computers

MDPI

Article

Security Pattern for Cloud SaaS: From System and Data Security to Privacy Case Study in AWS and Azure †

Annanda Rath *, Bojan Spasic, Nick Boucart and Philippe Thiran

Software Engineering Department, Sirris, BE-1030 Brussels, Belgium; Bojan.Spasic@sirris.be (B.S.); Nick.Boucart@sirris.be (N.B.); Philippe.Thiran@sirris.be (P.T.)

* Correspondence: annanda.rath@sirris.be

† This paper is an extended version of our paper published in IEEE CloudTech 2018; Security Pattern for Cloud SaaS: from system and data security to privacy—Annanda Rath, Philippe Thiran, Bojan Spasic and Nick Boucart.

Received: 29 March 2019; Accepted: 28 April 2019; Published: 3 May 2019

Abstract: The Cloud is fast becoming a popular platform for SaaS, a popular software delivery model. This is because the Cloud has many advantages over the traditional private infrastructure, such as increased flexibility, no maintenance, less management burden, easy access and easy to share information. However, there are many concerns around issues like system security, communication security, data security, privacy, latency and availability. In addition, when designing and developing Cloud SaaS application, these security issues need to be addressed in order to ensure regulatory compliance, security and trusted environment for Cloud SaaS users. In this paper, we explore the security patterns for Cloud SaaS. We work on the patterns covering different security aspects from system and data security to privacy. Our goal is to produce the security best practices and security knowledge documentation that SaaS developer can use as a guideline for developing Cloud SaaS applications from the ground up. In addition to that, we also provide a case study of security patterns and solutions in AWS and Azure.

Keywords: cloud; security patterns; SaaS; software security; AWS; Azure; security patterns definition methodology

1. Introduction

The concept of patterns as way of capturing and expressing tried-and-true solutions to recurring problems was first formalised by Christopher Alexander in his book about urban planning and architecture [1], in which he attempted to explain and classify common recurring problems in designing and building physical structures, using a specific, structured language. A decade and a half later, the concept was successfully applied to software engineering by Gamma, Helm, Johnson and Vlissides (also known as the Gang of Four, or simply GoF) in their influential "Design Patterns: Elements of Reusable Object-Oriented Software" [2], which was followed by "Pattern-Oriented Software Architecture" (known as POSA) by Buschmann et al. [3]. These seminal software engineering pattern works, along with the original inspiration of the Alexandrian pattern concept, laid the foundations for pattern semantics and structure. Building upon their success, the pattern community has grown and expanded into the fields of software security and security engineering. In software security, Cloud SaaS (Software as a Service) security pattern is one of the most important areas being actively studied given its popularity and adoption rates.

In recent decades, the Cloud security pattern has been involving in response to ever-increasing number of security attacks, vulnerabilities and exploits. As such, the concept of security is quickly

shifting from an often-overlooked afterthought to a mandatory design requirement. In order to realise this requirement, the Cloud SaaS community is in overwhelming need for structured information about best security practices and security knowledge, to help them design and develop secure Cloud SaaS. So far, there is some research [4–14] on security patterns, and they tend to address a specific security area (e.g., OWASP [11,14] web development security guideline where the focus is only on how to develop a secure code and how to prevent cyber attacks) and do not cover many important aspects, such as privacy and governance. Moreover, there is a lack of structured and united information about security best practices and security knowledge that SaaS developer can use as a guideline for developing Cloud SaaS application from the ground up.

In this paper, we aim at providing a complete list of security patterns applied to Cloud SaaS application. The patterns cover four important areas of Cloud security including system security, communication security, data security and privacy. Furthermore, based on the security patterns we defined, we produce a security best practices and security knowledge guideline [15] for Cloud SaaS developer. In addition to that, we look at the security solutions in Amazon Web Service (AWS) and Azure and map our defined patterns to the solutions offered by both Cloud services providers. It is worth noting that this paper is an extended version of the paper published in the IEEE CloudTech 2018 [16]. The extension focuses on security pattern solutions and detailed discussion on solutions in AWS and Azure, which are absent in the published paper [16].

The paper is organised as follows. Section 2 is about our proposed methodology used to define the Cloud security patterns. Section 3 presents the security patterns in Cloud SaaS. We provide a high level classification of the security pattern and a complete list of patterns in each class/category. Section 4 presents the pattern expression and structure. The solutions to each security pattern identified in Section 3 are also presented in this section. Section 5 is about a case study of AWS and Azure. Section 6 presents related work and we conclude this paper in Section 7.

2. Cloud Security Pattern Definition Methodology

In this section, we present our proposed methodology used to define and classify the security patterns in Cloud SaaS. As shown in Figure 1, we divide the whole process into five steps: from security requirements identification to security pattern classification in step 5. In each step, existing guidelines may be used. For example, security and risk assessment is conducted based on OWASP [11,14] guideline.

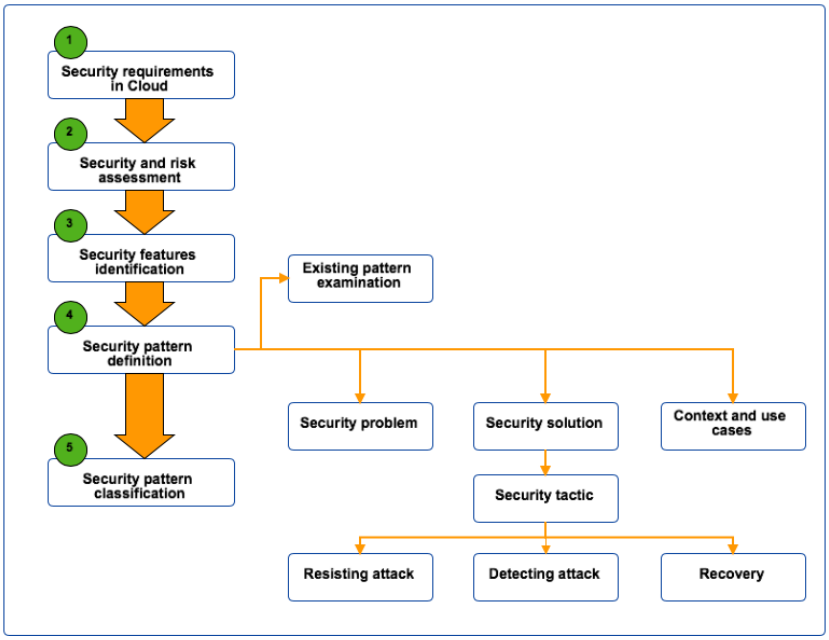

Figure 1. Cloud security pattern definition process.

2.1. Security Requirements in Cloud SaaS

The first step of the process focuses on defining the security requirements in Cloud SaaS. We study all the possible security requirements covering different aspects from data and system security to communication security and privacy. Our ultimate goal is to list all the security requirements needed for building the secure, trust and legal compliance Cloud SaaS application. Concerning legal requirements, we analyse different data processing requirements (e.g., the role and responsibility of data processor or personal data usage control) under the scope of GDPR [17]. The outcome of this step is a general Cloud security checklist (see Table 1) for SaaS application. This high-level security requirements are then used for assessing security and risk in Cloud SaaS.

Table 1. Top-Level Cloud Security Requirements.

ID	Requirement Description
R1	must provide protection to system's components
R2	must be able to prevent unauthorised access and intrusion to system and resources
R3	must be able to monitor network requests
R4	must have auditing option and be able to recover from a breach
R5	must ensure data protection at rest and in transit
R6	must ensure privacy protection and regulatory compliance
R7	must provide secure communication between modules
R8	must provide protection to system's resources

2.2. Security and Risk Assessment

Security and risk assessment is an explicit study to identify security vulnerabilities and risks in Cloud SaaS. The main goal is to study the required security and identify improvements to secure the systems and to ensure that necessary security controls are integrated into the design and implementation of a Cloud SaaS project. The outcome of this task is a properly completed security assessment documentation outlining any security risk that might have. This security and risk assessment report is then used for analysing and extracting the security features in Cloud SaaS.

2.3. Security Features Identification

Security feature refers to a specific security protection against attack. There are different kinds of attacks designed to target system, resources or user. Thus, different security features are required in order to be able to cope with these attacks.

Since Cloud applications and services are delivered through the Internet, Cloud computing faces various kinds of external security risks, such as denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks. In addition, and particularly in the public Cloud where data is hosted by the Cloud Service Provider (CPS), trust, confidentiality, and privacy are also important issues. To identify the security features, we take the security and risk assessment defined in the second step as the inputs. The by-product is a complete list of required security features.

2.4. Security Pattern Definition

The definition of security pattern in Cloud SaaS is based on the required security features defined in step 3. We also examine the existing security patterns in the traditional systems. The idea is to find out whether or not the patterns in other system can be used in Cloud SaaS environment. We look at three important security areas for pattern definition: (1) communication security; (2) data security and (3) system security.

For each defined pattern, they have the following structure.

- Security problem. Define the potential problem and their consequence if there is no security implementation to address it.

- Context and use cases. Identity in which context and use cases that this security issue may happen.
- Security solution. This provides solution to address the security issue. Our proposed security tactic principles are.
  - Resisting attack. This is the prevention technique where the designed solution should resist to any attack that might happen during application service. For example, a solution for an application to resist an unauthorised access is to have a strong access control model and system (e.g., multi-factor authentication).
  - Detecting attack. This technique allows system to detect attacks and react early to minimise the risk and serious consequence as the result of the attack.
  - Recovery. In case the system can not prevent attack, the fast recovery solution should be in place in order to minimise the disruption of service.

2.5. Security Pattern Classification

After defining the security patterns, we arrange and classify them into different categories depending on their problem domain. For example, communication security pattern, data security pattern or system security pattern. In general, the patterns are grouped together if they are related in problem and context. We also define the relationship of crossed categories patterns.

3. Security Patterns in Cloud SaaS

We use the pattern definition methodology presented in Section 2 to define the Cloud SaaS security pattern. We start with top-level security requirements presented in Table 1. Below are the short description of each requirement.

