Institute of Parallel and Distributed Systems
University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Master Thesis Nr. 3452

Securing Cloud Applications with Two-Factor Authentication

Umair Ashraf

Course of Study:	M.Sc. INFOTECH
Examiner:	Prof. Dr.-Ing. habil. Bernhard Mitschang
Supervisor:	Dipl. Inf. Tim Waizenegger
Commenced:	January 01, 2013
Completed:	July 03, 2013
CR-Classification:	D.4.6, K.6.5

Abstract

Content management Software as a Service (SaaS) applications have made a lot of attention in the recent years. The software and related content is hosted in cloud and remote access is given to the users through a web browser or a thin web client. The content management SaaS solutions store the regulatory content of an organization in cloud. Any successful attempt of unauthorized access to the cloud content can pose certain security risks, ranging from financial loss, defamation, to civil or criminal crime.

Security and privacy are two major hindrance for cloud consumers in adopting SaaS based cloud applications [SS11]. We need a solution to maximize the level of trust between the cloud consumers and the cloud providers. The level of trust can be increased by increasing information security and privacy, which boils down to strong authentication, authorization and access control mechanism. This thesis focuses on new technologies to improve authentication of services consumed in the cloud. Password authentication is the commonly used single-factor authentication mechanism. The password authentication is defenceless to many security threats. Passwords are vulnerable to replay and discovery attacks. They also do not show any resistance to eavesdropping, man-in-the-middle or phishing attacks. Two-factor authentication opens up new horizons in security enhancement. It mandates users to provide two authentication tokens during the authentication phase. The two authentication tokens cover vulnerabilities of each other and combine together to provide higher information security.

Ensuring strong authentication is a complete process within itself. The probability of occurrence of a security breach and the loss involved in it play a decisive role in selecting an authentication assurance level. The assurance level is the measurement of the strength of an authentication process. The appropriate technology is selected to meet a certain assurance level and mitigates the exposed risk to an information system. Selecting the appropriate technology includes selecting the authentication tokens, choosing the token management policy and determining the communication protocol between the client and the server. Also, authentication security enhancement is a cyclic process and requires continuous monitoring and improvement.

The two-factor authentication solution must secure all the SaaS software and services. While most of the software and services support password authentication, not all of them provide support for two-factor authentication. Ensuring two-factor authentication in a SaaS model is a challenging task and requires all the software and services to be brought under one authentication policy.

Contents

List of Figures

List of Tables

1 Introduction

Content Management as a Service (CMaaS) is a term coined to mention that the traditional content management solutions can be operated in a cloud to offer a fully managed service. CMaaS like a SaaS model provides access to the remote web-based services. The services provide an access point normally in the form of a thin client or a web browser. The customers can access the SaaS services on the public internet and the services require high authentication security. Security and privacy are the two major concerns in cloud computing [SS11]. The data has to leave the boundaries of an organization in a cloud environment and requires high security. Also, the data may belong to federal agencies, banks and insurance organizations and is highly confidential. Any incident of loss or disclosure of cloud data can result in severe consequences and/or legal obligations.

IBM SmartCloud Content Management (SCCM) is a content management SaaS service. It stores the regulatory content in the cloud and provides the retrieval and search services via a web browser. IBM SCCM requires high security, as a variety of organizations may store their content in cloud. The content accessible through the SaaS services might be confidential and private. It requires a strong authentication mechanism, so that only the authorized entities can access the content.

Password authentication is the most commonly used and widely implemented single-factor authentication mechanism. The single-factor authentication mechanisms like passwords are vulnerable to many security threats, as further explained in section 1.1. On the other hand, two-factor authentication provides strength to the information security. It uses two authentication tokens during the authentication process. The tokens are bound together and both are simultaneously required for successful authentication. It is difficult to compromise two authentication tokens simultaneously as compared to compromising single authentication token. The two-factor authentication provides higher authentication security as compared to single-factor authentication.

Two-factor authentication also brings new challenges in the SaaS model. The different services combined in a SaaS model might offer different access points and authentication mechanisms. The services might also not support two-factor authentication. Some of these services might be licensed by a third party and do not allow changes in the software. The two-factor authentication solution must combine these services to provide a single access point and a generic authentication policy. Anyone using the SaaS services must be authenticated through a generic two-factor authentication mechanism, which is applicable to all the services.

The rest of this chapter explains why single-factor authentication is not enough and provides introduction to the two-factor authentication.

1.1 Password Are At The Edge Of Breaking Down

Password authentication is the most commonly used single-factor authentication mechanism. In this section, I analyse the shortcomings of a password authentication mechanism and argue that passwords are at the edge of breaking down, especially in the cloud environments.

A password is a secret shared between the claimant and the verifier. A claimant is one who claims to know the secret and the verifier is one who confirms or denies this claim. A password is bound to a unique user name and is known only to the claimant, apart from the verifier. The correct user name and password is provided to the verifier to access the secured resources.

