clear how a CSP might go about obtaining actual values for the criteria it selects. Wu even makes the point that the ISO/IEC 25099 quality model contains quality characteristics that are difficult to quantify or measure. However the approach that Wu et al take towards measuring and choosing their criteria is worth further discussion. Even more so, as the standard actually provides approaches to measurement of sub-characteristics in ISO/IEC 25023:2016, yet Wu sometimes proposes alternative measurements which don't fit either the standard or the use case.

6.1 Security

The chosen sub-characteristics of confidentiality, integrity and non-repudiation are reasonable choices; however not all the measurement approaches make sense.

6.1.1 Confidentiality

Wu's measure of confidentiality is given as the level of entropy of an object, however while this fits well with the stated aim of having something that can be objectively measured, it does not fit well with the use case of encrypted storage. The customer's requirement is simply that confidentiality is maintained, they will not place a numerical value on that requirement as it is likely to be a simple yes or no. The argument that

entropy can be an effective measure of the strength of encryption mechanism ignores factors such as poor key management, side channel attacks, poor implementation and user error or insider threat.

The standard proposes measuring confidentiality through either **access controllability**, **data encryption correctness**, or **strength of cryptographic mechanism**. We must bear in mind that these confidentiality characteristics are very general in nature, aimed to be used for any software product, not specifically a product or service which has confidentiality provision as a primary function. Access controllability might not be a good measure as it considers the amount of objects to which access control is applied held against the number of objects in total. Data encryption correctness cannot apply as again it is function of how many objects are encrypted against how many should be. In an ECS offering that number should be 1, any other result and the ECS offering has failed. We have also seen in the lit review that verifying encryption has taken place is extremely difficult.

6.1.2 Integrity

The metric given for integrity is to compute a value for the amount of data held against a value for any changed data. This gives a ratio which then indicates the probability value of integrity loss. This actually follows the standard, yet, much like the security metric, this suffers from the need to give a numerical value for something that can be easily measured. The metric doesn't work for this use case because it gives a value after the fact of loss of integrity. While some services can accept a degree of loss or corruption (for example voice telephony still works with a small amount of integrity loss because a human listener can still make sense of what is being said), remember that the use case proposed for this study was encrypted cloud storage, which must surely guarantee a total maintenance of integrity for all files stored. Performing measurements of data once already corrupted surely misses the point that any loss of integrity, no matter how small, is likely not acceptable at all to a potential customer.

6.1.3 Non-repudiation

The value for non-repudiation is calculated as a function of the number of files in the system, against the probability that a user's actions with respect to an individual can be logged and tracked. This seems an overly complex measure, the standard suggested measurement of tracking the use of digital signatures for any action requiring

non-repudiation seems more appropriate for the use case.

6.1.4 Findings

Although it would initially be reasonable to include security as part of the model, the flaw here is that the model's intended use is to produce numerical values which can be weighted appropriately in order to conduct a comparison between service offerings. However, while it might be useful for a customer to compare some quality and performance values numerically, doing this with security becomes meaningless. What does it mean to a customer that offering A has a security value of x and offering B has a security value of y if it tells them nothing about the intrinsic quality of the service, only the comparative quality score between two different services.

The standard does have a characteristic that might better capture the quality of a EaaS offering instead of security and this is the functional suitability characteristic. This is the sub characteristics of completeness, correctness and appropriateness. Using these characteristics would allow the model to apply to different types of CaaS offerings with different security properties and functionality. Measurement for functional completeness, correctness and appropriateness is addressed in the principles-based assurance approach which is the subject of the next chapter.

6.2 Efficiency

Wu proposes that efficiency ($EEaaS$) is defined from time behavior and resource utilization:

$$EEaaS = Userve + Uresource$$

Time behaviour ($Userve$) is a function of the wait time and the execution time for encryption to take place. A longer wait time results in a longer time behaviour measurement, while time for execution is assumed to be constant. Resource utilisation is calculated by taking the sum of all resources used to complete a particular operation, CPU, storage or networking and dividing it by the expected resource utilisation for a particular operation.

$$Uresource = ResourceAllocated/ResourcePredefined$$

These efficiency measurements seem reasonable, although they do not follow the measurements from the standard. When considered in light of the use case/architecture the time behaviour measurement could be used for retrieval of documents by the user from the CSP. The resource allocation formula does not really work for the ECS use case offered in the paper, as most service offerings offer pricing based on the amount of data storage used by the customer, something that is after the fact and not pre-defined.

6.3 Reliability

Reliability is made up of the further sub characteristics availability, fault tolerance and recoverability. The measurement approaches for these follow the recommendations in the standard. This characteristic is an absolute cornerstone of every software and service quality metric so its inclusion makes sense. It is difficult to see how this is a characteristic for Encryption as a Service as opposed to a characteristic for cloud storage in general. These characteristics are still worth including as quality measures and almost all major cloud offerings give availability figures.

6.4 Maintainability

The maintainability characteristic is made up of the sub characteristics modularity and re-usability. For modularity the paper proposed a value derived from the number of modules in a system and the number of couplings between these modules. Re-usability in a similar way, with the amount of reusable modules compared to the total amount of modules in the system. Again, these provide a numerical value to enter into a wider comparison calculation, which is the point of the paper. However it does not fit so well for our intended use, which is to inform a set of requirements for a managed service. First of all, it is very difficult for a customer to be able to verify this calculation, and this value could change every time there is a new update to the system which makes up the service. Secondly, it doesn't really fit with the SECaaS paradigm, which is that the customer only cares about outcomes of the service, if a design is not modular but still meets the customer's desired outcomes, then why would the customer insist on more modularity?

6.5 SaaS Quality Models

We can see that the model put forward by Wu cannot actually be fully applied to the CaaS offerings discussed in previous chapters. Part of this is because Wu only considers Encryption as a Service, albeit one in a cloud model, and this project has a wider scope than just that. Another consideration is that most of the products and services discussed in Chapter 3 are delivered as Software as a Service (SaaS). SaaS quality models may be more useful to assess CaaS offerings, although it must be borne in mind that any models chosen will be very general in nature and not specific to any cryptographic services. We will now describe and assess two such models and determine their value to a potential CaaS customer assessing a new service.

6.5.1 SaaS Qual Model

The SaaS Qual model [17] takes as inspiration an earlier well known service quality model SERVQUAL, which was first developed in 1985 and used to gauge customer satisfaction as a measure of service quality. The SaaS Qual model was developed through interviews with people in leadership positions within IT based organisations with experience in consuming SaaS products. The main criteria for evaluation are shown in Table 6.1.

This model focuses very heavily on behaviours, customer service and the relationship management aspect of service quality. It is best used to evaluate a service that is already known to the customer as many of the sub-criteria are difficult to define in terms of data collection and empirical evidence. Furthermore, some of the sub-criteria cannot actually be given a measurement or value such as the sub-criterion 'Choice of billing option' under 'Flexibility' which will either be available or not.

6.5.2 Three Layer Model

The Three Layer model [116] was developed by analysing the features of a SaaS offering and then considering quality aspects for the SaaS Platform, the SaaS Application and the SaaS customer. It also separates security out from general quality metrics and aligns security criteria to the platform, application and customer. The general model is shown at Figure 6.2 and the sub-criteria for the three layers of service quality are shown in Table 6.2.

The model then allows for further assessment by defining four levels of SaaS depending

Criteria	Sub-Criteria
Rapport	Supplier behaviours which demonstrated an understanding of the customer and commitment to joint problem solving and good working relationships
Responsiveness	Technical performance of networks, technical support availability, contingency, disaster recovery, redundancy.
Reliability	Technical performance in line with the SLA
Flexibility	Scalability, choice of feature packaging, choice of billing option
Security	Data backup, security audits, use of encryption, anti-virus protection
Features	Visually appealing, ease of use, dashboard facility, user admin

Table 6.1: The SaaS Qual model quality criteria and the sub criteria

upon which collection of metrics across the entire matrix are met. The lowest achievable level is that of basic SaaS, then as more metrics are met the levels progress to standard SaaS, optimised SaaS with the highest level being integrated SaaS. When considering the criteria in this model, what makes it most useful for CaaS is in splitting out security as a separate metric for evaluation as opposed to including security as part of the quality model. This allows for more appropriate criteria for security to be defined, while still being able to come to a quality evaluation.

This model has useful criteria for quality evaluation of a CaaS offering but there are some flaws. Some of the sub-criteria, such as 'Data Isolation (in Database)' are too general in nature to fit CaaS. Equally some such as 'Service Capability' and 'Software Fault Tolerance' will be difficult for a potential customer to measure prior to actually buying a CaaS offering. Additionally, Wen does not explain their rationale behind what metrics align to what level of offering. 'Location Aware Capability' is aligned to the optimised level, when for some products like ECS that might be better aligned to a standard offering.

6.6 Conclusion

Wu's model is an interesting attempt to try and define quality for Encryption as a Service however it is not fit for purpose for several reasons. Its choices of metrics seem sensible but its proposed methods for measuring them do not fit the use case that it is working with

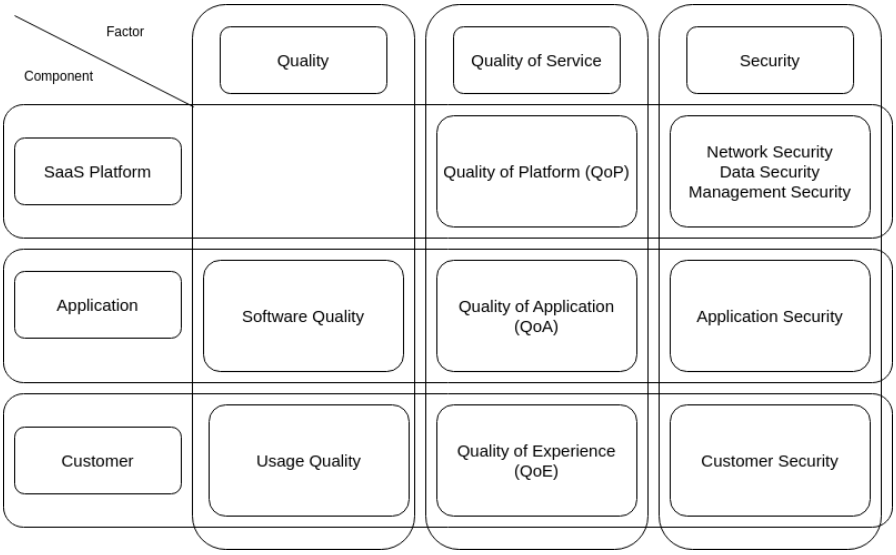

Figure 6.2: The Three Layer SaaS Quality Model

and it does not follow the measurement advice from the standard it uses as a base for the model. It also doesn't explain why it has chosen different methods of measurement to those recommended by the standard. Finally, although supposedly developed as a means for customers to compare different offerings, it does not recognise that customers are likely not going to have access to some of the measurements and metrics required for comparison.