1. **R1: must provide protection to system's components.** This requirement concerns the protection of system's components both the softwares (e.g., piece of code) and hardwares (e.g., sensor devices) that are parts of system.
2. **R2: must be able to prevent unauthorised access and intrusion to system and resources.** This requirement is about assuring that only genius user or application can access to application or system's resources.
3. **R3: must be able to monitor network requests.** The main goal is to monitor network requests in order to prevent potential attacks to system and its resources.
4. **R4: must have auditing option and be able to recover from a breach.** This requirement concerns the auditing of system and resources usage to find out the anomaly.
5. **R5: must ensure data protection at rest and in transit.** This requirement concentrates on how to protect data both in transit and at rest, especially when they are in public Cloud platform.
6. **R6: must ensure privacy protection and regulatory compliance.** This requirement is about how to ensure privacy protection and regulatory compliance of data processed in the Cloud infrastructure.
7. **R7: must provide secure communication between modules.** A system may be made of different modules deployed in the same or different Cloud platforms. Thus, it is important to ensure a secure communication between those modules.
8. **R8: must provide protection to system's resources.** The system's resources here refer to the Cloud resources required to run Cloud application. How to protect Cloud's resources from excessive and unnecessary use in order to ensure economic durability and durable availability of application running on the Cloud platform.

Based on our thorough Cloud SaaS security study with above security requirements, we classify the Cloud SaaS security patterns into five categories (see Figure 2). Below are the detailed explanation of the five security pattern categories we identified.

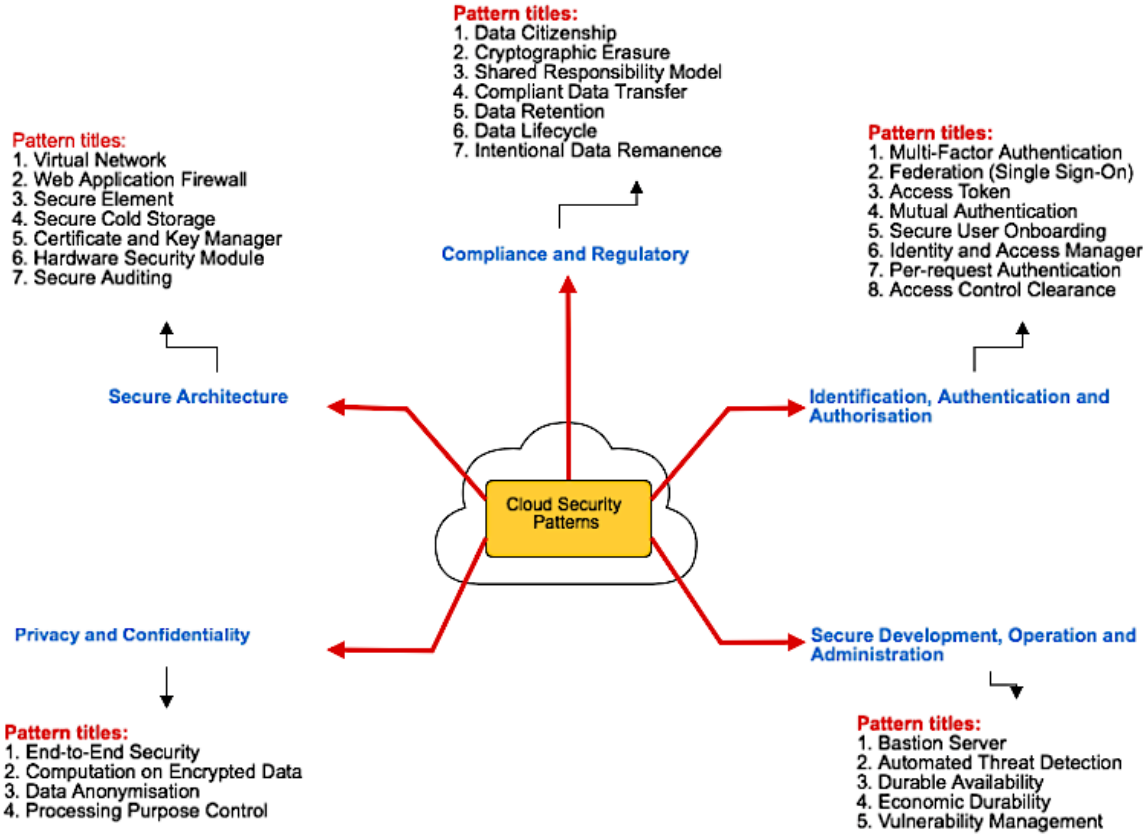

Figure 2. Cloud SaaS security patterns.

1. **Compliance and Regulatory**. Different countries have different laws governing the processing and usage of data; hence, proper control and handling of data are important to ensure regulatory compliance in data usage if we want to make data accessible for different geographical regions (e.g., countries). This category covers mostly the security patterns for data security.

    - **Data Citizenship**. Storing data gathered from users in Cloud SaaS is subjected to legal issue and improper handling of data can have serious legal consequence. The Cloud SaaS application developer must find a Cloud-based solution achieving regulatory compliance with respect to data storage locality.
    - **Cryptographic Erasure**. Right to be forgotten is one of many requirements in GPDR [17], finding a way to erase data reliably and securely after it was stored in the Cloud is important in order to maintain the regulatory compliance requirement.
    - **Shared Responsibility Model**. Who is responsible for in case of data lost, alteration or any other kind of data mismanagement? How can a Cloud services consumer effectively manage their Cloud application legal and regulatory compliance?
    - **Compliant Data Transfer**. Data in Cloud may be transferred across applications in the same platform or different platforms. Thus, data processing may be subjected to different jurisdictions. Ensuring the safety and regulation compliance of transferred data processed at other parties is important.
    - **Data Retention**. Processing of personal data is generally subjected to tighter control than normal data and the retention of such data is regulated by law. The data retention policy is different from countries to countries and the implementation of such policy in Cloud SaaS application must be adapted accordingly.

- **Data Lifecycle** refers to a complete cycle of data usage. It starts from when data is produced or created untill it is destroyed or erased. Managing data lifecycle effectively in the Cloud environment is important, especially private data.
- **Intentional Data Remanence**. Accidental or malicious deletion of data is one of the important data security issues. How to recover data when it is accidentally erased and how to prevent malicious user from erasing data are important issues to be addressed.

2. **Identification, Authentication and Authorisation**. This category focuses on the security patterns for user and resources management. It covers all the patterns in authentication and authorisation for system resources management and user access control.

- **Multi-Factor Authentication**. With the advance of cryptography algorithm and computing power, the traditional method of using only username and password is no longer secure. In the Cloud context, multi-factor authentication should be used to securely authenticate physical users of Cloud-based applications.
- **Federation (Single Sign-On)** provides a way to authenticate users without the burden of setting up and securely maintaining a user database.
- **Access Token**. In Cloud context, applications can exchange data in the same Cloud platform or different platforms. Making sure that data is from trusted application is important. Access token is generally used to control human or machine user access to Cloud APIs.
- **Mutual Authentication** is used to establish identity of parties in a Cloud communication channel.
- **Secure User Onboarding** relates to the security of the Cloud application users in initial registration phase.
- **Identity and Access Manager** relates to the management of user identity and access control in a Cloud application.
- **Per-request Authentication** is about the continuous proof of the identity of the user when they perform sensitive operations.
- **Access Control Clearance** relates to the enforcement of access and usage control policies for different types of authentication.

3. **Secure Development, Operation and Administration**. This category focuses on the general security for secure development, operation and administration of Cloud SaaS application.

- **Bastion Server**. Protecting data and application resources is important and bastion server is designed to address these issues. It allows data to be accessed without exposing them directly to the Internet.
- **Automated Threat Detection**. Automatic detection of threats (or attacks) in Cloud SaaS is important to ensure proper functioning and better experiences for users in the system.
- **Durable Availability**. Denial-of-service is the common attack in Cloud and maintaining availability of the Cloud services in the face of distributed denial-of-service attacks is important to ensure proper service for users.
- **Economic Durability**. Running application on Cloud is not for free, people hosting Cloud SaaS application in Cloud are charged based on the sources application uses including disk space and computing power. If attacker is able to make system to unnecessary running, they can drain the resources in the Cloud.
- **Vulnerability Management**. Early detection and response of found vulnerabilities are important in order to minimise risk that might cause severe damage to the system.

4. **Privacy and Confidentiality**. This category is about the data privacy, confidentiality and integrity.

- **End-to-End Security.** It is about ensuring that message communicated between two parties is protected across all components in the Cloud communication channel.
- **Computation on Encrypted Data**. It is all about data confidentiality and integrity assurance in Cloud service by allowing data to be processed without disclosing them.
- **Data Anonymisation**. This pattern is about how to securely and safely process personal data and at the same time maintaining regulatory compliance(e.g., GDPR [17]). How to remove personal identifiers from datasets to protect privacy, while keeping the datasets still valuable for processing?
- **Processing Purpose Control**. This pattern concerns on how to assure that personal data is processed in accordance with its intended purpose and data owner's consent.

5. **Secure Architecture** This category focuses on the secure architecture.

- **Virtual Network.** This pattern is about protecting the communication between end-device and Cloud SaaS application from unnecessarily exposing them to the Internet.
- **Web Application Firewall.** This pattern focuses on how to protect web API endpoints from unauthorised access and abuse.
- **Secure Element.** In IoT context, protecting the identity of devices connected to the Cloud is very important in order to ensure that only genius devices are allowed to connect to the Cloud. This pattern is all about the secure on-boarding of IoT device on Cloud.
- **Secure Cold Storage.** This pattern concerns about how to provide the availability of large amounts of data in a cost-effective way.
- **Certificate and Key Manager.** This pattern relates to cryptographic keys and certificate management for securing data at rest and in transit.
- **Hardware Security Module.** This pattern is about the protection of the cryptographic secrets owned by Cloud tenants while still enabling Cloud processing infrastructure to compute on the tenant data.
- **Secure Auditing.** This pattern is about providing the secure environment for auditing and reporting security-related behaviour in an operating Cloud system.

4. Pattern Expression and Structure: Security Patterns and Solutions

Security patterns are generally expressed as templates, following a certain structure. We structure the patterns, like in GoF [2], featuring sections, such as "Problem", "Context", "Solution", "Related Patterns", "Consequences", "Known Use", and "References". However, to keep it short, in this paper we present only "Problem", "Context", and "Solution". The detailed features of all the patterns can be found at [15].