Although password authentication is widely adopted because of its ease of use, the fact is, it is not the best authentication mechanism. Password authentication is vulnerable to brute-force attacks. Brute-force attacks can be divided into online and offline attacks. As the name suggests, different password combinations are tried online during the online brute-force attack. On the other hand, during an offline brute-force attack, the attacker gets access to the encrypted authentication key. He can then try different combinations at his leisure to discover the key. The password authentication is vulnerable to both brute-force attacks.

Password authentication has received criticism in the past for allowing users to choose weak passwords. Weak passwords are easy to crack and the ratio of successful brute-force attacks increases significantly. The studies have shown that a large number of users choose weak passwords. 22 percent of the user-chosen passwords could be easily recovered according to Columbia University research [1]. Bruce Schneier used a publicly available password recovery software and experimented that 55 percent of MySpace [2] passwords could be cracked within 8 hours [3]. The password recovery software he used was capable of testing 200,000 passwords per second. He further mentioned that "password1" was the most commonly used password among the users. These studies show ignorance and carelessness among users while selecting passwords.

Online brute force attacks can be avoided by blocking the user's account after a certain number of wrong authentication attempts. While this solution seems helpful in the beginning, it also brings some other problems. Disabling an account on multiple wrong authentication attempts opens door for the denial of service attacks. The attacker can easily disable the actual user from using its account by intentional false attempts. Also, blocking the user account will not help in resisting the offline brute-force attacks.

Recent security implementations and guidelines force the users to choose random passwords [4] [5] [egib]. The idea behind these guidelines is that randomness increases the password

[1] Columbia University
[2] MySpace
[3] Bruce Schneier; Real-World Passwords
[4] Microsoft Create Strong Passwords
[5] Bruce Schneier; Choosing Secure Passwords

strength. Strong passwords are those with higher randomness involved. The users are asked to use numerals along with capital and lower-case letters in the passwords to increase its randomness. They are encouraged to use long passwords with eight or more characters. Although most of the organizations force users to choose strong passwords, the fact is, the computational capability is also rising with the passage of time. Less time is required these days to crack a password with high speed computers.

Strong passwords are those with higher randomness involved. Entropy is the measure of randomness [WM10]. It is the uncertainty in guessing the password and is calculated in bits. Higher number of bits means higher entropy. Cracking an n-bits password is equivalent to the amount of effort required to find out a random number of n-bits. An n-bits random number can hold 2^n values and without any prior knowledge of the password, the attacker has to try all the 2^n values to crack the password.

Figure 1.1 shows the relation between entropy and password length with respect to different password choosing techniques. The passwords can be chosen randomly, using abbreviations, using phone numbers or using a combination of all these. For any given password length, a randomly chosen password has the highest entropy, as shown in figure 1.1. As an example, an English word has less entropy as compared to a completely random word of same length. It is because English words can not utilize complete randomness. After a certain number of characters, the next characters become obvious. On the other hand, an English word is easy to memorize as compared to a complete random number. It shows that the entropy of a password and the ease of memorizing it are competing characteristics.

A password can be made stronger by increasing its randomness. It is the only option to strengthen the password authentication. On the other hand, we believe that password randomness is already at its peak. We are using numbers, special characters, small and large alphabet letters in our passwords. Most of the organizations already require strong passwords with up to 8 characters length. If we force the users to choose a highly random password, it will be difficult for them to memorize it. They will be inclined to write it down somewhere, resulting in increased risk of inside attacks.

System generated passwords have high entropy values. At the same time, they are hard to memorize and more prone to be written down by users. A similar research was conducted by Zviran and Haga [YJ04] in which 106 students participated. The students were asked to choose a password of their own choice. Also, they were given a random password to remember. After a three months gap, the students were asked to recall both the chosen and random password. The results are given in table 1.1. The numbers in the table give insight in the user's behaviour of using passwords. Not only the users forgot random passwords more easily, most of them also wrote them down. The probability of writing down the passwords increased almost five times for random passwords. The results showed that forcing users to choose high entropy passwords will not help in future. Most of them will tend to write down their passwords and these written passwords will further pose security threats, instead of helping the authentication system.