Other quality models for SaaS exist however these have either been developed for the purpose of the supplier to measure their own service internally, or they include metrics which can't actually be measured as they are based on behaviours. The Three Layer model seems like good model for providing an initial assessment of an offering, but as it is intended for any SaaS offering regardless of use case it might be too general.

Criteria	**Sub-Criteria**
Quality of Platform	Transparency Location-aware Capability SLA Management Portability Data Auditing Physical Isolation of Data App Isolation Service Capability Penetration Testing
Quality of Application	Multi-tenancy Configurability Data Isolation (in Database) Response to Customisation Interoperability Software Fault Tolerance
Quality of Experience	Service Availability Usability Performance (Response Time Throughput) Response Timeliness Total Cost of Ownership Return on Investment

Table 6.2: The sub-criteria for the different service layers

Chapter 7

CaaS Market Findings

7.1 Methodology

Following on from the standards based and thematic analysis of CaaS, we now return to the findings of CaaS suppliers from Chapter 3. These providers will now be evaluated in light of the thematic analyses conducted in Chapters 4, 5, and 6 and the findings discussed here.

This market analysis was conducted as an internet search and document analysis [40] using search terms derived from the literature review, and then followed up with search terms based on the product definitions found in Chapter 3. As the analysis was for existing products and services, two major search engines (Google, Duck Duck Go) were used, academic databases were not consulted.

With the exception of TEE and remote attestation, the results do not include services from major CSPs with a global reach and multi-product capability (such as AWS, Azure, Google Cloud or AliCloud), all of whom offered one or more of these services. For some offerings, especially ECS, the variety of providers was extremely varied yet there was almost no difference between these major providers across the evaluation criteria. These companies will still be listed (but not evaluated) as providers in the detailed findings.

For standards evaluation suppliers were assessed on whether they referenced or showed evidence of following any standards discussed in this project. Evidence of certification was noted, as well as where the supplier claims could not be validated.

For trust evaluation suppliers were assessed on how much they could conform to the

technical and organisational controls defined by Balsa. Evidence of holding SOC 2 or 3 certification was also taken into account.

For quality evaluation, suppliers were assessed on what level of the SaaS Qual model could be achieved based on the information available from publicly available sources. Pronouncements made on supplier websites were treated as truthful, although any that could be backed up by audit reports, certification, open designs or open source code were noted.

A finding which is common to all providers is that possession of ISO/IEC 27001 and ISO/IEC 27017 certification can be taken as evidence that the supplier has sufficient technical and organisational controls in place to ensure adequate security management of their service offering and in a cloud setting. This will be discussed later when considering trust and quality.

7.2 Encrypted Cloud Storage and Secrets Management

Analysis and findings for these products have been grouped together as they are common across both product types. As the service provided here is that of encryption, then the concern is with the use of ciphers for that purpose. An additional consideration is that of authentication and zero knowledge encryption and what ciphers and protocols are used for those purposes. All providers can claim use of a particular cipher, the most reliable way to validate this is if the supplier provides source code, or states which cryptographic libraries were used. Many popular libraries have been validated by the NIST CAVP, which can be taken as evidence that they work correctly. What is more difficult to uncover is if the supplier has used those libraries correctly - this can only be ascertained by reviewing the source code, either by the client or by the provider paying for a third party audit. Plenty of suppliers in the survey included this kind of information, however others gave extremely little, sometimes making overly broad statements such as “we use AES 256 encryption” with no further detail on mode of operation.

Mention of use of cryptographic standards was overall so vaguely worded by so many suppliers and missing so much information that it can be inferred to be purely a marketing tactic. Interestingly one supplier made a statement about not using AES, instead citing their use of another cipher Twofish and made misleading statements about AES shown in the screenshot at 7.1. It should be noted that this particular supplier provided very little evidence for all three themes considered in this project.

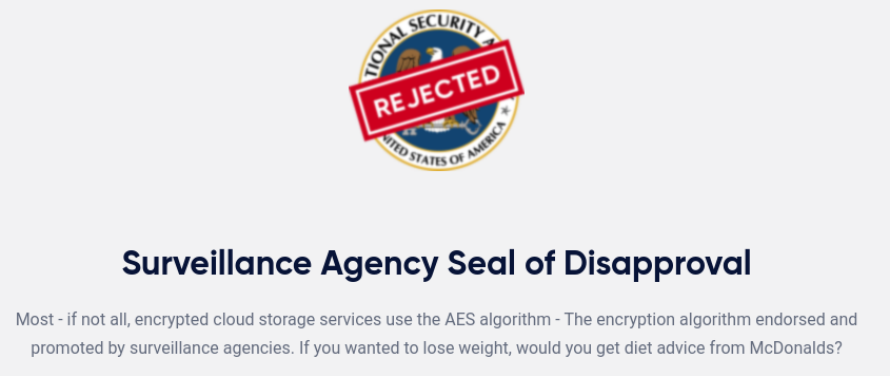

Figure 7.1: Example of misleading statements made by a supplier about AES [51]

In keeping with the findings from Chapter 5, customers considering using zero knowledge encryption offerings from ECS and secrets management suppliers should examine what other technical and organisational controls have been put in place. Possession of in-date SOC 2/3 or ISO/IEC 27001 certification would address some of the organisational controls suggested, however those certifications are not written with open source principles in mind. Results of the survey varied a great deal here, with suppliers running the full range of providing certification and designs and source code and not providing any. It should also be noted that these types of offering were the most likely ones to contain supplier provided designs and to be open source in nature.

All ECS offerings surveyed followed a SaaS model for delivery, however in most cases this was only a Basic offering according to Wen's model. Wens' model considers customer choice over data storage geography and location aware storage as part of the SLA to be an Optimised offering, yet very few providers offered this. Some providers showed a service availability dashboard and others committed to an availability figure (typically 99.9

Similar research into ECS providers was conducted at the same time as this project by Mollakuque et al, with the results published (awaiting peer review at time of writing) in June 2024. The study reviewed four providers, three of which (Sync, pCloud and IceDrive) were also surveyed in this project. The criteria selected for review by Mollakuque et al differed to those chosen by this project and focused on data security controls, the ability to security collaborate and ability to show compliance with various standards and regulations such as HIPAA and GDPR. The use of encryption was noted as a positive, however no further analysis was given to the fact that all providers surveyed only offered zero knowledge encryption, or whether additional trust controls would

be required. Their methodology on feature testing was uneven, while some features such as the presence of multi-factor authentication, could be validated, they state that “encryption protocols were examined” [78]. However the findings of this project show that none of those providers gave sufficient detail to allow any reasonable examination to take place. Furthermore zero knowledge encryption was incorrectly referred to by Mollakuque et al as a standard instead of as a marketing term. On other features their investigation was more thorough than this project, which is to be expected as they only surveyed four providers, and their findings were added to the survey results where appropriate as a secondary source.

7.3 Entropy

There is a NIST publication SP 800-90B which covers the design requirement for entropy sources as well as requirements for validation testing. Validation is achieved through the NIST Cryptographic Module Validation Program and validated sources are listed on the NIST website. None of the providers surveyed held source validation certificates from NIST, however all claimed that the entropy source was quantum in nature. Based on the information provided by vendors this claim is next to impossible to ascertain in the survey.

The trust model for EntaaS is very different from other services covered in this study, as there is no point at which the customer has to trust the provider with their data (other than standard customer data used for billing purposes) and in keeping with every single provider surveyed in this study all EntaaS providers had privacy policies covering the processing of user data. No other trust artefacts or certification were provided by any supplier - however most suppliers did not only offer EntaaS but also offered other services such as Quantum Key Distribution and Post Quantum Cryptography services - which would rely some element of trust in the provider.

Entropy as a Service has been defined as a cryptographically enabling service in the taxonomy as its outputs are intended to feed into other cryptographic operations such as key generation. This assumes that the intended customer for these services is in fact knowledgeable and suitably resourced to manage cryptographic technologies, which means Entropy as a Service does not fit the SECaaS model as denied by the Cloud Security Alliance or academic literature.

There are several well regarded open source projects for entropy, including one that

adheres to the NIST approved architecture. NIST itself hosts a randomness beacon as does the League of Entropy. Commercial offerings in this space focus on the fact that they can provide high quality sources of randomness in a more convenient and automated manner than the open source solutions, often through API calls. However it should be noted that not all EntaaS providers surveyed actually charge for their services, although those providing it free stated their intent to eventually make it a paid for service.

It should be noted that all of the providers found offered EntaaS as one of several other services. All of the other services offered were based around the use of quantum technology in a range of other cryptographic applications.

7.4 Digital Rights Management

DRM technologies are mainly proprietary, with three major providers Microsoft, Apple and Google covering almost the entire streaming market. An effective DRMaaS solution therefore must be able to support all three of these technologies to ensure the best coverage of video content distribution to as many devices as possible.