- **Problem** is a statement relating to the security issue for a given pattern.
- **Context** is about the security context, in which context the security issue occurs.
- **Solution** is about the solutions to address the defined security pattern.

4.1. Compliance and Regulatory

In this category, there are seven patterns.

1. Data Citizenship

- **Problem**. How can a Cloud-based solution achieve regulatory compliance with respect to data storage locality?
- **Context**. Different legal and regulatory requirements and standards in different geographic regions might call for specific types of data to be physically stored in a designated country/legal jurisdiction. For example, EU GDPR differentiates between data storage and transfer within the EU boundaries and the export and storage of data outside of the EU.

- **Solution**. Cloud providers offer their services with location tags. When instantiating a service, the Cloud user can choose the geographic location, which is specified by a regional designation (e.g., EU-West). While Cloud providers usually do not advertise the exact physical location of their data centres, they do provide guarantees that a geographic location designation falls under a certain legal jurisdiction. Geographic designations, however, do not extend to cover all Cloud services; large Cloud environments remain at least partially location-agnostic, especially for the services that need to have dispersed infrastructure to ensure functionality, such as DNS or Web Application Firewalls.

2. Cryptographic Erasure.

- **Problem**. How can a dataset be reliably and securely erased after it was stored in the Cloud?
- **Context**. In Cloud environments, including those with endpoint devices deployed at large, data is often replicated and shared across a large number of physical devices, sometimes geographically dispersed. This makes guaranteed secure data deletion in the traditional sense difficult, if not impossible.
- **Solution** Encrypting the data at rest reduces the problem of managing entire data set deletion to the problem of managing cryptographic key lifetime. As cryptographic keys used for encryption at rest are small, they are far more manageable than potentially huge datasets, and can be kept in controlled storage (e.g., HSMs). Cryptographic deletion then amounts to secure destruction of the key data. Provided that the keys have not been compromised throughout their lifetime, and forward-secure cryptographic algorithms have been used, cryptographic deletion guarantees illegibility of the encrypted data set, up to the security guarantees provided by the encryption algorithm.

3. Shared Responsibility Model.

- **Problem**. How can a Cloud services consumer effectively manage their Cloud application legal and regulatory compliance?
- **Context**. One of the benefits of the Cloud offerings is the reduction of the total cost of ownership, as well as the liability and responsibility for the functioning of the Cloud infrastructure. Acquiring and managing own data centre to run applications bears a lot of hidden costs and regulatory and compliance requirements and risks, which can put a significant burden on an organisation.
- **Solution**. In the early days of initial Cloud service offerings, Cloud providers offered only infrastructure-as-a-service, i.e., computing facilities where Cloud tenants could install and run virtual machines. The natural responsibility for the uptime, reliability, availability and security of the infrastructure was on the Cloud provider, while the tenants assumed responsibility for the choice, installation, maintenance and running of the VMs and the applications installed on them. As the Cloud offerings grew into the more sophisticated platform-as-a-service domain, the responsibility for more and more functionality was assumed by the Cloud providers.

4. Compliant Data Transfer.

- **Problem**. How can data be transferred for processing to other parties in potentially different jurisdictions while staying in compliance with legal and regulatory requirements?
- **Context**. Modern SaaS applications are often composed of multiple APIs. For example, an online store may focus its own application logic on the specific product catalogue, but will potentially outsource standard functions, such as user sign-on, sign-in, email notifications, billing etc. to third party providers. This third-party functionality is often exposed through an API and is a part of the business offering in an API economy.

- **Solution**. Different laws and regulations have incorporated the concept of compliant data transfer in their body. For example, EU GDPR provides possibility for compliant transfer of data to third-country data processors using the contractual "model clauses".

5. Data Retention.

- **Problem**. How long is personal information retained? Which retention policy governs the data? Who enforces the retention policy in the Cloud?
- **Context**. Privacy laws in various countries place limitations on the ability of organisations to retain some types of personal information and each country has their own legal retention period. Thus, the control of the data retention needs to be adjustable to the jurisdiction under which the Cloud service is operating. Additionally, in Cloud scenarios, the governance of data storage (data citizenship) may be different to the governance of the data users; e.g., Asian customers using European data storage location.
- **Solution**. To avoid a potential legal violation, an automatic module (or tool) should be used to control the data retention period. When the legally allowed retention period elapses, the data should be permanently erased from storage. Automatising this process eases the management of data as well as application.

6. Data Lifecycle

- **Problem**. How to efficiently and securely manage data lifecycle in the Cloud?
- **Context**. Data lineage helps data lifecycle management by including metadata such as the data origin and where it moves over time, how is it accessed/modified and by whom. Data lineage increases visibility in the data analytics pipeline and simplifies tracing errors back to their sources. In the Cloud context, sharing data across applications is what is happening in the current connected world. However, with fake data everywhere, without proper trace and track the source of data and how the data was processed over time, data reliability is reduced.
- **Solution**. All data exchanged between different entities across applications (systems) must be attached with (1) access and usage control policy used to govern/control the access and usage of data and (2) the access and usage history that stores all access and usage information at any point in the data lifecycle.

7. Intentional Data Remanence.

- **Problem**. How can data in the Cloud be protected from accidental or malicious deletion?
- **Context**. Some types of data, as required by law, need to be retained and stored for a specific period of time for the purpose of investigation or research. For example, data generated by public visual camera need to be stored sometime for investigation purpose. In the Cloud context, data is separated logically, but not physically. If malicious user is able to get hand on data and intentionally destroy it, the application or Cloud platform provider is reliable to this. The data recovery plan designed for such event should be in place to address this data loss challenge.
- **Solution**. The solution is to design a system in such a way so that the representation of digital data remains even after attempts have been made to remove or erase it. This makes possible by using data replication/redundancy in the physically distributed Cloud system.

4.2. Identification, Authentication and Authorisation

There are 8 patterns in this category.

1. Multi-Factor Authentication

- **Problem**. How to simply, yet securely authenticate physical users of Cloud-based applications?

- **Context**. Authentication of humans by machines is a problem of balancing usability and security. The combination of the traditional three factors something the user knows (secret password), something the user has (physical possession) and something that is an unique trait of the user (biometrics) provide high level of security, each imposing a different burden on the part of the user. Passwords have been the main authentication factor through the history of computing and there is a large body of knowledge pointing out to the deficiencies in the treatment of passwords by users. Physical tokens are often used as a second authentication factor.
- **Solution**. While presenting all three authentication factors at the same time remains the most secure option, this level of security is often not required in the typical Cloud application scenarios. In order to mitigate the vulnerabilities associated with passwords, multi-factor systems often include authentication levels based on the access scenarios, sensitivity level and the associated risk of the operations that the user wishes to perform; e.g., in a banking application, a user may authenticate with a fingerprint on their mobile device (2 factors) to access their accounts for viewing, but a large money transfer may require an additional input of a password or pin.

2. Federation (Single Sign-On)

- **Problem**. How to authenticate users without the burden of setting up and securely maintaining a user database?
- **Context**. Management of the user identities is often a burdensome task and in SaaS applications it often consumes a lot of time if done right.
- **Solution**. Reusing existing user sign-in and sign-on features developed and maintained by third parties is an effective way to outsource authentication tasks to a third party. While the technical solutions employed by the third parties are often state-of-the-art, such outsourcing bears inevitable risks of privacy protection of the users, especially when the federated entities are social networks.

3. Access Token

- **Problem**. How to control human or machine user access to Cloud APIs?
- **Context**. Access to Cloud API endpoints often needs to be given on per-use or basis to achieve certain security levels. Using user credentials directly every time does not allow such levels of control.
- **Solution**. Access tokens are cryptographic secrets issued to API users, allowing them to programmatically access API endpoints. Access tokens enable fine-grained temporal and functional control, enabling access only to specific functions at the specified time. The lifetime of tokens is easily controlled, so their issuing and revocation can easily be automated.

4. Mutual Authentication

- **Problem**. How to establish identity of parties in a Cloud communication channel?
- **Context**. In a Cloud environment, multiple physical and logical components connect and exchange information. Without proper authentication between communicating parties, man-in-the-middle attacks are possible.
- **Solution**. Using two-way authentication to allow both entities in a communications link to authenticate each other.

5. Secure User Onboarding

    - **Problem**. How to securely perform initial registration of Cloud application users?
    - **Context**. When new device or user gains access to system for the first time, they need to be securely onboarding the system. If it is not handled properly, we place users, devices, data and the network at risk.
    - **Solution**. Define a secure onboarding process for new device or user who wants to access the system for the first time. The process must consist of at least the identity establishment and validation.

6. Identity and Access Manager

    - **Problem**. How to securely and effectively manage a user database and provide authentication and authorisation functionality in a Cloud application?
    - **Context**. In Cloud context, it is important to establish user identity and also control access to system's resources in order to protect system and resources from unauthorised or malicious users.
    - **Solution**. A proper tool should be used to define and manage the roles and access privileges of individual application users and the circumstances in which users are granted (or denied) those privileges.

7. Per-request Authentication

    - **Problem**. How to continuously prove the identity of the user when they perform sensitive operations?
    - **Context**. In current Cloud environment there is no continuous control over user activities once user is authenticated. If an attacker is able to hack an account, he can do whatever he wants on both user and system resources. Controlling user activities during usage session is important in some use cases (e.g., smart home, healthcare) in order to prevent or minimise the damage that might happen as the result of account hacking. With continuous control, system can react on time to the abnormal activities done or being done by user.
    - **Solution**. The solution is to develop the intelligent usage control tools monitoring the usage activities of user from the start till the end of usage session. The tools must work in the background and be intelligent enough to detect any abnormal activities and prevent user from making further damage if abnormal activities are detected.

8. Access Control Clearance

    - **Problem**. How to enforce access and usage control policies for different types of authentication?
    - **Context**. In general, the access and usage control of data are governed by policies in which it defines who can do what in which circumstance. However, ensuring user respects what is defined in policy is a challenge. For example, if policy states that user needs to notify data owner before accessing or using it, it is an obligation to ensure that notification message reaches data owner before data is released or made accessible to user.
    - **Solution**. The key solution to policy enforcement is to develop the policy enforcement point (PEP) acting as the intermediary between policy decision point (PDP) and client application. PEP forwards request from client to PDP system and retrieves access and usage control decision from PDP. PEP is also responsible for enforcing policy by executing obligation if needed.