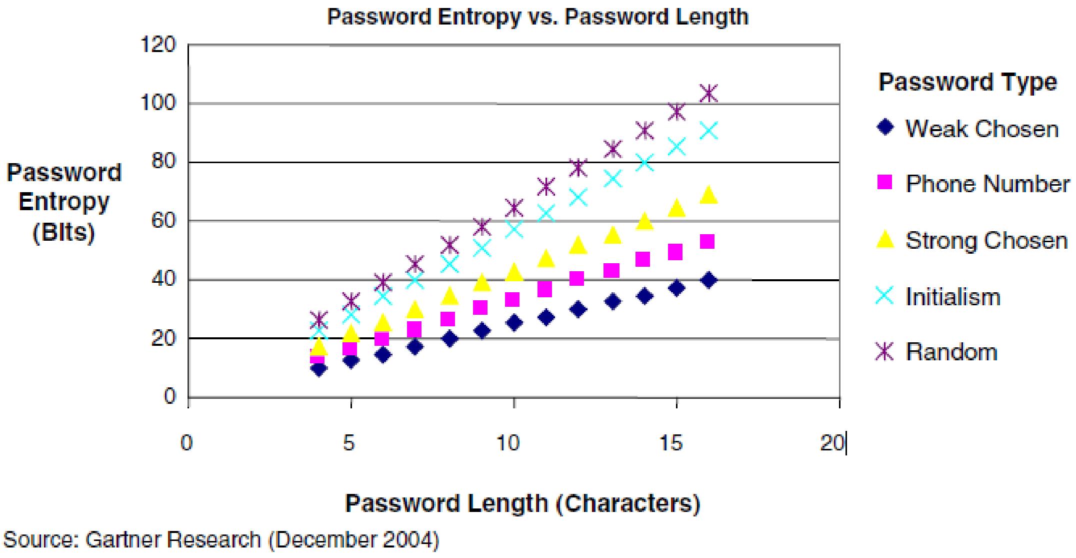

Figure 1.1: Password Length vs Password Entropy [All04]

After all this discussion, one can realize that passwords do not provide further room for strong authentication and any attempt to use passwords only to strengthen the security can result in collapse of the whole process.

1.2 Authentication and Authorization

Authentication is the process of establishing confidence in the identity of users or information systems [nisa]. It is the process to verify, if the entity is the one it claims to be. Authentication is the key to ensure security.

Authorization is the process of determining access rights to the secured resources. Authorization is done after successful authentication. It determines the level of access of a user for certain resources and makes sure that a certain entity can exactly do what it is supposed to do.

Authentication and authorization are two different concepts and should not be confused with each other.

1.3 The Factors of Authentication

The possible approach of authenticating someone falls into three categories [egia]. These categories are based on the factors of authentication used during the authentication process. There are three factors of authentication, as explained below.

	$Self-selected$	$Random$
Successful recall	35%	23%
Wrote it down	14%	66%

Table 1.1: Password Recall Survey

1.3.1 Proof of Knowledge Factor

Something a person *KNOWS* falls into this factor. A secret is shared between the user and the information security system. The shared secret is used as an authentication token. Some examples of the secret are password, PIN, personal question or a picture. The shared secret can be revealed to an attacker in case of a security breach.

1.3.2 Proof of Possession Factor

Something a person *HAS* is used as an authentication token during the authentication process. The possession factor can be a software or a hardware token. Some examples are smart cards, USB tokens and one-time password generators. The authentication token can be stolen or copied by an attacker in case of a security breach.

1.3.3 Proof of Characteristics Factor

Authentication tokens belonging to this factor are based on something a person *IS*. Each person has certain characteristics, which distinguish him from others. Biometrics like fingerprints, iris and retina patterns can be used as authentication tokens.

Table 1.2 provides some examples of authentication tokens related to the three authentication factors.

1.4 Two-Factor Authentication

Two-factor authentication is an approach to assert the identity of an entity using two factors of authentication. The two factors of authentication must be used together for successful authentication. Two-factor authentication binds the two tokens together to achieve higher security.

Multi-factor authentication mandates use of more than one factor of authentication. Two or more than two factors are used together during authentication. The security level provided by multi-factor authentication is supposed to be equal or higher than two-factor authentication.

Something A Person __			
Knows	*Has*	*Has*	*IS*
Password	Smart card	USB token	Fingerprints
PIN	Cryptographic key	OTP Generator	Retina
Secret question	TAN list	Java smart card	Iris
	SIM card	Scratch card	Face
	Driving license		Hand geometry

Table 1.2: The Factors of Authentication

Two-factor authentication is a stronger form of authentication. It uses tokens from two different factors. Tokens combined from different factors are resistant to a wider range of security threats. They cover vulnerabilities of each other and combine together to provide higher security level. A token belonging to the proof of knowledge factor can be read on wire and may suffer brute force attacks. On the other hand, a token belonging to the proof of possession factor is resistant to all these threats. An attacker must steal this token or make an exact copy of it.

A famous example of two-factor authentication is using an ATM card and a personal identification number (PIN). The ATM card is something a user has and the PIN number is something a user knows. The PIN can be revealed to an attacker by guessing, cracking or shoulder surfing attacks but ATM card is resistant to all these security threats. The attacker must steal the card or make an exact copy of it. Combining both authentication tokens increases the required range of attacks to gain simultaneous access to both tokens, thus enhancing security.