Trust and privacy concerns for DRMaaS providers are arguably not as stringent as for ECS and secrets management. If a provider was malicious then the threat is that they would steal the uploaded video before it was encrypted or abuse their control of the DRM licenses to do so. No DRM providers surveyed promised any privacy of data expect for that of customer data held for billing purposes. The other aspect of trust is trust in the implementation of their license management system so that a malicious actor cannot steal DRM licenses by exploiting a flaw in the provider's security management architecture. Providers should show evidence of putting relevant measures in place. Almost none of the providers surveyed were able to do this, with only one holding any form of security management certification (ISO/IEC 27001) and one holding trust certification. Most providers offer very little customer control over the licenses, with only offering any form of client side license management control.

Quality concerns for a DRMaaS provider would centre around responsiveness of license issue to end devices and elasticity of the service to allow for rapid scaling, as well as more generic usability metrics. Following the Three Layer quality model, a customer would expect an SLA on availability, performance, response time and usability - which is at the Standard level. Only one of the providers surveyed could anything more than a Basic level of quality, with some not even meeting that due to them failing to show

evidence of any security management processes.

7.5 Trusted Execution Environment and Remote Attestation

Global Platform provides a standard for TEE, however it is not yet universally adopted. All TEE enabling hardware is based on a TPM and the accepted specification for this is held by the Trusted Computing Group. Architectures for TEE solutions differ a great deal and are currently based on products from vendors such as Intel (with the Intel SGX) or AMD (with the ARM TrustZone). A TEE protection profile is part of the Common Criteria certification, products holding this certification can be searched for on their website. The survey was unable to ascertain to what extent existing hardware products used in cloud based TEEs have been validated or certified against any known hardware security standards.

The findings from chapter 5 on trust in cloud based TEEs are that, if following the IETF architecture, a client must trust their CSP to some extent, particularly the relationship between the CSP Attester and a trusted third party Verifier. Trusting this relationship will be entirely dependent upon organisational controls. The client also needs to trust the Verifier, and according to the architecture this can be achieved by them owning and controlling their own trust anchor store.

The survey found relatively few cloud based TEE service providers compared to other types of CaaS offering, almost all of these providers were from large, well established CSPs with global reach. Some provided both a TEE and an attestation service, while two only provided an attestation service. All the major providers unsurprisingly showed strong evidence for all organisational security and service management standards as well as meeting organisational trust criteria.

Cloud based TEE is still a relatively small market, dominated almost entirely by well established CSPs such as AWS and Azure. They provide a high quality of service and can meet many of the trust criteria outlined in chapter 5, although they unsurprisingly do not ascribe to the open design or open source controls. These providers were able to provide a complete attestation service likely because they have the resources to use their own certificate authority as a verifying party, thus ensuring trust between the attester and verifier (by dint of being owned by the same company) and enabling the relying party to trust the verifier through their CA, which is itself backed by a high degree of organisational trust measures.

Only one smaller provider was found, who offered a form of TEE and attestation piggybacking on the services of another major provider. Their trust controls are comprehensive, and their SaaS quality is only at basic standard. This supplier offers other products which differentiates them from major providers, TEE is not their only offering.

7.6 Key Management

Effective KMaaS offerings must be able to offer solutions that work with multiple cloud offerings and across multiple standards. When it comes to managing keys, the KMIP can ensure management of those solutions that have implemented it. Open standards like the KMIP can give assurance of this. If a KMaaS offering also includes HSM management and orchestration then compliance with the PKCS#11 standard for cryptographic module interfaces will allow a customer increased choice and the ability to use different brands of HSM if they wish to. Any HSM included in an offering should of course come with FIPS certification, however the findings of the survey showed that any HSMs included were often mentioned by brand, allowing for relatively simple validation of this claim.

KMaaS offerings should be able to show the same level of organisation technical trust controls as encountered in other offering types. A trustworthy KMaaS offering should offer the customer large amounts of control while still enabling automation of repetitive tasks. Features that allow a customer to import their own keys (known as Bring Your Own Key) and hold master keys on customer owned infrastructure (known as Hold Your Own Key) can ensure the customer ultimately has control of encryption key management.

The survey results showed the most variation in use cases between suppliers. While all suppliers offered KMaaS, they differed in exactly what use cases were supported. Several included the ability for HSM management and orchestration, others actually included HSMs themselves bundled in with the offering. KMaaS offerings stressed the ability to automate the management of large numbers of keys across different cloud platforms, however with all of them a high level of customer knowledge is needed in order to effectively and security implement proper key management life-cycles.

Chapter 8

Conclusion

The aim of this project was to better understand the CaaS landscape as it is being offered by today's CSPs and to enable a security practitioner or potential customer of CaaS to determine whether these offerings could fit well into an existing organisation's security architecture or whether there would be any gaps. An internet search was carried out of existing providers and a taxonomy was developed to better understand the breadth of CaaS offerings, as well as deployment models, billing models and actual security services offered. The relevant standards, trust approaches and frameworks and quality frameworks were reviewed and discussed for their applicability to CaaS and how they might help a customer better understand any challenges and unforeseen complications in procuring CaaS. Finally the results of the internet search were revisited so that the state of the CaaS market could be evaluated against the standards, trust and quality criteria that had been identified. These results were discussed in some detail and an attempt was made to characterise providers against the service taxonomy defined in Chapter 3.

This project was intended to help a security practitioner understand how a CaaS offering could be evaluated using standards, trust evidence and quality models. It was also an attempt to see if today's CaaS providers give sufficient information on the chosen assessment criteria, and whether there are any gaps in guidance and best practice that a practitioner might need to be aware of.

8.1 General Observations

This project started with the view that Cryptography as a Service should be viewed as a type of SECaaS offering. SECaaS already covered Encryption as a Service, however the view of this author was that this was too specific and missed out other cryptographic operations and enablers that are also being sold under a SaaS model by cloud service providers. Some service offerings, while undoubtedly relating to cryptography in some form, turned out to be a bad fit, as in the case with Entropy as a Service, as they didn't fully meet the SECaaS use cases of outsourcing expertise, and instead simply provided an input into a more complex customer managed process. CaaS also allowed for the consideration of integral activities such as KMaaS, which this author argues would require the same level of scrutiny as that of a service offering encryption. This allowed the project to also look at other types of key management, such as DRM licence management, and find that as a sector the DRMaaS market does not have anywhere near the same level of organisational trust controls or SaaS quality scores as other more 'conventional' offerings which appear at first glance to be more closely related to cryptography. The DRMaaS market would merit more in depth security analysis than this project was able to provide as the survey results indicated that most suppliers do not provide an adequately trustworthy service of any measurable quality.

When searching for Entropy as a Service providers, every single one found offered this alongside a variety of other quantum based technologies that were related to cryptography. Some providers offered a Post Quantum Cryptography (PQC) solution for asymmetric encryption, other offered services in Quantum Key Distribution (QKD). These technologies were not considered in either the literature review or the taxonomy and as a result the services weren't included in the findings for other CaaS types.

This could be viewed as an omission in this study, however all providers were still assessed for evidence of standards compliance, trust evidence and SaaS quality evidence, with very mixed results - no supplier in the search held a trust certification or gave any evidence in support of Balsa's trust controls.

8.2 Standards

Management standards were discussed and three major certifications were looked for in the survey which were ISO/IEC 27001, ISO/IEC 27017 and ISO/IEC 22000. Of these ISO/IEC 27001 was the most prevalent among suppliers, but by no means all CSPs held

this. ISO/IEC 27001 certainly shows that the supplier operates a security management system however if a customer is wishing to achieve ISO/IEC 27001 certification and wishes to include a CaaS supplier they should consider how the standard advises on using suppliers of cryptographic services. The controls advice is to ensure robust service level agreements [56], and very few of the suppliers surveyed could offer anything more than a basic SLA which could not be tailored. This could be viewed as the CaaS market offerings in this survey not being mature enough to actually use for customer organisations seeking this certification.

8.3 End to end encryption and ”zero knowledge” encryption

The term zero knowledge encryption was encountered in many ECS and secrets management offerings. The growing market for end-to-end ECS was already noted by Backendal [12] and the findings of this project show that secrets management is another use case where using terms like zero knowledge encryption is increasingly being adopted. Although there is a lively scene of academic security analyses of end-to-end encryption in ECS, the same level of scrutiny has not yet been given to secrets management/encryption engine services. Similarly, there is no useful industry guidance for this, with the Cloud Security Alliance's EaaS guidance stating that customers should use key management practices such as BYOK if they do not fully trust a provider [30]. The CSA guidance on EaaS was published in 2012, long before ”zero knowledge” offerings were common, and therefore their guidance should be updated to reflect this new offering.

What does this mean for a potential customer who is attracted to the notion of ”zero knowledge encryption”? Many providers posted detailed white papers explaining their design, and several also posted source code for validation. Several also showed strong evidence of adopting technical and organisational controls to show that they could be trusted. However a supplier's willingness to act in a trustworthy manner does not mean that their design can be unequivocally known to work correctly. Posting security white papers and open source code is meaningless to a customer unless those designs have been validated by a third party - the assumption here is that most customers will not have the domain knowledge to be able to check the supplier's design.

Another more long term approach would be to standardise a robust PBKDF design and associated key hierarchy for ECS and secrets management. This author is not the first to make this argument [12] and it would undoubtedly ensure more secure implementations. This project argues for it from the perspective of a better consumer experience.

Since the use cases for ECS and secrets management do not differ between suppliers a standardised design would ensure that the customer is selecting a supplier for actual performance metrics that are relevant to their use case, rather than whether a design will work or not. As an example from the survey - many suppliers stated that they used a NIST standard Password Based Key Derivation Function (PBKDF), yet all selected massively different iterations without explaining the reasons for those choices or how they would impact performance. These PBKDFs were then combined with various other ciphers, with slight variations in each design. A customer cannot actually use this detail to compare whether an offering can meet their use case, and this project argues that they are not actually being published for the customer's benefit, they are being published in the hope that the wider security research community will scrutinise them.