4.3. Secure Development, Operation and Administration

In this category, there are five patterns.

1. Bastion Server.

   - **Problem**. How to access Cloud resources without exposing them directly to the Internet?
   - **Context**. Managing a secure virtual Cloud network requires privileged access to that network. Without proper isolation, such privileged access introduces vulnerabilities.
   - **Solution**. Using a special purpose computer or software module on a network specifically designed and configured to withstand attacks. The computer generally hosts a single application and all other services are removed or limited to reduce the threat to the computer. It is hardened in this manner primarily due to its location and purpose, which is either on the outside of a firewall or in a demilitarised zone (DMZ) and usually involves access from untrusted networks or computers.

2. Automated Threat Detection.

   - **Problem**. How to detect network attacks on Cloud internet endpoints?
   - **Context**. In today's complex Cloud systems, with multiple endpoints and edge nodes, make manual daily systems administration and security monitoring and analysis difficult if not entirely impossible.
   - **Solution**. Use the industrial specialised tool or software package for detecting automatically different form of threats or attacks. For example, AWS DIS. We can also look at other solutions available in research literature like that in [18,19].

3. Economic Durability.

   - **Problem**. How to establish and maintain availability of the Cloud services in the face of distributed denial-of-service attacks?
   - **Context**. A Denial of Service (DoS) attack is one in which a server or service is overwhelmed by traffic and consequently either disabled or made unavailable to its customers. In general, the effect on the target of a DoS attack is a loss of business, or in the less critical cases, just failure to get his/her message out. In Cloud context, DoS attacks on pay-as-you-go Cloud applications will result in a dramatic increase in Cloud utility bill, if Cloud-based service is designed to scale up automatically (e.g., Amazon EC2): user will see increased use of network bandwidth, CPU, and storage consumption. This type of attack is characterised as economic denial of sustainability (EDoS).
   - **Solution**. EDoS-Shield (e.g., AWS shield) is used to mitigate the Economic Denial of Sustainability (EDoS) attack in the Cloud computing systems. One technique used to mitigate of the EDoS attacks creating from spoofed IP addresses is hop-count filtering. Time to Live (TTL) parameter is used for calculating the supreme life time of packet inside the network. The TTL value was decremented each time when packet permitted through any router. When TTL value became zero, the packet was rejected.

4. Vulnerability Management.

   - **Problem**. How to detect and respond to found vulnerabilities?
   - **Context**. When an organisation moves applications and data to the Cloud, they will shift some - but not all - security responsibility to the Cloud provider. Most Cloud providers are responsible for securing their Cloud infrastructure (such as physical data centre security), while the Cloud user is responsible for their applications and data running in the Cloud platform. Thus, when developing and deploying application in Cloud, a key responsibility

for a security professional is to keep that environment free from vulnerabilities that attackers could use to get at organisation applications and data.

- **Solution**. To scan for vulnerabilities, a tool (e.g., Nessus) can be used in Cloud to scan for software flaws. Software update package is important in case a flaw is detected.

4.4. Privacy and Confidentiality

There are 4 patterns in this category.

1. End-to-End Security

   - **Problem**. How to communicate a message between two parties so that its confidentiality is protected across all components in the Cloud communication channel?
   - **Context**. In a Cloud environment, data passes through multiple communication channels and is stored in different layers that might be physically dispersed and controlled by different entities. Furthermore, Cloud providers' terms of service are often such that they are labelled as honest-but-curious in a threat model, which is a problem in situations where elevated security guarantees are needed for data in transit and at rest.
   - **Solution**. The cryptographic encryption should be used to secure the data exchanged between different entities in the network, both in transit and at rest.

2. Computation on Encrypted Data

   - **Problem**. How to outsource data for computation to a Cloud service without disclosing it in the process?
   - **Context**. Cloud services provide an attractive elastic computation model; however, to use Cloud beyond mere storage of encrypted data, the keys need to be made available to the Cloud provider in order to decrypt the data before computing on it. In threat models where Cloud is all but fully trusted, this poses a security issue.
   - **Solution**. Using fully homomorphic or SGX [20] scheme that allows complicated processing even though the data was encrypted and users couldn't see it.

3. Data Anonymisation

   - **Problem**. How to remove personal identifiers from datasets to protect privacy, while keeping the datasets still valuable for processing?
   - **Context**. Cloud services are often used to process large datasets, potentially containing primary or secondary private data, or such data can be inferred by correlating different datasets.
   - **Solution**. The identity of data owner needs to be stripped off from the records in such a way so that the data owner cannot be identified directly or indirectly from that anonymised data.

4. Processing Purpose Control

   - **Problem**. How to ensure data is used or processed in accordance with its original intended purpose?
   - **Context**. There are many challenges when building a Cloud-based application for storing and processing personal and private sensitive information (e.g., healthcare system). The challenges range from security to legal aspects; one of the challenging issue to address is to ensure that data shared between different concern parties in the network is used in accordance with its defined purpose. By law (e.g., GDPR), data processor is liable to the misuse of personal information and it has legal responsibilities to ensure that data processed in their system or in other system they share with is used in accordance with law, declared purpose and user consent.

- **Solution**. A reliable data usage control tools is required to control the usage of data. Usage control tool allows user to not only control and enforce the usage of data, but also trace and audit the usage of it.

4.5. Secure Architecture

There are 7 patterns in this category.

1. Virtual Network

   - **Problem**. How to connect components of a Cloud application architecture without unnecessarily exposing them to the Internet?
   - **Context**. Cloud applications often consist of public endpoints (APIs or web front ends) and a back-end infrastructure. The back-end infrastructure needs to be made unavailable from the outside world in order to reduce the attack surface.
   - **Solution**. The SaaS application and the end-point should be running in the virtual network, which protects the communication between application and end-point from exposing to the Internet. The solutions proposed in the literature such as that in [21] can also be considered.

2. Web Application Firewall

   - **Problem**. How to protect web API endpoints from unauthorised access and abuse?
   - **Context**. Cloud applications expose API end-points. These endpoints are often exposed to the Internet and as such are prone to different attacks.
   - **Solution**. Web access firewall should be used to control incoming and outgoing access to and from end-points.

3. Secure Element

   - **Problem**. How to securely provide and strongly protect identity of IoT devices?
   - **Context**. Protecting cryptographic secrets on IoT devices used to perform device identification to the Cloud backbone is difficult using off-the-shelf embedded setups which store secrets using standard file systems on flash drives. There are multiple side channels that can be exploited to extract secret keys and other sensitive information.
   - **Solution**. Using a unique identity, PKI should be the foundation of any IoT security strategy. With a unique strong device identity, things can be authenticated when they come online and ensure secure and encrypted communication between other devices, services and users. Because PKI is an established technology, it can be implemented immediately into IoT ecosystem today and easily integrates with other components of IoT security solutions. Other solutions proposed in the literature such as the one in [22] can also be considered.

4. Secure Cold Storage

   - **Problem**. How to protect the availability of large amounts of data securely and cost-effectively?
   - **Context**. Cloud computing service consumers often have legal or regulatory obligation to keep certain, otherwise no longer or seldom used, data for a predefined amount of time before deletion is allowed. Such data, when left in the online, readily-available application storage not only poses unnecessary burden on the budget, but also increases the vulnerability footprint.
   - **Solution**. Most of the Cloud service platforms provide an option for cold storage where the data can be stored temporarily or permanently. However, data stored on Cloud storage should be encrypted to ensure data confidentiality and integrity.

5. Certificate and Key Manager

    - **Problem**. How to securely and effectively create, provision and revoke certificates and keys for securing data at rest and in transit?
    - **Context**. In properly managed Cloud systems, cryptographic material, such as private/public key pairs and secret symmetric keys have a well-defined lifetime. However, already in simple systems, there could be tens of certificates and keys which require lifetime management.
    - **Solution**. Renewing certificate and cryptographic key is important in order to minimise the risk that may occur as the result of repeated usage of the old one. There should be the renewal policy for certificate and key either manually or automatically. In most of Cloud service platforms, they provide a tool to manage key as well as certificate, user can adopt either manual or automatic key renewal. User can also use/configure their own defined certificate and key (e.g., AWS key manager).

6. Hardware Security Module

    - **Problem**. How to best protect the cryptographic secrets owned by Cloud tenants while still enabling Cloud processing infrastructure to compute on the tenant data?
    - **Context**. Cloud providers use public and private cryptography to protect their tenant data in transit and at rest. However, to be able to freely compute on the data, the Cloud infrastructure needs access to plaintext information.
    - **Solution**. Use physical computing device that safeguards and manages digital keys for strong authentication and provides cryptoprocessing.

7. Secure Auditing

    - **Problem**. How to record and report security-related behaviour in an operating Cloud system?
    - **Context**. Audits are important for organisations in Cloud environment given a pay-per-use business model. A thoroughly conducted audit program can assure financial and operational well-being of an organisation.
    - **Solution**. Security Audit Guidelines defined the important steps or tasks to be executed for systematically reviewing and monitoring Cloud resources for security best practices. For example, AWS security audit guideline provides the steps to review the application security configuration ranging from user credential management to identity and access control management.

In Section 5, we discuss in detail the solutions in AWS and Microsoft Azure to our defined security patterns. The idea is to map each defined pattern to the solutions provided in AWS or/and Microsoft Azure.

Special note: We produced an official documentation and guideline of security patterns for Cloud SaaS. Each pattern is represented by an icon (see Figure 3) and accompanies by texts in which the detailed description of pattern and its associate solution are provided. All the documents are hosted on a webpage and are accessible at [23].