An authentication technique using multiple tokens from one factor is considered a multi-stage authentication and not a multi-factor authentication. An example of multi-stage authentication is a user providing a password and answering a secret question to authenticate itself. This technique uses two authentication tokens from the proof of knowledge factor and is not a multi-factor authentication.

While Multi-stage authentication might increase the security level, it can not gain advantages of multi-factor authentication. It is because the authentication tokens belong to only one factor and share the same security vulnerabilities. If one token is compromised during a security breach, the other one can also be compromised in result of it. The authentication tokens used in the previous example are password and a secret question. Both the tokens are vulnerable to same security threats like guessing, key-logging and shoulder surfing.

2 Motivation and Related Work

2.1 Importance of Security in Cloud Environments

Data security and privacy are the two major security concerns for the organizations to adopt cloud services [SS11]. The organizations are reluctant to store their content outside their own premises because of the exposed security threats. As organizations loose control over data in the cloud environments, they believe that the content stored in the cloud is more prone to security threats. A foolproof security plan must be provided to increase the level of trust between the cloud providers and the cloud consumers. The cloud providers must provide state of the art security solutions to establish the required level of trust. They have to prove scientifically that the data stored in the cloud is secure and only the authenticated and authorized personnel have the ability to access the cloud data. In my opinion, the Cloud industry can thrive rapidly if it takes solid actions to diminish security and privacy concerns of the organizations considering to use cloud services.

To understand the challenges in adopting cloud services, the International Data Corporation (IDC) conducted a survey on 244 IT executives and their line-of-business colleagues [Gen08]. They were asked to provide their views on usability of IT cloud services. Figure 2.1 shows their nine major concerns on adopting the cloud services. The survey results show that security is their top concern in adopting cloud services. The survey also implies that security issues in the cloud should be taken seriously by the cloud providers.

Cybercrime report for 2012 globally observed more than 1.5 million cybercrime victims each day [1]. In other words, 18 adults become victims of cybercrime each second. Global direct cost for these crimes was 110 billion United States (U.S.) dollars over the past 12 months. It is on average 197 U.S. dollars per victim. With such a massive amount of loss, it is becoming inevitable for organizations to adopt higher security solutions like two-factor authentication. It would help them reduce their financial costs, protect their private content and improve credibility.

The information security paradigm is based on five goals; availability, confidentiality, data integrity, authenticity, control and auditing [MZ10]. The cloud service providers should strive to accomplish these goals. Achieving these goals in an IT system is sufficient to mitigate the data security concerns. The first three goals (availability, integrity, confidentiality) are competitive in nature. An attempt to increase one of them might negatively effect others.

[1] Symantec Corporation, 2012 Norton Cybercrime Report, September 2012

For example, high confidentiality might negatively effect the availability of the information system and vice versa.

Availability makes sure the data is available to the consumer at any time and place. The computing systems and communication channels should function correctly to ensure availability. Security tightening has normally a negative effect on availability.

Confidentiality ensures the data stored in cloud is kept secret. Data encryption and physical isolation can be helpful to achieve confidentiality.

Data integrity is another well known information security goal. The data uploaded to the cloud must not be altered or lost by any means. It requires successfully uploading the data in cloud and retrieving the same data later.

Authenticity ensures that the data communicated is authentic and the communicating parties are who they claim to be. A secure and well-managed authentication mechanism is required to fulfil this security goal.

Control regulates the cloud applications and infrastructure. It makes sure they all work together to fulfil the required security standards. Control is important for the cloud, as it is a combination of different distributed machines.

Audit means to keep track of ongoing events in a cloud environment. Auditing can be done by logging events and monitoring the system. Only the functional aspects of cloud should be monitored from a cloud provider and might require a prior agreement with the customer.

Lack of one or more of the above mentioned information security goals can result in a security threat. Necessary measures should be taken to mitigate the security threats and a security plan should be designed and implemented to achieve the information security goals.

The authentication mechanism plays a vital role in security enhancement. Authentication mechanism is like an entrance door and will allow only the trusted individuals to enter in the cloud premises. The mechanism should be robust enough to ensure availability by letting the right person in, any time and any place. At the same time, it must ensure confidentiality. Authentication mechanism can be combined with cryptographic techniques to ensure confidentiality of data. Data integrity can also be ensured if only authenticated persons can access the cloud services and proper encryption is done while transferring data. Having the best possible authentication mechanism along with a complete security plan can mitigate most of the security concerns of cloud consumers.

2.2 Security Threats in Cloud

Security threats related to the authentication process are discussed below. These threats are discussed with respect to internet and cloud scenarios and include security risks related to the remote authentication process.