The findings from this project are that if a customer has a threat model where the cloud service provider cannot be trusted, their best mitigation is to follow the CSA guidance and combine an offering with a key management solution that ensures they hold their own master key, or consider an on-premise deployment. If they truly want a zero knowledge solution then they should look for a supplier who has had their design validated by a third party, which was a very tiny minority at the time of survey.

8.4 Quality

Quality of service was difficult to gauge with most suppliers offering only a basic level quality of service if we follow Wen's model. Wen's model was the only one in the literature that attempted to give different levels of quality management (from basic to integrated), and not all of their scores were appropriate for the CaaS offerings. For example the ability to have geographical control of where data is stored is placed by Wen on the second highest scoring tier for SaaS Qual, yet for use cases like ECS, location of data might arguably be a more standard offering. This is probably a weakness of the model, as different use cases will likely put different values on certain metrics, as well as a weakness in the methodology, as using this single model to survey numerous different providers gives very shallow results.

However the finding still stands that the majority of CaaS offerings surveyed offered very little opportunity for a customer to determine detailed SLAs with the provider. The expectations to this were the major providers, who could offer varying types of SLA depending upon price point - essentially quality management being a 'you get what you pay for' type situation.

From the perspective of the smaller providers in this survey, committing to a fixed service quality regardless of the customer is a reasonable choice to make given they are likely to have limited resources, however it does show a possible weakness in the cloud SaaS model, in that relatively small providers can scale rapidly in terms of technology, global reach and pure compute power, yet are unable to make the organisation changes required for proper service level definition and service management of different types of customers.

8.5 Recommendations for Further Research

The results of the survey are included in an annex to this paper. This was a wide ranging survey which looked for relevant themes in CaaS and the results on a single provider cannot be viewed as a definitive judgement on their product. Detailed security analyses on some of the surveyed providers do exist, and this remains a fruitful area of study for the more practically inclined. Similarly this paper has identified that certain market sectors such as DRM as a Service appear to be lacking in-depth security based analysis.

The survey results did not control for the cost of the service and it is likely that this is a huge driver of both quality and trust in a provider. Billing models such as subscriptions or per use billing were not considered - any future work in this area would do well to consider how a service provider's size, revenue and billing practices could affect a particular offering.

This project assumed that CaaS offerings could be assessed for quality under a SaaS based quality model. However some offerings were likely a better fit for PaaS or IaaS and more in depth review of some providers using a PaaS or IaaS model might yield more insightful results. Similarly an understanding of how quality for cryptographic services can be assessed at SaaS, PaaS and IaaS and how they might differ would be a useful area of study. One attempt has been made to define a quality model for Encryption as a Service, which was found not fit for purpose in chapter 6. Whether this is achievable or necessary given existing quality models is still an unanswered question.

Bibliography

[1] AICPA. *2017 Trust Services Criteria for Security, Availability, Processing Integrity, Confidentiality, and Privacy (With Revised Points of Focus — 2022)*. 2022. URL: https://www.aicpa-cima.com/feed.

[2] Orestis Alpos et al. "Thetacrypt: A Distributed Service for Threshold Cryptography On-Demand: Demo Abstract". en. In: *Proceedings of the 24th International Middleware Conference: Demos, Posters and Doctoral Symposium*. Bologna Italy: ACM, Dec. 2023, pp. 33–34. ISBN: 9798400704291. DOI: 10.1145/3626564.3629100. URL: https://dl.acm.org/doi/10.1145/3626564.3629100 (visited on 03/04/2024).

[3] P.K. Anders, Dalskov, and Claudio Orlandi. "Can You Trust Your Encrypted Cloud? An Assessment of SpiderOakONE's Security". In: *Cryptology e-Print archive* (2017). URL: https://eprint.iacr.org/2017/570 (visited on 03/20/2024).

[4] Anders P. K. Dalskov et al. "Can You Trust Your Encrypted Cloud?: An Assessment of SpiderOakONE's Security". In: *ACM Asia Conference on Computer and Communications Security* (May 2018). MAG ID: 2806424702 S2ID: 3070a3389218065dfdd1793ba0e21ed4325e11d1, pp. 343–355. DOI: 10.1145/3196494.3196547. URL: https://dl.acm.org/doi/abs/10.1145/3196494.3196547 (visited on 03/19/2024).

[5] Ross Anderson. *Why cryptosystems fail*. eng. ISSN: 0001-0782 Issue: 11 Pages: 32-40 Place: New York Publication Title: Communications of the ACM Volume: 37. 1994.

[6] Ross Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Indianapolis: John Wiley and Sons, 2020. ISBN: 978-1-119-64278-7.

[7] Apple. *FairPlay Streaming*. URL: https://developer.apple.com/streaming/fps/ (visited on 04/15/2024).

[8] AWS. *Amazon S3 now automatically encrypts all new objects*. URL: https://docs.aws.amazon.com/AmazonS3/latest/userguide/default-encryption-faq.html (visited on 04/13/2024).

[9] AWS. *AWS Key Management Service*. URL: https://docs.aws.amazon.com/kms/latest/developerguide/overview.html (visited on 03/15/2024).

[10] AWS. *What is AWS Nitro Enclaves?* URL: https://docs.aws.amazon.com/enclaves/latest/user/nitro-enclave.html (visited on 04/27/2024).

[11] Matilda Backendal, Miro Haller, and Kenneth G. Paterson. "MEGA: Malleable Encryption Goes Awry". en. In: *2023 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE, May 2023, pp. 146–163. ISBN: 978-1-66549-336-9. DOI: 10.1109/SP46215.2023.10179290. URL: https://ieeexplore.ieee.org/document/10179290/ (visited on 03/23/2024).

[12] Matilda Backendal, Miro Haller, and Kenny Paterson. "End-to-End Encrypted Cloud Storage". In: *IEEE Security & Privacy* 22.2 (Mar. 2024), pp. 69–74. ISSN: 1540-7993, 1558-4046. DOI: 10.1109/MSEC.2024.3352788. URL: https://ieeexplore.ieee.org/document/10488835/ (visited on 05/07/2024).

[13] Ero Balsa, Helen Nissenbaum, and Sunoo Park. "Cryptography, Trust and Privacy: It's Complicated". en. In: *Proceedings of the 2022 Symposium on Computer Science and Law*. Washington DC USA: ACM, Nov. 2022, pp. 167–179. ISBN: 978-1-4503-9234-1. DOI: 10.1145/3511265.3550443. URL: https://dl.acm.org/doi/10.1145/3511265.3550443 (visited on 03/12/2024).

[14] Elaine B. Barker and Quynh H. Dang. *Recommendation for Key Management Part 3: Application-Specific Key Management Guidance*. en. Tech. rep. NIST SP 800-57Pt3r1. National Institute of Standards and Technology, Jan. 2015, NIST SP 800–57Pt3r1. DOI: 10.6028/NIST.SP.800-57Pt3r1. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57Pt3r1.pdf (visited on 06/02/2024).

[15] Steven M. Bellovin and Peter G. Neumann. "The big picture". en. In: *Communications of the ACM* 61.11 (Oct. 2018), pp. 24–26. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3277564. URL: https://dl.acm.org/doi/10.1145/3277564 (visited on 06/02/2024).

[16] Fabrice Benhamouda et al. "Threshold Cryptography as a Service (in the Multiserver and YOSO Models)". en. In: *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. Los Angeles CA USA: ACM, Nov. 2022, pp. 323–336. ISBN: 978-1-4503-9450-5. DOI: 10.1145/3548606.

3559397. URL: https://dl.acm.org/doi/10.1145/3548606.3559397 (visited on 01/19/2024).

[17] Alexander Benlian, Marios Koufaris, and Thomas Hess. “Service Quality in Software-as-a-Service: Developing the SaaS-Qual Measure and Examining Its Role in Usage Continuance”. en. In: *Journal of Management Information Systems* 28.3 (Dec. 2011), pp. 85–126. ISSN: 0742-1222, 1557-928X. DOI: 10.2753/MIS0742-1222280303. URL: https://www.tandfonline.com/doi/full/10.2753/MIS0742-1222280303 (visited on 08/04/2024).

[18] T A Berson et al. “Cryptography as a Network Service”. en. In: (May 2003).

[19] Ian Beveridge et al. *Hardware Security Module (HSM) as a Service*. Apr. 2024.

[20] H Birkholz et al. *Remote Attestation Procedures Architecture*. Sept. 2022. URL: https://www.ietf.org/archive/id/draft-ietf-rats-architecture-22.html#name-background-check-model.

[21] Sören Bleikertz et al. “Client-controlled cryptography-as-a-service in the cloud”. In: *Applied Cryptography and Network Security: 11th International Conference, ACNS 2013, Banff, AB, Canada, June 25-28, 2013. Proceedings 11*. Springer. 2013, pp. 19–36.

[22] Abderrahmane Boudi et al. “Assessing Lightweight Virtualization for Security-as-a-Service at the Network Edge”. en. In: *IEICE Transactions on Communications* E102.B.5 (May 2019), pp. 970–977. ISSN: 0916-8516, 1745-1345. DOI: 10.1587/transcom.2018EUI0001. URL: https://www.jstage.jst.go.jp/article/transcom/E102.B/5/E102.B_2018EUI0001/_article (visited on 02/23/2024).

[23] Ismail Butun et al. “Cloud-centric multi-level authentication as a service for secure public safety device networks”. en. In: *IEEE Communications Magazine* 54.4 (Apr. 2016), pp. 47–53. ISSN: 0163-6804. DOI: 10.1109/MCOM.2016.7452265. URL: http://ieeexplore.ieee.org/document/7452265/ (visited on 02/03/2024).