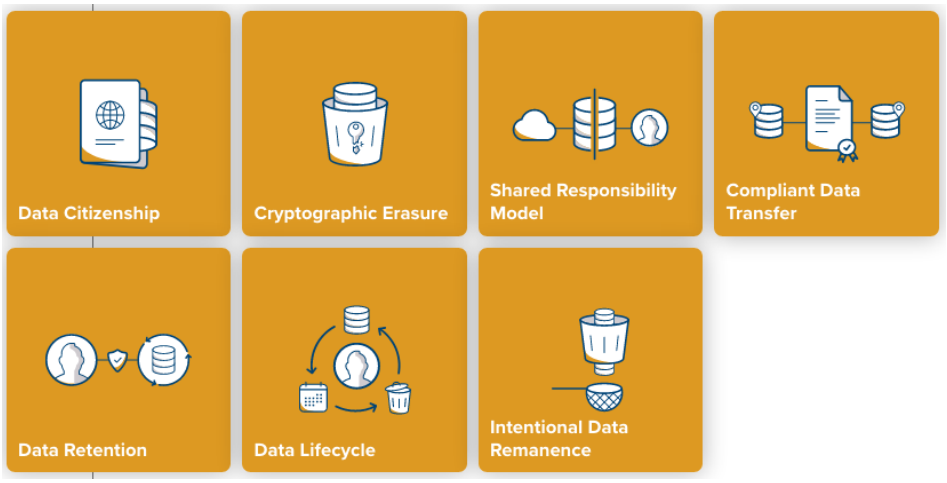

Figure 3. Official icons for compliance and regulatory pattern category. For other pattern categories, one can find them at [15].

5. Case study: Solutions in AWS and Azure

In this section, we focus on the results of our study on the security pattern solutions in two popular Cloud service providers: AWS and Azure. Other Cloud platform such as Google Cloud [24] will be included in our future study.

5.1. AWS

As shown in Figure 4, we can find in AWS most of the solutions required to address our defined security patterns. In this section, we provide short description of those solutions (see Figure 4).

1. **Compliance and Regulatory**. AWS provides different tools to support the regulatory compliance processing and management of data.

    - **Data Citizenship**. AWS offers location tags tool, known as geo restriction or geoblocking, to mange user access geographically. There are two options available for geo restriction in AWS. One is to use CloudFront geo restriction [23], which is able to limit access to content based on countries where access requests come from. CloudFront allows access to content if request is from a whitelist of approved countries and prevent access if request is from a blacklist of banned countries. In case the content distribution does not follow the countries boundaries, a combination of CloudFront and third-party geolocation service [25] can be used. This can allow access control to content based not only on country but also based on other parameters, such as city, zip or postal code, or latitude and longitude.
    - **Cryptographic Erasure**. AWS provides a service, known as KMS [26], for managing the cryptographic keys used in AWS. KMS allows user to create and manage keys and control the use of encryption across a wide range of AWS services and in applications.
    - **Shared Responsibility Model**. AWS provides different services and tools to ensure protection of data and system. Some services or tools are free and some are not. It is up to client to use it or not. The main responsibility of the AWS is to ensure the availability and basic security of Cloud platform.
    - **Compliant Data Transfer**. Similar to Data citizenship, the geo restriction tool can be used to control the transfer of data across geographical boundaries.
    - **Data Retention**. AWS also provides different type of data storage and backup tool, which can be used to protect data from accidental deletion and deliberate attacks on data. For example, DynamoDB [27] is a AWS database system that provides configurable backup (e.g., user can define their own backup policy).

- **Data Lifecycle**. AWS data lifecycle manager [28] allows use to manage the lifecycle of AWS resources and application data.With this service, user can create lifecycle policies that are used to automate operations on the specified resources.
- **Intentional Data Remanence**. To ensure that data remains even after attempts to delete or remove it, AWS provides backup services to all their database systems and storages. For example, DynamoDB backup service provides tools to user for defining their own backup policy to safeguard data in case of intentional or unintentional deletion of data.

2. **Identification, Authentication and Authorisation**. AWS provides a complete set of tools to support secure identification, authentication and authorisation of devices and users using AWS platform.

- **Multi-factor Authentication**. AWS provides different authentication tools for users as well as application API. For example, AWS Cognito [29] is an access control tool, which can be used to authenticate user and application API. Cognito supports both the standard username and password authentication as well as multi-factor authentication.
- **Federation (single sign-on)**. AWS SSO [30] provides a federate single-sign-on service and allows AWS accounts and business applications to be easily managed. SSO makes it easy to centrally manage access to multiple AWS accounts and business applications.
- **Access Token**. AWS security token service [31] is a web service that enables user to request temporary, limited-privilege credentials, which can be used to authenticate user or application API.
- **Mutual Authentication**. AWS client VPN [32] offers mutual authentication, which use certificates, generated from AWS certificate manager, to perform authentication between client and server.
- **Secure User on-boarding**. AWS customer on boarding service [33] provides protection to content from unauthorised use by combining a Secure Packager and Encoder Key Exchange (SPEKE) digital rights management (DRM) [33].
- **Identify and Access Manager**. AWS IAM [34] is a tools used to securely manage access to AWS services and resources. With IAM, one can create and manage AWS users and groups. We can also set permissions to allow or deny user's access to AWS resources.
- **Per-request Authentication**. AWS provides a service named "Signing and Authenticating REST Requests" [35] to allow or deny access or operation based on the identity of the requester. For example, a group of user is assigned the right to create buckets while another group has the right to create objects in a bucket.
- **Access Control Clearance**. AWS provides different tools and services to securely control and manage access to data and system resources. Some of which are IAM and Cognito for authentication and user management and AWS CloudWatch [36] for monitoring user as well as resources.

3. **Secure Development, Operation and Administration**. AWS provides enough tools for secure development, operation and administration of system deployed on its platform.

- **Bastion server**. AWS bastion [37] is specifically designed to withstand attacks. Its purpose is to provide access to a private network from an external network, such as the Internet and to minimise the chances of penetration of potential attacks.
- **Automated threat detection**. AWS GuardDuty [38] is a threat detection service that continuously monitors for malicious activity and unauthorised behaviour to protect AWS accounts, workloads and resources.
- **Durable Availability**. To ensure good service and availability of application running on AWS, some tools and services are in place, such as CloudWatch and WAF [39]. CloudWatch

provides secure user and resources monitoring while WAF acts as firewall protecting application from external attacks.

- **Economic Durability**. To prevent attackers from draining resources (e.g., computational resources), AWS puts in place CloudWatch [36] service to monitoring user, operation and the usage of resources of application deployed in the AWS platform.
- **Vulnerability Management**. There are some software tools, developed by third party, that can be used to find the vulnerability of application software deployed in the AWS platform. One of which is AlienVault USM [40].

Category	Pattern title	Solutions in AWS	Solutions in Azure
Compliance and Regulatory	**Data Citizenship**	Use AWS location tags to designate the location for data processing	Azure information protection and location tag. Azure frontdoor service
	Cryptographic Erasure	Use AWS KMS	Use Azure Key Vault
	Shared Responsibility Model	AWS provides different services to ensure protection of data and system. It is upto client to use it or not. However, AWS is responsible for only the vailability and basic security of cloud platform.	Azure provides different security tools to ensure protection of data and system. It is upto client to use it or not. However, Azure is responsible for only the vailability and basic security of cloud platform
	Compliant Data Transfer	AWS locaton tags	Azure location tag
	Data Retention	The data retention policies can be defined and executed by AWS. For example Lambda	Azure provides option to define data retention policy in Database system
	Data Lifecycle	AWS data lifecycle manager	Azure blob storage lifecycle
	Intentional Data Remanence	database (e.g. DynamoDB)	database (e.g. Azure backup)
Identification, Authentication and Authorisation	**Multi-Factor Authentication**	AWS Cognito	Azure active directory : multi-factor
	Federation (Single Sign-On)	AWS SSO (Single Sign-On)	Azure AD Seamless Single Sign-On
	Access Token	AWS security token service	Azure active directory : Token service
	Mutual Authentication	Use AWS TLS/SSL certificate, Certificate feature of API Gateway (AWS client VPN)	Azure App service
	Secure User Onboarding	AWS customer on boarding	Azure security center
	Identity and Access Manager	AWS IAM and Cognito	Azure IAM
	Per-request Authentication	AWS Signing and Authenticating REST Requests	Azure API management & REST API authentication
	Access Control Clearance	AWS cloud watch and AWS Cognito/IAM	Azure access control service
Secure Development, Operation and Administration	**Bastion Server**	AWS bastion host	Azure Bastion host
	Automated Threat Detection	AWS GuardDuty	Azure advanced threat protection
	Durable Availability	AWS cloud watch, AWS WAF	Azure web access firewall & firewall application gateway
	Economic Durability	AWS cloud watch	Azure Monitor
	Vulnerability Management	AWS vulnerability scanning	Vulnerability scan in Azure security center
Privacy and Confidentiality	**End-to-End Security**	AWS KMS, Certificate manager	Azure Key Vault
	Computation on Encrypted Data	N/A	N/A
	Data Anonymisation	Algorithms can be defined and ran by AWS module (e.g. lambda)	Azure provides Dynamic Data Masking on SQL database
	Processing Purpose Control	N/A	N/A
Secure Architecture	**Virtual Network**	AWS Virtual Private Cloud	Azure Virtual Network
	Web Application Firewall	AWS WAF	Azure application firewall gateway
	Secure Element	AWS IoT Device Management	Azure IoT Hub & IoT Suit
	Secure Cold Storage	AWS Glacier	Azure Coldblob storage
	Certificate and Key Manager	AWS Certificate and Key manager (AWS KMS)	Azure Key Vault
	Hardware Security Module	AWS CloudHSM	Azure Dedicated HSM
	Secure Auditing	AWS Auditing Security Checklist	Azure Monitor, Stream, Network Watcher

Figure 4. Cloud SaaS security patterns and solutions in AWS and Microsoft Azure.

4. **Privacy and Confidentiality**. WAS provides some services and tools to support privacy and confidentiality protection. However, some of requirements are not yet addressed by AWS. SaaS developer needs to develop their own solutions to address them, such as data anonymisation, computation on encrypted data and processing purpose control.

- **End-to-end Security**. AWS provides services to secure data exchanged between different entities (e.g., back-end and front-end). AWS certificate manager [41] is a service that allows easy provisioning, management, and deployment of public and private SSL/TLS certificates for use with AWS services and internal connected resources.

- **Computation on Encrypted Data**. The solution for computation on encrypted data is not yet available in AWS platform.
- **Data Anonymisation**. The solution for automatic data anonymisation is not yet available in AWS platform.
- **Processing purpose control**. The solution for processing purpose control on data circulated in AWS platform is not yet available.

5. **Secure Architecture**. A set of secure architecture tools and services are available in AWS.