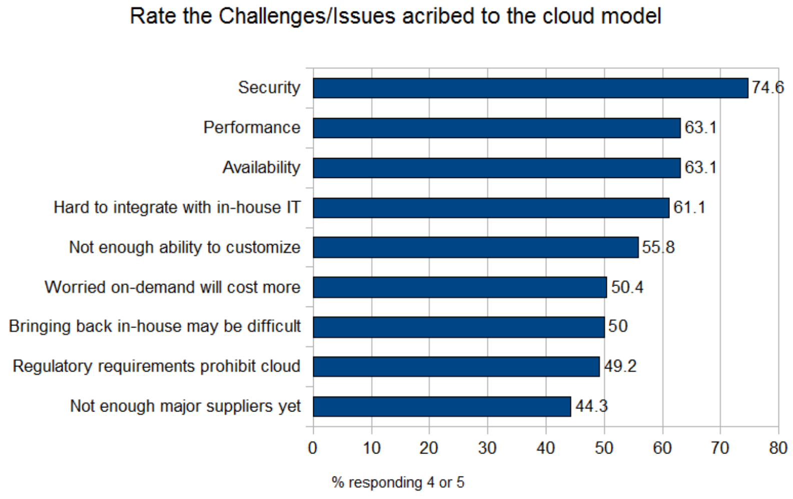

Figure 2.1: Top 9 Concerns of Organizations in Using Cloud Services [Gen08]

2.2.1 Replay

In a replay attack, the attacker records the data communicated during a successful authentication attempt. The attacker can then replay the recorded data and authenticate itself any time later.

Replay attacks are possible if the user and verifier repeatedly transmit the same authentication data and token during the authentication process. Replay attacks can be avoided by including nonce or time stamps during authentication. Session ID can also be used to avoid this problem. Session ID helps in the identification of the current session. The verifier sends a one-time secret to the user. The user hashes the password using this secret and sends it to the verifier. The verifier compares the incoming hash value with the one it calculated locally. If both the values match, the user is authenticated. The verifier sends a new one-time secret each time to the user and replay attacks can be resisted.

2.2.2 Eavesdropping

Eavesdropping is the act of listening the secret information between two authorized parties. The attacker can use the listened information to learn the secret. Eavesdropping can be done on internet, telephone and wireless communication channels.

2.2.3 Session hijacking

An attacker waits until the user authenticates against the verifier. Once the communication session is established, the attacker takes over the session.

Session hijacking attacks can occur even if good security measures are taken during the authentication process. These attacks take advantage of the fact that enough security was not ensured after the completion of the authentication process. Some session hijacking techniques are discussed below.

A session ID is issued to the user in the beginning of the communication session. The user uses the session ID during the rest of session. If the session ID is not protected properly, an attacker can read it on the wire and exploit the session. This session hijacking technique can be mitigated by completely encrypting the communication between the client and the server. The intruder will no be able to read the session ID, as the communication channel is encrypted.

Session hijacking is also possible, if an attacker has physical access to one of the machines taking part in the communication. The session key can be stolen from the machine. Later, it can be used to communicate as an authenticated user.

Another session hijacking technique is called session fixation. In this technique, the attacker manipulates the session ID to his own desire. Once the session ID of his own desire is set, he can successfully communicate with the server.

Yet another possibility of session hijacking is to guess the session ID. Long and random session IDs are used to resist an attacker from guessing the session ID.

2.2.4 Phishing/ Verifier Impersonation

A phishing attacker traps the user through forged electronic communication. The victim provides its secrets to the attacker, considering it as the actual verifier. While the victim believes it is communicating with a trustworthy entity, it is actually communicating to an attacker. Phishing attacks can be avoided by user awareness and continuous training.

Phishing attacks can be performed using fake websites, forged emails, fraudulent telephone calls, or by exploiting other electronic communication channels. In the case of a fake website, the attacker constructs a website similar to the actual one. A naive user does not understand the trap and reveals the secrets (password etc.) on the fake website. Other possibility of phishing is to send the victim a forged email and prompt him to reveal the secrets.

Phishing has always caused a lot of trouble for the organizations. It has caused them financial loss and reputation degradation. On the other hand, repeated phishing in the past has significantly increased the rate of denial of service attacks. They web users, these days, put little trust in the emails because of the potential phishing attack. The emails sent by the organizations are ignored by the web users, thus forcing the organizations to adopt more costly ways of communication.

2.2.5 Man-in-the-middle

In a man-in-the-middle (MITM) attack, the attacker places itself in the middle of a communication channel and poses itself as a valid verifier to the user and as a valid user to the verifier.

Figure 2.2 is a graphical representation of a man-in-the-middle attack. The legitimate communication channel is broken and is shown as a disconnected black line. The attacker establishes a new connection between the user and the verifier by placing itself in the middle of it. A MITM attack scenario is explained below.

As the communication between the client and server starts, the server sends an encryption key to the client. The encryption key is used to encrypt the messages exchanged between the client and server. Before the encryption key can reach the client, an attacker comes in and interrupts the communication channel. It replaces the encryption key sent by the server with an encryption key of its own. The user receives the encryption key from an attacker, encrypts the authentication token using the attacker's encryption key and sends it back. The attacker can now decrypt the data received from the user, as it owns the encryption key. It decrypts the authentication token, encrypts it with the encryption key sent by the server and sends it back to the server for authentication. From now on, the attacker can communicate with both the client and server without their knowledge. It can read or modify the communication data between the client and the server with out their knowledge.