[24] Ramaswamy Chandramouli, Michaela Iorga, and Santosh Chokhani. *Cryptographic Key Management Issues and Challenges in Cloud Services*. en. Tech. rep. NIST IR 7956. National Institute of Standards and Technology, Sept. 2013, NIST IR 7956. DOI: 10.6028/NIST.IR.7956. URL: https://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7956.pdf (visited on 02/03/2024).

[25] Juntao Chen and Quanyan Zhu. “Security as a Service for Cloud-Enabled Internet of Controlled Things Under Advanced Persistent Threats: A Contract Design Approach”. In: *IEEE Transactions on Information Forensics and Security* 12.11

(Nov. 2017), pp. 2736–2750. ISSN: 1556-6013, 1556-6021. DOI: 10.1109/TIFS.2017.2718489. URL: http://ieeexplore.ieee.org/document/7954676/ (visited on 02/20/2024).

[26] Sarra Cherbal et al. “Security in internet of things: a review on approaches based on blockchain, machine learning, cryptography, and quantum computing”. en. In: *The Journal of Supercomputing* 80.3 (Feb. 2024), pp. 3738–3816. ISSN: 0920-8542, 1573-0484. DOI: 10.1007/s11227-023-05616-2. URL: https://link.springer.com/10.1007/s11227-023-05616-2 (visited on 02/03/2024).

[27] Maurizio Colombo et al. “Data Protection as a Service in the Multi-Cloud Environment”. In: *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*. ISSN: 2159-6190. July 2019, pp. 81–85. DOI: 10.1109/CLOUD.2019.00025. URL: https://ieeexplore.ieee.org/document/8814489/ (visited on 01/16/2024).

[28] CPIX. *Content Protection Information Exchange*. URL: https://docs.unified-streaming.com/documentation/drm/cpix_intro.html.

[29] Cas Cremers and Marko Horvat. “Improving the ISO/IEC 11770 standard for key management techniques”. en. In: *International Journal of Information Security* 15.6 (Nov. 2016), pp. 659–673. ISSN: 1615-5262, 1615-5270. DOI: 10.1007/s10207-015-0306-9. URL: http://link.springer.com/10.1007/s10207-015-0306-9 (visited on 05/10/2024).

[30] CSA. *SecaaS Implementation Guidance Category 8 //Encryption*. Sept. 2012. URL: https://cloudsecurityalliance.org/artifacts/secaas-category-8-encryption-implementation-guidance (visited on 01/20/2024).

[31] CSA. *Roles and Responsibilities of Third Party Security Services Providers*. 2021. URL: https://cloudsecurityalliance.org/artifacts/roles-and-responsibilities-of-third-party-security-services (visited on 03/20/2024).

[32] Pallav Kumar Deb, Anandarup Mukherjee, and Sudip Misra. “CEaaS: Constrained Encryption as a Service in Fog-Enabled IoT”. en. In: *IEEE Internet of Things Journal* 9.20 (Oct. 2022), pp. 19803–19810. ISSN: 2327-4662, 2372-2541. DOI: 10.1109/JIOT.2022.3167832. URL: https://ieeexplore.ieee.org/document/9758049/ (visited on 02/03/2024).

[33] Yvo G. Desmedt. “Threshold cryptography”. en. In: *European Transactions on Telecommunications* 5.4 (July 1994), pp. 449–458. ISSN: 1124-318X, 1541-8251. DOI: 10.1002/ett.4460050407. URL: https://onlinelibrary.wiley.com/doi/10.1002/ett.4460050407 (visited on 03/23/2024).

[34] Marwa Elsayed and Mohammad Zulkernine. “A Taxonomy of Security as a Service”. In: *On the Move to Meaningful Internet Systems. OTM 2018 Conferences: Confederated International Conferences: CoopIS, C&TC, and ODBASE 2018, Valletta, Malta, October 22-26, 2018, Proceedings, Part II*. Springer. 2018, pp. 305–312.

[35] Gabe Elton. *Purchasing DRM Services: An Insider’s View*. Mar. 2018. URL: https://go.buydrm.com/thedrmblog/buydrm-purchasing-drm-services-an-insiders-view (visited on 04/22/2024).

[36] Entrust. “2022 Global Encryption Trends Study”. en. In: *RESEARCH REPORT* (2022).

[37] EU. *General Data Protection Regulation*. May 2018. URL: https://gdpr-info.eu/art-32-gdpr/.

[38] Evervault. *Secure Enclaves*. URL: https://docs.evervault.com/security/secure-enclaves.

[39] Saad Fehis, Omar Nouali, and Tahar Kechadi. “Secure encryption key management as a SecaaS based on Chinese wall security policy”. en. In: *Journal of Information Security and Applications* 63 (Dec. 2021), p. 102975. ISSN: 22142126. DOI: 10.1016/j.jisa.2021.102975. URL: https://linkinghub.elsevier.com/retrieve/pii/S2214212621001861 (visited on 01/29/2024).

[40] Nigel G Fielding, Grant Blank, and Raymond M Lee. “The SAGE handbook of online research methods”. In: (2016).

[41] Dan Goodin. *Hackers steal secret crypto keys for NordVPN. Here’s what we know so far*. Oct. 2019. URL: https://arstechnica.com/information-technology/2019/10/hackers-steal-secret-crypto-keys-for-nordvpn-heres-what-we-know-so-far/.

[42] Google. *Default encryption at rest*. Sept. 2022. URL: https://cloud.google.com/docs/security/encryption/default-encryption (visited on 04/28/2024).

[43] Matthew Green. *OpenSSL and NSS are FIPS 140 certified. Is the Internet safe now?* en. Jan. 2012. URL: https://blog.cryptographyengineering.com/2012/01/02/openssl-and-nss-are-fips-140-certified/ (visited on 05/27/2024).

[44] Matthew Green, Susan Hohenberger, and Brent Waters. “Outsourcing the Decryption of ABE Ciphertexts”. en. In: *20th Usenix Security Symposium*. San Francisco, Aug. 2011.

[45] Franz Gregor et al. “Trust Management as a Service: Enabling Trusted Execution in the Face of Byzantine Stakeholders”. en. In: *2020 50th Annual IEEE/IFIP*

International Conference on Dependable Systems and Networks (DSN). Valencia, Spain: IEEE, June 2020, pp. 502–514. ISBN: 978-1-72815-809-9. DOI: 10.1109/DSN48063.2020.00063. URL: https://ieeexplore.ieee.org/document/9153433/ (visited on 01/19/2024).

[46] Object Management Group. *XaaS (Anything as a Service) Glossary*. Tech. rep. Object Management Group. URL: https://www.omg.org/cloud/Anything-as-a-Service-Glossary-22-06-08.pdf (visited on 08/18/2024).

[47] Sheikh Mahbub Habib, Sebastian Ries, and Max Muhlhauser. "Towards a Trust Management System for Cloud Computing". en. In: *2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*. Changsha, China: IEEE, Nov. 2011, pp. 933–939. ISBN: 978-1-4577-2135-9. DOI: 10.1109/TrustCom.2011.129. URL: http://ieeexplore.ieee.org/document/6120922/ (visited on 03/22/2024).

[48] Md. Mahmud Hasan and Hussein T. Mouftah. "Encryption as a service for smart grid advanced metering infrastructure". en. In: *2015 IEEE Symposium on Computers and Communication (ISCC)*. Larnaca: IEEE, July 2015, pp. 216–221. ISBN: 978-1-4673-7194-0. DOI: 10.1109/ISCC.2015.7405519. URL: http://ieeexplore.ieee.org/document/7405519/ (visited on 01/19/2024).

[49] Xiaolong Huang and Ruining Chen. "A Survey of Key Management Service in Cloud". en. In: *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*. Beijing, China: IEEE, Nov. 2018, pp. 916–919. ISBN: 978-1-5386-6565-7. DOI: 10.1109/ICSESS.2018.8663805. URL: https://ieeexplore.ieee.org/document/8663805/ (visited on 03/16/2024).

[50] Mouhib Ibtihal, El Ouadghiri Driss, and Naanani Hassan. "Homomorphic Encryption as a Service for Outsourced Images in Mobile Cloud Computing Environment:" ng. In: *International Journal of Cloud Applications and Computing* 7.2 (Apr. 2017), pp. 27–40. ISSN: 2156-1834, 2156-1826. DOI: 10.4018/IJCAC.2017040103. URL: https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/IJCAC.2017040103 (visited on 01/16/2024).

[51] Icedrive. *Icedrive*. URL: https://icedrive.net/encrypted-cloud-storage (visited on 06/12/2024).

[52] ISO. *ISO/IEC 27017:2015 Information technology — Security techniques — Code of practice for information security controls based on ISO/IEC 27002 for cloud services*. 2015.

[53] ISO. *ISO/IEC 20000-1:2018 Information technology — Service management-Part 1: Service management system requirements*. 2018.

[54] ISO. *ISO/IEC 20000-3:2019 Information technology — Service management-Part 3: Guidance on scope definition and applicability of ISO/IEC 20000-*. 2019.

[55] ISO. *BS EN ISO/IEC 19790:2020 Information Technology - Security Techniques - Security requirements for cryptogrpahic modules*. 2020.

[56] ISO. *ISO/IEC 27002:2022 Information security, cybersecurity and privacy protection — Information security controls*. 2022.

[57] ISO. *ISO/IEC 23001-7:2023 Part 7: Common encryption in ISO base media file format files*. Geneva, Aug. 2023. URL: https://www.iso.org/standard/84637.html (visited on 04/02/2024).

[58] ITU. *Overview of the Internet of things*. Recommendation. June 2012. URL: https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=y.2060 (visited on 03/17/2024).

[59] David Johnston and Richard Fant. *Designing to FIPS-140: A Guide for Engineers and Programmers*. en. Berkeley, CA: Apress, 2024. ISBN: 9798868801242 9798868801259. DOI: 10.1007/979-8-8688-0125-9. URL: https://link.springer.com/10.1007/979-8-8688-0125-9 (visited on 05/27/2024).