- **Virtual Network**. AWS virtual private Cloud [42] is a service allowing user to provision a logically isolated section of the AWS Cloud where user can launch AWS resources in his defined virtual network. User can have complete control over his virtual networking environment, including selection of IP address range, creation of subnets, and configuration of routing tables and network gateways.
- **Web Application Firewall**. AWS WAF [39] is a web application firewall that protects web applications from common web exploits that could affect application availability, compromise security, or consume large amount of resources.
- **Secure Element**. AWS IoT Device Management [43] offers an easy way to securely onboard, monitor and manage IoT devices at scale. With this tools, user can register their connected devices individually or in group, and easily manage permissions so that devices can remain secure after on boarding.
- **Secure Cold Storage**. Retrieving data from large database can take longer time than a smaller one. To maintain a fast data retrieval time, it is important to store infrequent access data in a separate storage. AWS offers a storage service named Glacier [44] for storing infrequent use/access data.
- **Certificate and Key Manager**. AWS provides two services for managing cryptographic keys and certificate for TLS/SSL. AWS key manager allows user to create, provision and manage the cryptographic keys while certificate manager [41] allows user to provision and manage TLS/SSL certificates.
- **Hardware Security Module**. AWS CloudHSM [45] is a Cloud-based hardware security module that allows user to generate encryption keys or configure his own cryptographic key to be used for applications and services in AWS.
- **Secure Auditing**. AWS has published the auditing security checklist [46] to facilitate and assist customer (or AWS user) in evaluating the ability of AWS services to meet their information security objectives and security controls needed by their specific industrial and governing body.

5.2. Azure

Similar to AWS, Azure provides most of the solutions required to address security patterns defined in Section 4. Figure 4 provides the mappings between defined security patterns and solutions available in Azure. We provide, in this section, short description of Azure solutions to the defined security patterns (see Figure 4).

1. **Compliance and regulatory**. Different tools and services available in Azure can be used to support the regulatory compliance processing and management of data. We describe each solution below.

- **Data Citizenship**. Azure Front Door (AFD) [47] Service can be used to limit access to data and application geographically. Web application firewall (WAF) service at Front Door allows user to define a policy using custom access rules for specific path on endpoint to allow or deny access from specified countries.

- **Cryptographic Erasure**. Azure key vault [48] enables user to manage their cryptographic key securely (e.g., delete or erase key). Azure key vault contains also other services such as, secrets management, certificate management and also stores secrets backed by hardware security modules.
- **Shared Responsibility Model**. Similar to AWS, Azure offers different services and tools to ensure protection of data and system. However, not all services and tools are free. It is up to client to use them to their needs. The legal responsibility of Azure is to ensure the availability and basic security of Cloud platform.
- **Compliant Data Transfer**. Similar to Data citizenship, the AFD service can be used to control the transfer of data across geographical boundaries.
- **Data Retention**. Azure offers different type of data storage and backup tools (e.g., Blob storage, files storage and table storage), which can be used to protect data from accidental deletion and deliberate attacks on data, for instance, Azure Cosmos DB [49], a distributed multi-model database service..
- **Data Lifecycle**. To control the lifecycle of data in Azure, Azure blob storage lifecycle [50] management can be used. This service offers a rich, rule-based policy which user can use to control data from its creation, its usage to end of its lifecycle.
- **Intentional Data Remanence**. Azure Backup [51] can be used to store data in such a way so that it can remain even after attempts to delete or remove. Azure Backup offers backup services to all resources in Azure Cloud platform.

2. **Identification, Authentication and Authorisation**. A wide range of tools are available in Azure to support secure identification, authentication and authorisation of users, applications and devices using Azure platform.

   - **Multi-factor Authentication**. Azure active directory provides a collection of services for user and data access control, one of which is multi-factor authentication [52]. Azure uses two-steps verification consisting of the traditional username and password method and the additional authentication method (e.g., proof of possession).
   - **Federation (single sign-on)**. Azure Active Directory Seamless Single Sign-On (Azure AD Seamless SSO) [53] allows user to be automatically sign in when they are on their corporate devices connected to network. With AD single sign-on, users do not need to provide their passwords to sign in to Azure AD. This feature makes access to Cloud-based applications easier, especially, when user has multiple Cloud accounts.
   - **Access Token**. Azure Active Directory access tokens [54] allows user to generate access token, which can be used to securely call APIs protected by Azure.
   - **Mutual Authentication**. Azure app service [55] provides mutual authentication service that can be used to restrict access to Azure applications and resources.
   - **Secure User on-boarding**. Azure Security Centre [56] consists of different security services for protecting applications, resources and devices on boarding the network. Secure on-boarding service is used to securely on-board non-Azure devices to the network.
   - **Identity and Access Manager**. In Azure, identity of user and application can be protected by using Azure IAM [57]. This service enables protection of users, applications and data at the front gate and defends against malicious login attempts and safeguard credentials with risk-based access controls and strong authentication method (e.g., two factors authentication).
   - **Per-request Authentication**. Azure API Management and REST API Authentication [58] enables user to create the access token required to make calls into the Azure APIs.
   - **Access Control Clearance**. Azure Access Control Service (ACS) and Azure IAM are two services enabling control of users, access and usage of data, application and resources in Azure.

3. **Secure Development, Operation and Administration**. Azure provides enough tools for secure development, operation and administration of system deployed on its platform.

    - **Bastion Server**. Azure bastion host [59] provides a single point of entry that allows users to access the deployed resources in Azure environment. The bastion host offers a secure connection to deployed resources by only allowing remote traffic from public IP addresses on a permitted list.
    - **Automated Threat Detection**. Azure Advanced Threat Protection (ATP) [60] is a Cloud-based security solution allowing identifing, detecting, and investigating advanced threats, compromised identities and malicious insider actions.
    - **Durable Availability**. Azure provides tools, such as Azure web access firewall (WAF) and firewall application gateway [61], to ensure good service and availability of application running on Azure. WAF acts as firewall protecting application from external attacks while Azure Application Gateway is a web traffic load balancer enabling effective management of traffic to web applications.
    - **Economic Durability**. Azure monitor [62] helps user to understand how his applications are performing and proactively identifies issues affecting them and the resources they depend on. This helps maximising the availability and performance of applications.
    - **Vulnerability Management**. Similar to AWS CloudWatch, Azure vulnerability scanning [63] in Azure security centre can be used to observe attacks, prevent unauthorised access and find vulnerability in the system.

4. **Privacy and Confidentiality**. End-to-end security is assured by Azure tools such as, Key Vault. However, many features concerning privacy protection are not yet supported by Azure. SaaS developer needs to develop their own solutions to support services, such as computation on encrypted data and processing purpose control.

    - **End-to-end Security**. Azure key vault provides different services for key management and certificate management, provisioning and deployment of public and private SSL/TLS.
    - **Computation on encrypted data**. Although Azure provides service such as confidential computing to offer protection of data while in use, it is not yet possible to compute encrypted data in Azure platform.
    - **Data Anonymisation**. With Azure SQL Database dynamic data masking [64], user can limit sensitive data exposure by masking it to non-privileged users. Dynamic data masking allows user to monitor and prevent unauthorised access to sensitive data by enabling user to assign the amount of sensitive data to reveal.
    - **Processing Purpose Control**. The solution for processing purpose control on data circulated in Azure platform is not yet available.

5. **Secure Architecture**. A set of secure architecture tools are available in Azure ranging from virtual private network to secure auditing.

    - **Virtual Network**. Azure Virtual Network [65] enables different types of Azure resources, such as Azure Virtual Machines, to securely communicate with each other. User can create multiple virtual networks within each Azure subscription and Azure region.
    - **Web Application Firewall**. Web application firewall (WAF), a service in Azure application gateway [61], provides centralised protection of web applications from common exploits and vulnerabilities, such as SQL injection and cross-site scripting.
    - **Secure Element**. Azure IoT hub and suit [66] provides IoT services, which enable user to easily connect devices to the Cloud.

- **Secure Cold Storage**. Azure Cool Blob Storage [67] is a low cost storage for infrequent use data. For example, the use cases of cool storage include backups, media content and archival data. In general, infrequent access data is a perfect candidate for cool storage.
- **Certificate and Key Manager**. Azure key vault [48] is Cloud service that provides supports for cryptographic key and certificate management. It allows user to create, provision and manage cryptographic key as well as TLS/SSL certificate.
- **Hardware Security Module**. Azure Dedicated HSM [68] allows user to manage and have full administrative and cryptographic control over Keys. Even Microsoft has no access to the keys stored in HSM.
- **Secure Auditing**. Azure provides different services for secure auditing of resource use and application operation. Azure monitor can be used to audit the use of resources and Azure network watcher [69] is used for diagnosing connections issues.

Remark: AWS and Azure provide solutions to most of the security patterns we defined in Section 4. However, some are still missing and SaaS developers need to develop their own solutions to address them (see notes in Figure 4).

6. Related Work

There is a significant number of security pattern research in software engineering [8,9,70–75]. However, that research focused on general topic and not particularly on the Cloud. Moreover, they limit themselves to very narrow topics, such as authentication and authorisation security or threat and ignore some other security issues that they think are not significant, for instance, resources management. In our work, we focus on all the aspects of security in Cloud SaaS. We cover data and system security and privacy, which is normally overlooked given its complexity. Below are some most relevant research to our work.

Yoder and Barcalow [76] presented a collection of seven patterns for application security, treating authentication and authorisation security aspects. The authors are motivated by the usual lack of security perspective in the early phases of the software design and attempt to solve this problem by providing design guidelines that make it easier to implement security details later in development. Some patterns, proposed by the authors, are described more on the architectural level while the others are more design-oriented. The authors provide also a pattern language defining relations between the presented security patterns. In this paper, the authors proposed the general patterns applied to software application and not specifically for Cloud SaaS. Moreover, there are many missing patterns that are not addressed in this paper, such as privacy and confidentiality, data management and governance, threat detection and resources management. These distinguish this work from ours.

Focussing on confidentiality, integrity and non-repudiation security aspects, Braga et al. [70] developed a pattern language "Tropyc", consisting of nine design patterns "for cryptographic software". The authors first introduced a "Generic Object- Oriented Cryptographic Architecture", a simple system of two template classes representing two sides in a communication channel with two helper classes representing encryption and decryption transformations. Based on this foundation and addressing four "well established cryptographic objectives", they went on to develop four basic patterns: Sender Authentication, Information Secrecy, Signature and Message Integrity and their further derivatives. This research focuses mainly on authentication, data integrity and end-to-end security including secure management of digital signature and secret information management. Unlike our work where we address all the aspects of security from data and system to privacy, this paper addresses only a small aspect of security issues and most importantly it is not about Cloud.