The attack discussed in the above scenario is possible because the encryption key sent from the verifier was not digitally signed. Digital signatures ensure properties like integrity, authenticity and non-repudiation of the encryption key. The TLS protocol is resistant to MITM attacks, as the encryption key (public key certificate) sent by the verifier is digitally signed and can not be replaced with a fake encryption key.

2.2.6 Customer Fraud Attacks

Customer fraud is a special type of attack where the web client deliberately compromises its authentication token. This can be done to take personal advantages or to defame an organization. To prevent this attack, the verifier must be able to prove that the authentication failure was the victim's own fault.

2.2.7 Insider Attacks

During an insider attack, someone inside the security perimeter deliberately compromises the security. The inside attackers pose big threat to the organization's security. They can create opportunities for the attackers by deliberately revealing the internal authentication process and secret tokens.

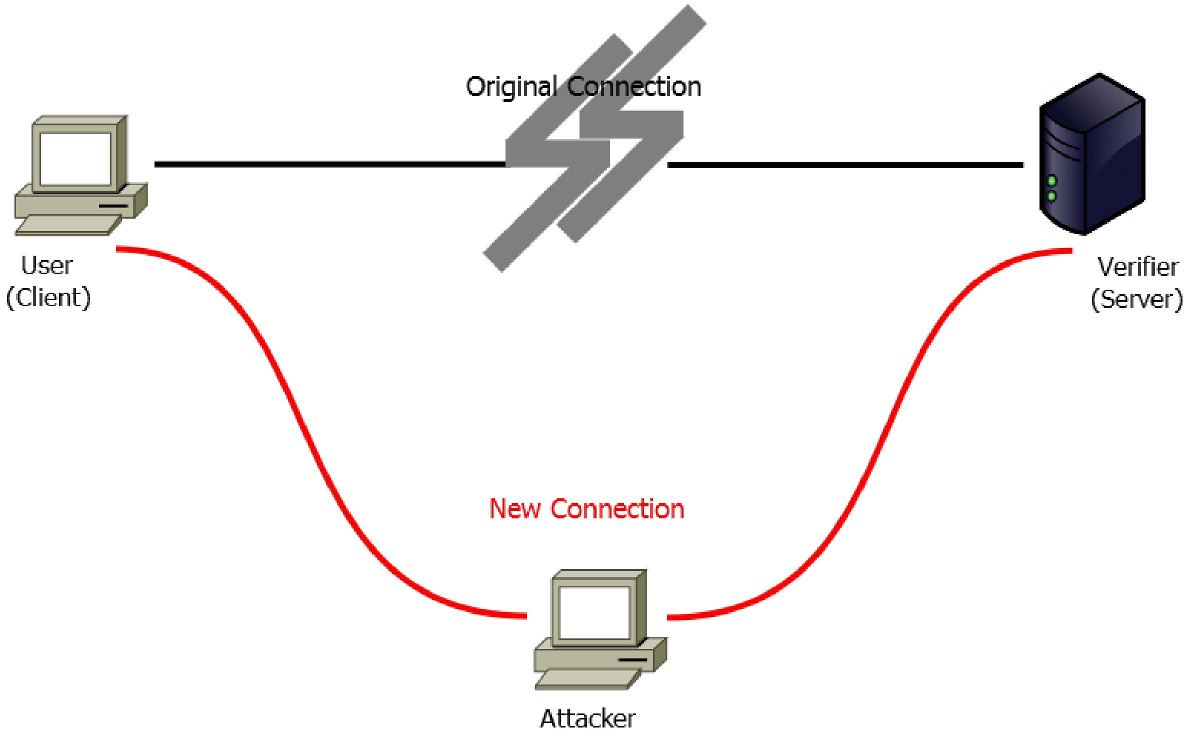

Figure 2.2: Man-in-the-middle Attack

2.2.8 Malware

Malware is a general term used for a piece of software trying to compromise the authentication process. Malwares try to collect confidential information and interfere in the authentication process using it. A malware can be a keylogger, trojan, computer virus or spyware etc.

Keyloggers are installed on a computing device and store the history of pressed keys in a log file. Secrets like passwords can then easily be extracted from the log files.

Trojan horse are 70% of the total malwares [2]. Trojans represent themselves as a useful piece of code and prompt the user to install them. Once installed, they can create security threats for the computing system.

2.2.9 Password Discovery Attacks

Password discovery attack is the process of illegal recovery of the passwords. Brute-force and dictionary attacks are two famous password discovery methods.