[60] Marc Joye and Gregory Neven. *Identity-based cryptography*. eng. OCLC: 1162045457. Amsterdam: IOS Press, 2008. ISBN: 978-1-282-07060-8.

[61] Junzuo Lai et al. "Attribute-Based Encryption With Verifiable Outsourced Decryption". en. In: *IEEE Transactions on Information Forensics and Security* 8.8 (Aug. 2013), pp. 1343–1354. ISSN: 1556-6013, 1556-6021. DOI: 10.1109/TIFS.2013.2271848. URL: http://ieeexplore.ieee.org/document/6553162/ (visited on 03/09/2024).

[62] Seungmin Kang, Bharadwaj Veeravalli, and Khin Mi Mi Aung. "ESPRESSO: An Encryption as a Service for Cloud Storage Systems". en. In: *Monitoring and Securing Virtualized Networks and Services*. Ed. by Anna Sperotto et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2014, pp. 15–28. ISBN: 978-3-662-43862-6. DOI: 10.1007/978-3-662-43862-6_2.

[63] KeyOS. *KEYOS WIDEVINE EVERYWHERE*. URL: https://widevineeverywhere.com/.

[64] Geir M. Køien. "Why Cryptosystems Fail Revisited". en. In: *Wireless Personal Communications* 106.1 (May 2019), pp. 85–117. ISSN: 0929-6212, 1572-834X. DOI: 10.1007/s11277-019-06265-6. URL: http://link.springer.com/10.1007/s11277-019-06265-6 (visited on 01/28/2024).

[65] Krishna Ksheerabdhi. *Key Management as a Service (KMaaS) Explained*. June 2023. URL: https://cpl.thalesgroup.com/blog/encryption/key-management-as-a-service-guide (visited on 03/16/2024).

[66] A Lee. "Cryptographic Key Management (CKM) Design Principles for the Advanced Metering Infrastructure (AMI)". en. In: *Electric Power Research Institute* (Nov. 2012).

[67] Liran Lerman, Olivier Markowitch, and Jorge Nakahara Jr. "Key Management as a Service". en. In: Rome, July 2012, pp. 276–281.

[68] S. Lin et al. "Revisiting Attribute-Based Encryption With Verifiable Outsourced Decryption". en. In: *IEEE Transactions on Information Forensics and Security* 10.10 (Oct. 2015), pp. 2119–2130. ISSN: 1556-6013, 1556-6021. DOI: 10.1109/TIFS.2015.2449264. URL: http://ieeexplore.ieee.org/document/7131527/ (visited on 03/09/2024).

[69] Brian Lowans and Neil MacDonald. *Select the Right Key Management as a Service to Mitigate Data Security and Privacy Risks in the Cloud*. en. Tech. rep. Gartner Research, Feb. 2021.

[70] Hui Ma et al. "Verifiable and Exculpable Outsourced Attribute-Based Encryption for Access Control in Cloud Computing". In: *IEEE Transactions on Dependable and Secure Computing* 14.6 (Nov. 2017), pp. 679–692. ISSN: 1545-5971. DOI: 10.1109/TDSC.2015.2499755. URL: http://ieeexplore.ieee.org/document/7327175/ (visited on 01/16/2024).

[71] Martin R. Albrecht et al. "Share with Care: Breaking E2EE in Nextcloud". In: *IACR Cryptology ePrint Archive* Paper 2024/546 (2024). S2ID: d123fc71d7295b71bee589b0139ead9907
URL: https://eprint.iacr.org/2024/546.pdf (visited on 06/09/2024).

[72] Mega. *Cloud Storage Securely store, manage, and share your data online*. URL: https://mega.io/storage (visited on 04/19/2024).

[73] Peter Mell and Tim Grance. *The NIST Definition of Cloud Computing*. Tech. rep. NIST SP 800-145. NIST, Sept. 2011. URL: https://doi.org/10.6028/NIST.SP.800-145 (visited on 03/16/2024).

[74] L Micheloud. "Securing Cloud Storage with OpenPGP: An Analysis of Proton Drive". en. PhD thesis. ETH Zurich. URL: https://ethz.ch/content/dam/ethz/special-interest/infk/inst-infsec/appliedcrypto/education/theses/lea-micheloud-master-thesis.pdf (visited on 06/09/2024).

[75] Microsoft. *Azure Storage encryption for data at rest*. Feb. 2023. URL: https://learn.microsoft.com/en-us/azure/storage/common/storage-service-encryption (visited on 04/12/2024).

[76] Microsoft. *Microsoft Azure Attestation*. URL: https://azure.microsoft.com/en-us/products/azure-attestation/#pricing (visited on 04/27/2024).

[77] Microsoft. *Microsoft PlayReady Enhanced Content Protection*. URL: https://www.microsoft.com/playready/features/EnhancedContentProtection/ (visited on 04/15/2024).

[78] Elissa Mollakuqe et al. "Comparison of cloud storage in terms of privacy and personal data - Sync, pCloud, IceDrive and Egnyte". en. In: *Open Research Europe* 4 (June 2024), p. 128. ISSN: 2732-5121. DOI: 10.12688/openreseurope.16631.1. URL: https://open-research-europe.ec.europa.eu/articles/4-128/v1 (visited on 08/01/2024).

[79] Antonio Muñoz et al. "A survey on the (in)security of trusted execution environments". en. In: *Computers & Security* 129 (June 2023), p. 103180. ISSN: 01674048. DOI: 10.1016/j.cose.2023.103180. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167404823000901 (visited on 07/06/2024).

[80] Derek L. Nazareth, Jae Choi, and Thomas L. Ngo-Ye. "The Security-as-a-Service Market for Small and Medium Enterprises". en. In: *Journal of Computer Information Systems* 62.5 (Sept. 2022), pp. 954–964. ISSN: 0887-4417, 2380-2057. DOI: 10.1080/08874417.2021.1954563. URL: https://www.tandfonline.com/doi/full/10.1080/08874417.2021.1954563 (visited on 02/20/2024).

[81] Kenny Niehage. "Cryptographic Vulnerabilities and Other Shortcomings of the Nextcloud Server Side Encryption as implemented by the Default Encryption Module". en. In: *Cryptology e-Print archive* Paper 2020/14391 (2020). URL: https://eprint.iacr.org/2020/1439.pdf (visited on 06/09/2024).

[82] NIST. *Cryptographic Algorithm Validation Program CAVP*. Mar. 2023. URL: https://csrc.nist.gov/Projects/Cryptographic-Algorithm-Validation-Program (visited on 05/18/2024).

[83] Talal H. Noor et al. "Trust management of services in cloud environments: Obstacles and solutions". en. In: *ACM Computing Surveys* 46.1 (Oct. 2013), pp. 1–30. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/2522968.2522980. URL: https://dl.acm.org/doi/10.1145/2522968.2522980 (visited on 03/22/2024).

[84] OASIS. *Key Management Interoperability Protocol (KMIP) White Paper*. White Paper. OASIS Open, May 2009. URL: https://xml.coverpages.org/KMIP/KMIP-WhitePaper.pdf.

[85] OWASP. *A02:2021 – Cryptographic Failures*. OWASP Top 10 : 2021. URL: https://owasp.org/Top10/A02_2021-Cryptographic_Failures/ (visited on 02/03/2024).

[86] OWASP. *Secrets Management Cheat Sheet¶*. URL: https://cheatsheetseries.owasp.org/cheatsheets/Secrets_Management_Cheat_Sheet.html (visited on 04/25/2024).

[87] Dong Pei, Xu Guo, and Junxing Zhang. "A video encryption service based on cloud computing". In: *2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC)*. Macau, China: IEEE, July 2017, pp. 167–171. ISBN: 978-1-5090-3025-5. DOI: 10.1109/ICEIEC.2017.8076536. URL: http://ieeexplore.ieee.org/document/8076536/ (visited on 01/16/2024).

[88] Stefan Pham et al. "On the current state of interoperable content protection for internet video streaming". en. In: *2018 IEEE Seventh International Conference on Communications and Electronics (ICCE)*. Hue, Vietnam: IEEE, July 2018, pp. 13–17. ISBN: 978-1-5386-3678-7 978-1-5386-3679-4. DOI: 10.1109/CCE.2018.8465735. URL: https://ieeexplore.ieee.org/document/8465735/ (visited on 04/22/2024).

[89] Luis Francisco Plá, Narasimha Shashidhar, and Cihan Varol. "On-Premises Versus SECaaS Security Models". In: *2020 8th International Symposium on Digital Forensics and Security (ISDFS)*. June 2020, pp. 1–6. DOI: 10.1109/ISDFS49300.2020.9116453. URL: https://ieeexplore.ieee.org/abstract/document/9116453 (visited on 01/29/2024).

[90] Qosmos. *Qosmos (Entropy as a Service)*. URL: https://aws.amazon.com/marketplace/pp/prodview-fzkuj32ktapmq (visited on 04/02/2024).

[91] Sandro Rafaeli and David Hutchison. "A survey of key management for secure group communication". en. In: *ACM Computing Surveys* 35.3 (Sept. 2003), pp. 309–329. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/937503.937506. URL: https://dl.acm.org/doi/10.1145/937503.937506 (visited on 03/16/2024).

[92] Nick Rahimi, Jacob J. Reed, and Bidyut Gupta. "On the Significance of Cryptography as a Service". en. In: *Journal of Information Security* 9.4 (Aug. 2018). Number: 4 Publisher: Scientific Research Publishing, pp. 242–256. DOI: 10.4236/jis.2018.94017. URL: https://www.scirp.org/journal/paperinformation.aspx?paperid=87093 (visited on 01/14/2024).