Romanosky [71] identifies eight enterprise security patterns, attempting to cover topics like data authenticity and data ownership, access control and authentication, risk assessment and management, communication with third parties, secure provision of data, as well as awareness of own vulnerabilities and threats. Using a brief form giving motivation and problem statements, followed by a description

of the forces governing the problem and a proposed solution with consequences, the author employs a series of questions and answers to provide guidance in development and enforcement of prudent security policies in an enterprise. The work of Romanosky covers more security aspects compared with the work of Yoder and Braga. However, Romanosky's work is not specifically for Cloud.

Kienzle et al. [77] published a catalogue of 30 patterns. They made one of the first attempts at security pattern classification, distinguishing between 16 "structural" and 14 "procedural" patterns. The structural patterns describe architectural and design aspects of the elements of a secure system, providing recipes for implementation of security mechanisms. They deal with security aspects, such as authentication and authorisation, web application session management, encryption of data in transit and at rest, sandboxing and need-to-know principle implementation, least privilege principle and secure transaction. This is very close to ours; however, the difference is that authors works on general software security pattern and not for Cloud. Although there are similarity and common security issues between Cloud and non-cloud application, Cloud application has more and specific security issues compared with non-cloud application, for instance, economic durability.

The book on security patterns by Schumacher et al. [78] represents a culmination of the work done to its date and the synthesis of the individual efforts of not just its large consortium of authors, but the security pattern community in general [72,73]. This large volume showcases about 70 patterns, classified using well-developed pattern taxonomy units: enterprise security and risk management, identification and authentication, access control, accounting, firewall, secure internet applications and cryptographic key management. In addition to being a pattern repository, this work includes four introductory chapters explaining the pattern approach in general, an overview of security foundations followed by an overview of the history, the concept and the scope of security patterns. Similar to the work of Kienzle et al. [77], Schumacher et al. [78] provide a rich documentation on security patterns for general information system, but not for Cloud SaaS.

Eduardo Fernandez-Buglioni, one of the stalwarts of the pattern movement and a credited co-author of [78], published his own catalogue of security patterns [79], integrating and systematising own earlier publications. Similar in organisation and volume, but decidedly different in classification, pattern names and their presentation, the work makes extensive use of UML component, class and collaboration diagrams. Patterns are classified by their usage areas: identity management, authentication, access control, (operating system) process management, secure execution and file management, secure OS architecture and administration, networking, web services and web service cryptography. Similar to previous work, there are many missing security patterns in Eduardo's work, such as data governance and privacy and confidentiality. Moreover, these patterns are not defined specifically for Cloud SaaS.

Langer et al. [12] work on Cloud security patterns to improve end user security and privacy in public Clouds. The authors developed several Cloud security patterns for common critical situations in the Cloud in the three fields of data storage in the Cloud, user privacy protection and data minimisation, and authentication of stored and processed data. The authors focus mainly on how to protect data in public Clouds, many security patterns, such as communication, secure architecture and data governance are not considered in this paper.

Oracle technical report [80] on "Securing SaaS at Scale" highlights some security challenges in Cloud SaaS, such as security and compliance, user management and monitoring and data residency and regulatory. However, in the report only high level and general discussion on the matters are provided. Moreover, no solutions to the highlighted issues are provided in the report. In addition, other security issues, such as secure development, operation and administration and secure architecture of Cloud SaaS are not discussed in the report. These differentiate their work from our work in this paper.

Remark: There is no attempt so far to fully document the security patterns for Cloud SaaS application, a security best practices and security knowledge documentation that SaaS developer can use as a guideline for developing Cloud SaaS application from the ground up. We put our effort in studying, defining and documenting the security patterns for Cloud SaaS. Among the 8 most relevant papers presented in this section, only 6 [70,71,76–79] work on security patterns for general software (non-cloud application) and only 2 papers [12,80] address security patterns for Cloud-based application. However, the the authors in the 2 papers do not cover all necessary security patterns in Cloud compared with our work. In our work, we provide richer patterns and cover more security aspects.

7. Conclusions

The security patterns have gone through their entire "hype-cycle" [81] and are now considered mature and well explored from the perspective of the pattern classification and their application. New areas and specialisations, such as security patterns arising from the specifics of Cloud computing environments are steadily coming into researchers' focus. In this paper, we address this new area, the Cloud. We identify the Cloud SaaS security patterns for different security aspects, from data and system security to privacy. We provide a complete list of patterns and their solutions applicable to Cloud computing environment, and to the best of our knowledge, there is no attempt so far to fully study and document them. There are, of course, some research literature focusing on security pattern; however, they address only the selected topics, such as authentication, authorisation or threat and ignore some other security issues that they think are not significant, for instance, resources and operation management and governance. In addition, and what make our work in this paper different from previous work is that we produce an official security best practices and security knowledge documentation [15] that SaaS developer can use as a guideline for developing Cloud SaaS application from the ground up. Another contribution that also distinguishes our work from previous work is the study of AWS and Azure security solutions. The main goal is to map each pattern to the solutions available in AWS and Azure.

Although most of security solutions are available in AWS and Azure, there are still some remaining issues that need to be addressed, such as, processing purpose control and effective data lifecycle management in distributed environment (e.g., when data is shared between Cloud back-ends system). Our future work will be around these issues. In addition, we plan to include Google Cloud solutions to our security patterns documentation.

Author Contributions: We list the contribution of each author as following. Conceptualization, A.R., B.S., N.B. and P.T.; methodology, A.R. and P.T.; validation, A.R. and B.S.; formal analysis, A.R. and B.S.; writing, review and editing, A.R., B.S. and P.T.

Funding: This research was funded by Innoviris http://innoviris.be grant number 2015-PFS-ICT-6. Innoviris is the Brussels Institute for the encouragement of scientific research and innovation. Their mission is to support and stimulate research, development and innovation in and for Brussels through the funding of innovative projects by companies, research organisations and the non-commercial sector.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AWS	Amazon Web Service
WAF	Web Access Firewall
SSO	Single-Sign-On
OWASP	Open Web Application Security Project
KMS	Key Management System
TLS	Transport Layer Security
SSL	Secure Sockets Layer

References

1. Alexander, C.; Ishikawa, S.; Silverstein, M. *A Pattern Language: Towns, Buildings, Construction*; Oxford University Press: Oxford, UK, 1977.
2. Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. *Design Patterns: Elements of Object-Oriented Software*; Addison-Wesley: Boston, MA, USA, 1994.
3. Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerlad, P.; Stal, M. *Pattern-Oriented Software Architecture: A System of Patterns*; Wiley & Sons: Hoboken, NJ, USA, 1996.
4. Schumacher, M. *Security Engineering with Patterns: Origins, Theoretical Models, and New Applications*; Springer: Berlin/Heidelberg, Germany, 2003.
5. Bunke, M.; Koschke, R.; Sohr, K. Organizing security patterns related to security and pattern recognition requirements. *Int. J. Adv. Secur.* **2012**, *5*, 46–67.
6. Bunke, M. Software-security patterns: Degree of maturity. In Proceedings of the 20th European Conference on Pattern Languages of Programs, Bavarian Kloster Irsee, Germany, 8–12 July 2015.
7. Blakley, B.; Heath, C. The Open Group Security Forum. Security Design Patterns. 2004. Available online: https://publications.opengroup.org/g044 (accessed on 29 March 2019).
8. Fernandez, E.B.; Monge, R. A security reference architecture for Cloud systems. In Proceedings of the WICSA 2014 Companion Volume, Sydney, Australia, 7–11 April 2014; p. 3.
9. Fernandez, E.B.; Monge, R.; Hashizume, K. Two patterns for Cloud computing: Secure virtual machine image repository and Cloud policy management point. In Proceedings of the 20th Conference on Pattern Languages of Programs, Monticello, IL, USA, 23–26 October 2013; p. 15.
10. Hafiz, M.; Johnson, R.E. Security Patterns and Their Classification Schemes. Department of Computer Science, University of Illinois at Urbana-Champaign, 2006. Available online: https://munawarhafiz.com/research/patterns/secpatclassify.pdf (accessed on 29 March 2019).
11. Web Application Security Guidance. Available online: https://www.owasp.org/index.php/Web_Application_Security_Guidance (accessed on 29 March 2019).
12. Langer, T.; Pohls, H.C.; Ghernaouti, S. *Selected Cloud Security Patterns to Improve End User Security and Privacy in Public Clouds. Privacy Technologies and Policy. APF 2016. Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2016; Volume 9857.
13. Fernandez, E.B.; Yoshioka, N.; Washizaki, H. Patterns for security and privacy in Cloud ecosystems. In Proceedings of the IEEE 2nd Workshop on Evolving Security and Privacy Requirements Engineering (ESPRE), Ottawa, ON, Canada, 25 August 2015.
14. OWASP Cloud Security Project. Available online: https://www.owasp.org/index.php/OWASP_Cloud_Security_Project (accessed on 29 March 2019).
15. Detailed Pattern Structure. Available online: http://www.sirris.be.s3-website-eu-west-1.amazonaws.com/ (accessed on 29 March 2019).
16. Spasic, B.; Rath, A.; Thiran, P.; Boucart, N. Security Pattern for Cloud SaaS: From system and data security to privacy. In Proceedings of the 4th IEEE International Conference on Cloud Computing Technologies and Applications, Brussels, Belgium, 26–28 November 2018.
17. General Data Protection Regulation (GDPR). Available online: https://ec.europa.eu/info/law/law-topic/data-protection_en (accessed on 29 March 2019).
18. Achbarou, O.; Kiram, M.A.E.; Bouanani, S.E. Securing Cloud Computing from Different Attacks Using Intrusion Detection Systems. *Int. J. Interact. Multimed. Artif. Intell.* **2017**, *4*, 61–64. [CrossRef]
19. Subramaniam, T.K.; Deepa, B. Security attack issues and mitigation techniques in Cloud computing environments. *Int. J. UbiComp (IJU)* **2016**, *7*. [CrossRef]
20. SGX. Available online: https://software.intel.com/en-us/blogs/2018/11/08/microsoft-azure-confidential-computing-with-intel-sgx (accessed on 29 March 2019).
21. Taherizadeh, S.; Stankovski, V.; Grobelnik, M. A Capillary Computing Architecture for Dynamic Internet of Things: Orchestration of Microservices from Edge Devices to Fog and Cloud Providers. *Sensors* **2018**, *18*, 2938. [CrossRef] [PubMed]
22. Ondiege, B.; Clarke, M.; Mapp, G. Exploring a new security framework for remote patient monitoring devices. *Computers* **2017**, *6*, 11. [CrossRef]