[2] Malware Statistics

Brute-force method tries all possible key combinations. This method is adopted by attackers when there is not much information available about the password key. Dictionary attack is a more sophisticated method of discovering password. It recovers the password by trying likely possible key combinations. Attackers build large dictionaries of likely possible key combinations and keep these dictionaries up to date from time to time. Dictionary attacks have higher success ratio, as a large number of users choose simple passwords like password1, password1234 etc. [3].

Password discovery attacks can be further categorized in online and offline discovery attacks. During an online attack, an attacker tries different password combinations to authenticate itself against an online web server. Online attacks can be detected and prevented by the server. Limiting the number of unsuccessful authentication attempts or putting certain delay period after an unsuccessful attempt are possible solutions to prevent offline attacks. While these solutions resist the online guessing attacks, they also provide opportunities for denial of service attacks. An attacker can deliberately make wrong authentication attempts and block the victim from using the IT services.

In the case of an offline discovery attack, the attacker has access to the encrypted data of a successful authentication attempt. He can try different key combinations locally in his leisure time to decrypt the data and extract the password. The victim and the verifier remain ignorant during this whole process. Strong passwords should be chosen to resist offline attacks.

2.2.10 Shoulder Surfing

An attacker tries to get the secret information (pin, password) as the victim enters it on the keyboard. Even a partially successful shoulder surfing attack can be dangerous when used with other security threat combinations. For example, getting information about the password length by shoulder surfing attack can become very handy for a password discovery attack.

2.2.11 Social Engineering Attacks

Social engineering attackers achieve certain level of trust, to persuade the victim to expose the secret. Social engineering attackers can use internet, telephonic conversation or even one to one conversation with the victim. An example is a colleague of the victim, tricking him to reveal the password as it is required by the higher management. Similarly, an attacker can trick the help desk guys to reveal some internal information to him during a fake telephonic conversation. Phishing is also considered as social engineering as the attacker achieves certain level of trust before performing the attack. In many situations, a social engineering attack creates ground for a more sophisticated security attacks.

[3] Bruce Schneier; Real-World Passwords

2.3 Success Criterion of an Authentication Solution

Many authentication solutions are available these days. The success of an authentication solution is determined by its technical and non-technical adoption feasibility. Also, the acceptance level of a particular solution is determined by the solution providers as well as by its users. The goal is to provide a solution which can achieve all the success criterion. Some major criterion deciding the success profile of an authentication system are explained below.

2.3.1 Customer Acceptance

Customer acceptance plays an important role in the success or failure of an authentication solution. Customer acceptance is indicated by the required skill level and the required hardware or software support.

The required skill level to perform the authentication process is an important success indicator. User unfriendly authentication processes require client education and result in higher cost. On the other hand, simplistic solutions might not provide required security level. A successful authentication solution should simultaneously provide high security level and ease of use.

Software and hardware requirements of an authentication system are the major acceptance indicators. Smart card authentication can only be used by the clients with the smart card reader installed on their machines. Similarly, the solutions using hardware token generator mandates the client to purchase a hardware token generator.

2.3.2 Token Management Difficulty

The success of an authentication solution highly depends on the creation, distribution and management of the authentication tokens. Some tokens are easy to manage than others. In a trivial solution like passwords, the users can create a password remotely and there is no need to distribute the tokens.

2.3.3 Credential Replacement

Authentication tokens expire after certain period of time and must be replaced. Also, the compromised tokens should be revoked and replaced with the new ones. A successful authentication solution should allow smooth replacement of tokens on large scale as well as on small scale.

2.3.4 System Costs

Cost is a major decision factor for customers to adopt a security solution. The cost should be calculated for the entire estimated life cycle of the authentication solution. Some solutions incur high acquisition cost and low maintenance cost while others may incur low acquisition cost and high maintenance cost. A solution with higher acquisition cost is not necessarily a costly solution. In the long run, it might be cheaper than the one incurring low acquisition cost and high maintenance.

2.4 Related Work

Many governments and industry-specific bodies have created guidelines for two-factor authentication. They realize the importance of two-factor authentication and consider it as a basic security requirement.

Federal Financial Institutions Examination Council (FFIEC) is a US government regulation body and mandates all the banks and financial institutions in the US to implement strong two-factor authentication [AB13]. The Government Chief Information Officer (GCIO) in an organization which provides leadership in information and communications technology (ICT) matters to the New Zealand government. It provides guidelines for multi-factor authentication and encourages the information technology systems to adopt it [egia].

Industry specific regulations also consider two-factor authentication as a security requirement. Payment Card Industry Data Security Standard (PCI DSS) in its requirements include implementation of two-factor authentication [pci]. Health Insurance Portability and Accountability Act (HIPAA) and Health Information Technology for Economic and Clinical Health (HITECH) also mandate organizations to implement two-factor authentication before giving remote access to protected health information [hip]. The Internet Banking and Technology Risk Management (IBTRM) dictates the banks to implement two-factor authentication for high value transactions or change to sensitive data [ibt].