[93] Hossein Rahmani et al. "Encryption as a Service (EaaS) as a Solution for Cryptography in Cloud". In: *Procedia Technology*. 4th International Conference on Electrical Engineering and Informatics, ICEEI 2013 11 (Jan. 2013), pp. 1202–1210. ISSN: 2212-0173. DOI: 10.1016/j.protcy.2013.12.314. URL: https:

//www.sciencedirect.com/science/article/pii/S2212017313004684 (visited on 01/15/2024).

[94] Subhabrata Rana et al. “A comprehensive survey of cryptography key management systems”. en. In: *Journal of Information Security and Applications* 78 (Nov. 2023), p. 103607. ISSN: 22142126. DOI: 10.1016/j.jisa.2023.103607. URL: https://linkinghub.elsevier.com/retrieve/pii/S2214212623001916 (visited on 03/16/2024).

[95] Peter Robinson. *Applying Cryptogrpahy as a Service to Mobile Applications*. San Francisco, Feb. 2014. URL: https://tailieu.antoanthongtin.vn/files/files/site-2/20170814/62-applying-cryptography-as-a-service-to-mobile-applications_final-14082017082253.pdf.

[96] Amit Sahai and Brent Waters. “Fuzzy identity-based encryption”. In: *Advances in Cryptology–EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005. Proceedings 24*. Springer. 2005, pp. 457–473.

[97] Bruce Schneier. “Why cryptography is harder than it looks”. In: *EDI FORUM-OAK PARK-*. Vol. 10. THE EDI GROUP, LTD., 1997, pp. 87–90. ISBN: 1048-3047.

[98] Bruce Schneier. “Security pitfalls in cryptography”. In: *EDI FORUM-OAK PARK-*. Vol. 11. THE EDI GROUP, LTD., 1998, pp. 65–69. ISBN: 1048-3047.

[99] Deepak H. Sharma, C A. Dhote, and Manish M. Potey. “Intelligent Transparent Encryption-Decryption as Security-as-a-Service from clouds”. en. In: *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*. Bengaluru, India: IEEE, Oct. 2016, pp. 359–362. ISBN: 978-1-5090-1020-2 978-1-5090-1022-6. DOI: 10.1109/CSITSS.2016.7779386. URL: http://ieeexplore.ieee.org/document/7779386/ (visited on 02/03/2024).

[100] S. Skeirik, R. B. Bobba, and J. Meseguer. “Formal Analysis of Fault-tolerant Group Key Management Using ZooKeeper”. en. In: *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*. Delft: IEEE, May 2013, pp. 636–641. ISBN: 978-0-7695-4996-5 978-1-4673-6465-2. DOI: 10.1109/CCGrid.2013.98. URL: http://ieeexplore.ieee.org/document/6546150/ (visited on 03/16/2024).

[101] Murugiah Souppaya. *Automation of the Cryptographic Module Validation Program (CMVP)*. en. Tech. rep. NIST SP 1800-40 ipd. Gaithersburg, MD: National Institute of Standards and Technology, 2023, NIST SP 1800–40 ipd. DOI: 10.

6028/NIST.SP.1800-40.ipd. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1800-40.ipd.pdf (visited on 05/18/2024).

[102] Colin Tankard. "The promise of managed security services". en. In: *Network Security* 2012.9 (Sept. 2012), pp. 10–15. ISSN: 13534858. DOI: 10.1016/S1353-4858(12)70082-X. URL: https://linkinghub.elsevier.com/retrieve/pii/S135348581270082X (visited on 02/10/2024).

[103] TCG. *TPM Certification*. URL: https://trustedcomputinggroup.org/membership/certification/ (visited on 05/27/2005).

[104] TCG. *What is a Trusted Platform Module (TPM)?* URL: https://trustedcomputinggroup.org/about/what-is-a-trusted-platform-module-tpm/ (visited on 04/15/2024).

[105] *The safest cloud storage for your files, photos, and more*. URL: https://internxt.com/drive (visited on 07/01/2024).

[106] Andrew Tomlinson, Simon Parkin, and Siraj Ahmed Shaikh. "Drivers and barriers for secure hardware adoption across ecosystem stakeholders". en. In: *Journal of Cybersecurity* 8.1 (Jan. 2022), tyac009. ISSN: 2057-2085, 2057-2093. DOI: 10.1093/cybsec/tyac009. URL: https://academic.oup.com/cybersecurity/article/doi/10.1093/cybsec/tyac009/6656148 (visited on 06/02/2024).

[107] TorGuard. *Why TorGuard's Network is Secure After an Isolated 2017 Server Breach*. Oct. 2019. URL: https://blog.torguard.net/why-torguards-network-is-secure-after-an-isolated-2017-server-breach/ (visited on 02/10/2024).

[108] Tresorit. *Tresorit Encryption Whitepaper*. Tech. rep. URL: https://cdn.tresorit.com/202208011608/tresorit-encryption-whitepaper.pdf (visited on 07/15/2024).

[109] Ikram Ullah, Nirvana Meratnia, and Paul J. M. Havinga. "Entropy as a Service: A Lightweight Random Number Generator for Decentralized IoT Applications". In: *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. Austin, TX, USA: IEEE, Mar. 2020, pp. 1–6. ISBN: 978-1-72814-716-1. DOI: 10.1109/PerComWorkshops48775.2020.9156205. URL: https://ieeexplore.ieee.org/document/9156205/ (visited on 04/23/2024).

[110] Vijay Varadharajan and Udaya Tupakula. "Security as a Service Model for Cloud Environment". en. In: *IEEE Transactions on Network and Service Management* 11.1 (Mar. 2014), pp. 60–75. ISSN: 1932-4537. DOI: 10.1109/TNSM.2014.041614.120394. URL: http://ieeexplore.ieee.org/document/6805344/ (visited on 02/03/2024).

[111] Apostol Vassilev and Robert Staples. “Entropy as a Service: Unlocking Cryptography’s Full Potential”. In: *Computer* 49.9 (Sept. 2016), pp. 98–102. ISSN: 0018-9162. DOI: 10.1109/MC.2016.275. URL: http://ieeexplore.ieee.org/document/7562338/ (visited on 03/04/2024).

[112] VDOCipher. *VDOCipher pricing*. URL: https://www.vdocipher.com/site/pricing/.

[113] Jai Vijayan. *Lion Air the Latest to Get Tripped Up by Misconfigured AWS S3*. Sept. 2019. URL: https://www.darkreading.com/cyberattacks-data-breaches/lion-air-the-latest-to-get-tripped-up-by-misconfigured-aws-s3 (visited on 04/27/2024).

[114] Patrick Walsh. *The Trouble with FIPS*. Hackernoon. Mar. 2022. URL: https://hackernoon.com/the-trouble-with-fips (visited on 05/25/2024).

[115] Wenyuan Wang and Sira Yongchareon. “Security-as-a-service: a literature review”. en. In: *International Journal of Web Information Systems* 16.5 (Sept. 2020), pp. 493–517. ISSN: 1744-0084, 1744-0084. DOI: 10.1108/IJWIS-06-2020-0031. URL: https://www.emerald.com/insight/content/doi/10.1108/IJWIS-06-2020-0031/full/html (visited on 01/19/2024).

[116] Pang Xiong Wen and Li Dong. “Quality Model for Evaluating SaaS Service”. en. In: *2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies*. Xi’an, Shaanxi, China: IEEE, Sept. 2013, pp. 83–87. ISBN: 978-1-4799-2141-6 978-1-4799-2140-9. DOI: 10.1109/EIDWT.2013.19. URL: http://ieeexplore.ieee.org/document/6631597/ (visited on 08/03/2024).

[117] *What Is Advanced Metering Infrastructure? | IBM — ibm.com*. https://www.ibm.com/topics/advanced-metering-infrastructure. [Accessed 23-03-2024].

[118] Alma Whitten and J D Tygar. “A Usability Evaluation of PGP 5.0”. en. In: (1999).

[119] Duane C Wilson and Giuseppe Ateniese. ““To Share or not to Share” in Client-Side Encrypted Clouds”. In: *Information Security: 17th International Conference, ISC 2014, Hong Kong, China, October 12-14, 2014. Proceedings 17*. Springer. 2014, pp. 401–412.

[120] Christopher A. Wood and Ersin Uzun. “Flexible end-to-end content security in CCN”. en. In: *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*. Las Vegas, NV: IEEE, Jan. 2014, pp. 858–865. ISBN: 978-1-4799-2355-7. DOI: 10.1109/CCNC.2014.6940528. URL: https://ieeexplore.ieee.org/document/6940528 (visited on 04/22/2024).

[121] Jasper van Woudenberg and Colin O'Flynn. *The Hardware Hacking Handbook: Breaking Embedded Security with Hardware Attacks*. en. No Starch Press, 2022. ISBN: 978-1-59327-875-5.

[122] Jin Wu, Zhiqiang Zhu, and Songhui Guo. "A Quality Model for Evaluating Encryption-as-a-Service". en. In: *Security, Privacy, and Anonymity in Computation, Communication, and Storage*. Ed. by Guojun Wang et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, pp. 557–569. ISBN: 978-3-319-72395-2. DOI: 10.1007/978-3-319-72395-2_51.

[123] Hu Xiong and Jianfei Sun. "Comments on "Verifiable and Exculpable Outsourced Attribute-Based Encryption for Access Control in Cloud Computing"". en. In: *IEEE Transactions on Dependable and Secure Computing* 14.4 (July 2017), pp. 461–462. ISSN: 1545-5971. DOI: 10.1109/TDSC.2017.2710119. URL: http://ieeexplore.ieee.org/document/7971997/ (visited on 01/28/2024).

[124] Dongyang Xu et al. "Fine-grained document sharing using attribute-based encryption in cloud servers". In: *Third International Conference on Innovative Computing Technology (INTECH 2013)*. Aug. 2013, pp. 65–70. DOI: 10.1109/INTECH.2013.6653703. URL: https://ieeexplore.ieee.org/document/6653703 (visited on 01/16/2024).