23. AWS CloudFront Geo Restriction. Available online: https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/georestrictions.html#georestrictions-Cloudfront (accessed on 29 March 2019).
24. Google Cloud. Available online: https://Cloud.google.com (accessed on 29 March 2019).
25. Third Party Geo Restriction. Available online: https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/georestrictions.html#georestrictions-geolocation-service (accessed on 29 March 2019).
26. AWS KMS. Available online: https://aws.amazon.com/kms/ (accessed on 29 March 2019).
27. AWS DynamoDB. Available online: https://aws.amazon.com/dynamodb/ (accessed on 29 March 2019).
28. AWS Data Lifecycle Manager. Available online: https://docs.aws.amazon.com/dlm/latest/APIReference/Welcome.html (accessed on 29 March 2019).
29. AWS Cognito. Available online: https://aws.amazon.com/cognito/ (accessed on 29 March 2019).
30. AWS SSO. Available online: https://aws.amazon.com/single-sign-on/ (accessed on 29 March 2019).
31. AWS SSO. Available online: https://docs.aws.amazon.com/STS/latest/APIReference/Welcome.html (accessed on 29 March 2019).
32. AWS Client VPN. Available online: https://docs.aws.amazon.com/vpn/latest/clientvpn-admin/authentication-authrization.html (accessed on 29 March 2019).
33. AWS Customer on Boarding Service. Available online: https://docs.aws.amazon.com/speke/latest/documentation/customer-onboarding.html (accessed on 29 March 2019).
34. AWS IAM. Available online: https://aws.amazon.com/iam/ (accessed on 29 March 2019).
35. AWS Signing and Authenticating REST Requests. Available online: https://docs.aws.amazon.com/AmazonS3/latest/dev/RESTAuthentication.html (accessed on 29 March 2019).
36. AWS CloudWatch. Available online: https://aws.amazon.com/Cloudwatch/ (accessed on 29 March 2019).
37. AWS Bastion Host. Available online: https://aws.amazon.com/blogs/security/controlling-network-access-to-ec2-instances-using-a-bastion-server/ (accessed on 29 March 2019).
38. AWS GuardDuty. Available online: https://aws.amazon.com/guardduty/ (accessed on 29 March 2019).
39. AWS WAF. Available online: https://aws.amazon.com/waf/ (accessed on 29 March 2019).
40. AlienVault USM. Available online: https://www.alienvault.com (accessed on 29 March 2019).
41. AWS Certificate Manager. Available online: https://aws.amazon.com/certificate-manager/ (accessed on 29 March 2019).
42. AWS Virtual Private Cloud. Available online: https://aws.amazon.com/vpc/ (accessed on 29 March 2019).
43. AWS IoT Device Management. Available online: https://aws.amazon.com/iot-device-management/ (accessed on 29 March 2019).
44. AWS Glacier. Available online: https://aws.amazon.com/glacier/ (accessed on 29 March 2019).
45. AWS CloudHSM. Available online: https://aws.amazon.com/Cloudhsm/ (accessed on 29 March 2019).
46. AWS Auditing Security Checklist. Available online: https://aws.amazon.com/blogs/security/auditing-security-checklist-for-aws-now-available/ (accessed on 29 March 2019).
47. Azure Front Door. Available online: https://docs.microsoft.com/en-us/azure/frontdoor/front-door-geo-filtering (accessed on 29 March 2019).
48. Azure Key Vault. Available online: https://docs.microsoft.com/en-us/azure/key-vault/key-vault-overview (accessed on 29 March 2019).
49. Azure Cosmos DB. Available online: https://docs.microsoft.com/en-us/azure/cosmos-db/introduction (accessed on 29 March 2019).
50. Azure Blob Storage Lifecycle Management. Available online: https://azure.microsoft.com/en-us/blog/azure-blob-storage-lifecycle-management-public-preview/ (accessed on 29 March 2019).
51. Azure Backup. Available online: https://azure.microsoft.com/en-us/services/backup/ (accessed on 29 March 2019).
52. Azure Multi-Factor Authentication. Available online: https://docs.microsoft.com/en-us/azure/active-directory/authentication/concept-mfa-howitworks (accessed on 29 March 2019).
53. Azure Single Sign-on. Available online: https://docs.microsoft.com/en-us/azure/active-directory/hybrid/how-to-connect-sso (accessed on 29 March 2019).
54. Azure Access Token. Available online: https://docs.microsoft.com/en-us/azure/active-directory/develop/access-tokens (accessed on 29 March 2019).
55. Azure App Service. Available online: https://docs.microsoft.com/en-us/azure/app-service/app-service-web-configure-tls-mutual-auth (accessed on 29 March 2019).

56. Azure Security Centre. Available online: https://docs.microsoft.com/en-us/azure/security-center/security-center-onboarding (accessed on 29 March 2019).
57. Azure IAM. Available online: https://azure.microsoft.com/en-us/product-categories/identity/ (accessed on 29 March 2019).
58. Azure API Management REST API Authentication. Available online: https://docs.microsoft.com/en-us/rest/api/apimanagement/apimanagementrest/azure-api-management-rest-api-authentication (accessed on 29 March 2019).
59. Azure Bastion Host. Available online: https://docs.microsoft.com/en-us/azure/security/blueprints/ffiec-paaswa-overview (accessed on 29 March 2019).
60. Azure Advanced Threat Protection. Available online: https://docs.microsoft.com/en-us/azure-advanced-threat-protection/what-is-atp (accessed on 29 March 2019).
61. Azure Application Gateway. Available online: https://docs.microsoft.com/en-us/azure/application-gateway/overview (accessed on 29 March 2019).
62. Azure Monitor. Available online: https://docs.microsoft.com/en-us/azure/azure-monitor/overview (accessed on 29 March 2019).
63. Azure Vulnerability Scanning. Available online: https://docs.microsoft.com/en-us/azure/security-center/security-center-vulnerability-assessment-recommendations (accessed on 29 March 2019).
64. SQL Database Dynamic Data Masking. Available online: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-dynamic-data-masking-get-started (accessed on 29 March 2019).
65. Azure Virtual Network. Available online: https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview (accessed on 29 March 2019).
66. Azure IoT Hub and Suit. Available online: https://azure.microsoft.com/en-us/resources/videos/connect-2015-introduction-to-azure-iot-suite-and-iot-hub-for-developers/ (accessed on 29 March 2019).
67. Azure Cool Blob Storage. Available online: https://azure.microsoft.com/pl-pl/blog/introducing-azure-cool-storage/ (accessed on 29 March 2019).
68. Azure Dedicated HSM. Available online: https://azure.microsoft.com/en-us/services/azure-dedicated-hsm/ (accessed on 29 March 2019).
69. Azure Network Watcher. Available online: https://azure.microsoft.com/en-us/services/network-watcher/ (accessed on 29 March 2019).
70. Braga, R.T.V.; Germano, F.S.; Masiero, P.C. A pattern language for business resource management. In Proceedings of the PLOP (PLoP'99), Monticello, IL, USA, 15–18 August 1999; pp. 1–33.
71. Romanosky, S. Security Design Patterns Part 1. 2001. Available online: https://citeseer.ist.psu.edu/viewdoc/download?doi=10.1.1.21.4117&rep=rep1&type=pdf (accessed on 29 March 2019).
72. Hafiz, M.; Adamczyk, P.; Johnson, R.E. Growing a pattern language (for security). In Proceedings of the ACM international symposium on New ideas, new paradigms, and reflections on programming and software, Tucson, AZ, USA, 21–25 October 2012; pp. 139–158.
73. Yskout, K.; Heyman, T.; Scandariato, R.; Joosen, W. Security Patterns: 10 Years Later. Interim Report. Available online: http://www.cs.kuleuven.be/publicaties/rapporten/cw/CW514.abs.html (accessed on 29 March 2019).
74. Hashizume, K.; Yoshioka, N.; Fernandez, E.B. Misuse patterns for Cloud computing. In Proceedings of the 2nd Asian Conference on Pattern Languages of Programs, Tokyo, Japan, 5–8 October 2011; p. 12.
75. Schumacher, M.; Roedig, U. Security engineering with patterns. In Proceedings of the 8th Conference on Pattern Languages of Programs, Allerton Park Monticello, IL, USA, 11–15 September 2001.
76. Yoder, J.; Barcalow, J. Architectural patterns for enabling application security. In Proceedings of the 4th Conference on Pattern Languages of Programs, Monticello, IL, USA, 2–5 September 1997; pp. 1–3.
77. Kienzle, D.M.; Elder, M.C.; Tyree, D.; Edwards- Hewitt, J. Security Patterns Repository, Version 1.0. 2002. Available online: http://www.cse.msu.edu/~cse870/Homework/SS2005/HW5/Kienzle.pdf (accessed on 29 March 2019).
78. Schumacher, M.; Fernandez-Buglioni, E.; Hybertson, D.; Buschmann, F.; Sommerlad, P. *Security Patterns: Integrating Security and Systems Engineering*; Wiley: Hoboken, NJ, USA, 2013.
79. Fernandez-Buglioni, E. *Security Patterns in Practice: Designing Secure Architectures Using Software Patterns*; John Wiley & Sons: Hoboken, NJ, USA, 2013.

80. Oracle Technical Report in 2018, "Securing SaaS at Scale". Available online: http://www.oracle.com/us/solutions/Cloud/Cloudessentials-securingsaas-5101707.pdf (accessed on 29 March 2019).
81. Heyman, T.; Yskout, K.; Scandariato, R.; Joosen, W. An analysis of the security patterns landscape. In Proceedings of the third International Workshop on Software Engineering for Secure Systems, Washington, DC, USA, 20–26 May 2007; p. 3.