[125] Weijia Xue, Congli Wang, and Jinhua Wang. "Research on Cryptography as a Service Technique Based on Commercial Cryptography". en. In: *2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI)*. Changchun, China: IEEE, May 2022, pp. 260–264. ISBN: 978-1-72818-115-8. DOI: 10.1109/ICETCI55101.2022.9832226. URL: https://ieeexplore.ieee.org/document/9832226/ (visited on 03/01/2024).

[126] Hua Zhang et al. "An Adaptive Encryption-as-a-Service Architecture Based on Fog Computing for Real-Time Substation Communications". en. In: *IEEE Transactions on Industrial Informatics* 16.1 (Jan. 2020), pp. 658–668. ISSN: 1551-3203, 1941-0050. DOI: 10.1109/TII.2019.2948113. URL: https://ieeexplore.ieee.org/document/8873681/ (visited on 01/19/2024).

[127] Karim Zkik et al. "A New Authentication and Homomorphic Encryption as a Service Model for Preserving Privacy in Clouds". en. In: *Journal of Computer Science* 13.12 (Dec. 2017), pp. 702–717. ISSN: 1549-3636. DOI: 10.3844/jcssp.2017.702.717. URL: http://thescipub.com/abstract/10.3844/jcssp.2017.702.717 (visited on 03/01/2024).

Appendices

Appendix A

Detailed Findings

ENCRYPTED CLOUD STORAGE

Provider	Standards	Trust	Quality	Other
Tresorit	Management - ISO 27001 PBKDF standards Uses auth encryption Crypto++ - no CMVP Bouncy Castle - FIPS 140-3	Security whitepaper External code audit Whistleblower protection policy Publishes Incident reports Data privacy statement Crypto++ and Bouncy Castle	SaaS Model - Basic Service status dashboard Service 'as is' and 'as available'	Client Side Encryption Claims zero knowledge
iDrive	Some mention of using ISO 27001 data centres however the supplier does not actually hold ISO 27001 certification.	User held priv key held on desktop client Privacy policy commitment to privacy SOC 2 certification Source code available (not signed)	SaaS Model - Basic 99.9% availability	Client Side Encryption Doesn't claim zero knowledge Very poor white paper
Mega	Management - none NIST ciphers	Vulnerability reward programme Security white paper Source code available (signed)	SaaS Qual - Basic 99.9% availability	Client side encryption Claims zero knowledge White paper out of date
Sync	Not known	SOC 3 report available. Committed privacy policy	SaaS Qual - Basic	Security whitepaper withdrawn Also assesed by Mollakuqe et al.
Pcloud	Management - ISO 27001:13	Time limited hacking challenge	SaaSQual - Basic	Also assesed by Mollakuqe et al. Client side encryption Claims zero knowledge
Icedrive	Management - none Uses Twofish cipher	None	SaaS Qual - Basic	Also assesed by Mollakuqe et al. Claims zero knowledge
Nextcloud	Management - none Claims security mgmt system. NIST ciphers	Source code available (signed) Code of ethics Privacy policy Security whitepaper External audit (5 years old)	SaaS Qual - Basic to Integrated Depends upon price point.	Server side encryption White paper out of date
Internxt	Truncated language NIST ciphers	Source code available (signed) External audits removed from site	SaaS Qual - Basic 99.99% availability	
Proton Drive	Management - none	Source code available (signed) Bug Bounty Security audit Security white paper	SaaS Qual - Standard	Offers encrypted file search Offers zero knowledge encryption
Nordlocker	Not known	Privacy policy	SaaS Qual - sub Basic (security)	ToS for personal accounts ToS for business accounts
Filen	Mention of using ISO 27001 certified data centres however does not hold ISO 27001. References NIST standardised ciphers etc.	Source code available (signed) Bug Bounty Security white paper	SaaS Qual - sub Basic (security)	

Other major providers - AWS, Azure, Google Cloud, AliCloud, Apple Drive

SECRETS MANAGEMENT AND ENCRYPTION ENGINES

Provider	Standards	Trust	Quality	Other
a Keyless	Management - none Own FIPS 140-2 crypto module	BYOK is available option. SOC 2 Balsa - Bug Bounty, Pen Tests	SaaS Qual - Basic	Patent for key frag tech Claims zero knowledge
Hashicorps	ISO 27001 FIPS 140-2 cert	SOC 2 Balsa - Source Code Signed Code Some design docs	SaaS Qual - Basic to Integrated Depends upon price point.	Training certs
Bitwarden	Management - None Text on website: ”based on ISO 27001 SMS” but no certification.	SOC 2 Source code (signed) Design documentation Code audit and pen tests Bug Bounty Multiple audit reports	SaaS Qual - Basic to Integrated Depends upon price point.	Claims zero knowledge
Keeper Security	Management - ISO 27001, ISO 27017 FIPS 140-2 certified (about to expire)	SOC 2 Bug Bounty Program Regular pen tests Design documentation	SaaS Qual - Basic to Integrated Depends upon price point.	
Evervault	Management - none	SOC 2 Design documentation	SaaS Qual - Basic.	Aimed at the payment security market. Claims zero knowledge

DRM AS A SERVICE

Provider	Standards	Trust	Quality	Other
Evercast	Management - Motion Picture Association Trusted Partner Network	None	SaaS Qual - Basic.	
VDO Cipher	Management - none	None	SaaS Qual - didn't achieve basic (security) SLA available for bandwidth, bitrates Some customisation Email support only.	
EZDRM	Management - none	None	SaaS Qual - didn't achieve basic (security)	On prem option
Buy DRM	Management - 27001, 27017	SOC 3	SaaS Qual - Basic.	Owned by OVH Cloud 27001, 27017, SOC 1-3, CSA Star 1
PallyCon	Management - none	None	SaaS Qual - didn't achieve basic (security)	Platform?
Viaccess.Orca	Management - none	None	SaaS Qual - Basic to Integrated Depends upon price point.	Farncombe Security Audit (Industry level cert)
castlabs	Management - none	None	SaaS Qual - Basic.	Farncombe Security Audit (Industry level cert)

TRUSTED EXECUTION AND REMOTE ATTESTATION

Provider	Standards	Trust	Quality	Other
Evervault	Follows IETF Built on AWS solution, inherits some of their standards.	SOC 2 report available on request. Attestation - must trust provider. Relying Party can't fully own TA store. Balsa - some designs.	SaaS Qual - Basic.	TEE + Attestation Key lifecycle in attestation unclear, not controlled by customer. in TOS "we will do everything reasonable but we're not certified."
AWS Nitro Enclave	ISO 27001 Follows IETF ISO 20000	SOC 3 report published. VP is the Amazon CA which can be trusted by RP. Can use customer KMS, customer can control TA store. Balsa - security white paper.	SaaS Qual - Basic to Integrated Depends upon price point.	TEE + Attestation
Microsoft	ISO 27001 Follows IETF ISO 20000	SOC 3 report published. VP is the Microsoft CA which can be trusted by RP. Can use customer KMS, customer can control TA store. Balsa - some designs.	SaaS Qual - Basic to Integrated Depends upon price point.	Attestation Service only
Intel Trust Authority	ISO 27001 Follows IETF ISO 20000	SOC 3 report published. VP is their own CA which can be trusted by RP. Customer can use own KMS Customer can control TA store. Balsa - some designs.	SaaS Qual - Basic to Integrated Depends upon price point.	Attestation Service only
Fortanix Confidential Compute	ISO 27001 Does not follow IETF Must assume trust between Attester and VP.	SOC 2 Client can't control TA store.	SaaS Qual - Elms of Basic + Standard	TEE + Attestation
Google Confidential VM	ISO 27001 Attestation arch unclear	SOC 3 report published. Client can use 3rd party attest service.	SaaS Qual - Basic to Optimised Depends upon price point.	TEE

ENTROPY AS A SERVICE

Provider	Standards	Trust	Quality	Other
Qunitessence Labs	NIST sp 800-90A 90B and 90C	None	SaaS Qual - didn’t achieve basic (security)	
Quantum Blockchains	None	None	SaaS Qual - didn’t achieve basic (security)	Makes claims that cannot be validated (four entropy providers)
Qosmos EntaaS (Qnulabs)	None	None	SaaS Qual - didn’t achieve basic (security)	
Quantum eMotion	None	None	SaaS Qual - didn’t achieve basic (security)	
Quantropi	None	None	Benchmarked by third party.	
Qrypt	Management - ISO 27001	None	SaaS Qual - Basic Only	

KEY MANAGEMENT AS A SERVICE

Provider	Standards	Trust	Quality	Other
IBM Cloud	Use of HSM so FIPS 140 HSM PKCS#11 API KMIP Not all stds referenced, eg key wrap is referenced but not key gen,	SOC 3 MK controlled by user but kept on supplier HW Support user created keys Bug bounty	Multi cloud solution Several dependiencies on other IBM services inc IAM soln SaaS Qual - From Basic to Integrated	Offers Dilitium KEM
Thales CipherTrust	ISO 27001 FIPS 140-2 Management - none	SOC 2 Can hold Master Key on prem Support user created keys	SaaS Qual - Standard 99.95% availability Multi cloud solution	Includes HSMs as part of service
StorMagic	KMIP, PKCS#11	Very little found	SaaS Qual - Basic to Integrated Depends upon price point.	Can also manage HSM (not included in service) Arguably not KMaaS requires a high degree of customer knowledge.
Fortanix	Claims FIPS 140-2 but cert cannot be found. Management - ISO 27001 KMIP, PKCS#11	SOC 2	SaaS Qual - Elements of Basic and Standard	Deployed on a TEE but no details given of attestation.
Cogito Group	Management - ISO 27001 No other stds mentioned	Very little found	SaaS Qual - Basic Only	
OVHcloud	Management - ISO 27001 FIPS 140-3, KMIP	BYOK	SaaS Qual - Basic to Integrated Depends upon price point.	