Table 2.1 summarizes some famous web services implementing two-factor authentication. Most of these web services offer two-factor authentication to register a new device. Once a device is registered, the web service authentication is switched to single-factor authentication.

Google uses cookies to register new devices. The client logs in from a new device using two-factor authentication. From now on, cookies are stored on the device and are communicated between the client and server. The web server will require two-factor authentication again only if the cookies are removed from a registered device or if a new workstation is used.

WebServices	1*stAuthenticationFactor*	2*ndAuthenticationFactor*
Amazon Web Services	Password	Hardware token generator OR Software token generator
Google	Password	Software token generator OR Message sent to mobile phone
Facebook	Password	Software token generator OR Message sent to mobile phone
PayPal/eBay	Password	Hardware token generator OR Text Message sent to mobile phone
Dropbox	Password	Software token generator OR Message sent to mobile phone

Table 2.1: Two-Factor Implementations Provided by Different Web Services

3 Authentication in Cloud Environment

While data is turning into electronic form, many organizations tend to store their data in the content management cloud services. The security requirements have changed dramatically for the electronic content stored in the cloud. The content is sent on the wire and travels on the internet. It can be modified, copied, or lost on its way. Also, the content resides on distributed machines, which are connected through the internet. The cloud consumers have little control over the data stored in the cloud. They are completely dependent on the cloud providers to ensure high security measures in protecting the cloud content.

Security measures must be taken by the cloud providers while dealing with confidential data. Efforts have been made to standardize the security of information systems. The security standards define the assurance level required by a specific information system. If the cloud providers want to store sensitive information in their cloud, they are supposed to meet the assurance level required by the information system.

This chapter describes the complete process of selecting an authentication solution. The authentication requirements of an information system are set by the security threats exposed to it. The security threats are the sources of risk involved in a security breach. The authentication solution tries to mitigate the risk by achieving an assurance level. The assurance level of an authentication process is an indicator of its strength. The authentication process with higher assurance level provides stronger security. There is also a subtle difference in the underlying technologies adopted by different authentication solutions. Certain technologies are resistant to certain security threats. An authentication solution should select the technologies which are resistant to the exposed security threats. The technologies involved in the authentication process and their pros and cons are also discussed in this chapter.

3.1 The Authentication Process Model

The security of the cloud applications heavily depends on a strong authentication mechanism. Authentication is the process of establishing confidence in the user identity, electronically presented to an information system [nisa]. The entities which meet a certain level of confidence are given access to the cloud. The authentication process is standardized and adopted by a large number of countries and industry related organizations.

The authentication related terminologies and the authentication process is described in this section below.

Subscriber requests for a token. It receives a token from the credential service provider, which it can use later for authentication. The subscriber is granted a token, once required information is provided by him.

Registration Authority (RA) is a trusted entity which collects credentials from subscribers. It further verifies the collected credentials and vouches for the identity of a subscriber to the credential service provider. Registration authority might be a separate entity or an integral part of the credential service provider.

Credential Service Provider (CSP) registers the subscriber and if required, issues the token to a subscriber. For example, in password authentication scenario, the credential service provider registers the subscriber but does not issue a token. In more complex scenarios, it also issues an authentication token to the subscriber.

Claimant claims to keep a token. The claimant wants to access certain resources on the basis of its claim.

Verifier verifies the claimant's identity and the token presented to it by the claimant. Verifier passes on its results to the relying party.

Relying Party relies on the verifier's assertion of claimant's identity and grants access to the resources or performs a transaction.

Authentication process has two main phases [stoa]. These are the registration phase and the authentication phase. The complete authentication process along with its two phases is shown in figure 3.1.

1. **Registration Phase:** Identity proofing is the first step of registration phase. The subscriber proves its identity by providing identity attributes to the registration authority. Registration authority collects the attributes and verifies the identity of the subscriber. Depending on the process chosen, identity proofing can be as simple as collecting the first and last name of the subscriber. For some more complex scenarios, it can be a complete verification of the user attributes including subscriber's national identity number and photograph.

   Once subscriber identity verification is complete, registration authority passes on the verification results to the credential service provider (CSP). CSP registers the user in its database and if required, generates a token. The subscriber is handed over the token, which he can use for authentication. Also, electronic credentials are issued to the subscriber to bind the token or subscriber attributes to its identity.

2. **Authentication Phase:** During the authentication phase, the claimant presents its token and/or attributes to the verifier. It has to prove the ownership and control over the token. Proof of ownership does not necessarily mean the claimant possesses the token. The token may also belong to the knowledge or characteristics factor. The claimant uses the pre-defined communication protocol to transfer the token to the verifier.

   The verifier interacts with the CSP to verify the token or credential presented. It then makes decision about the validity of the token and informs the relying party about the

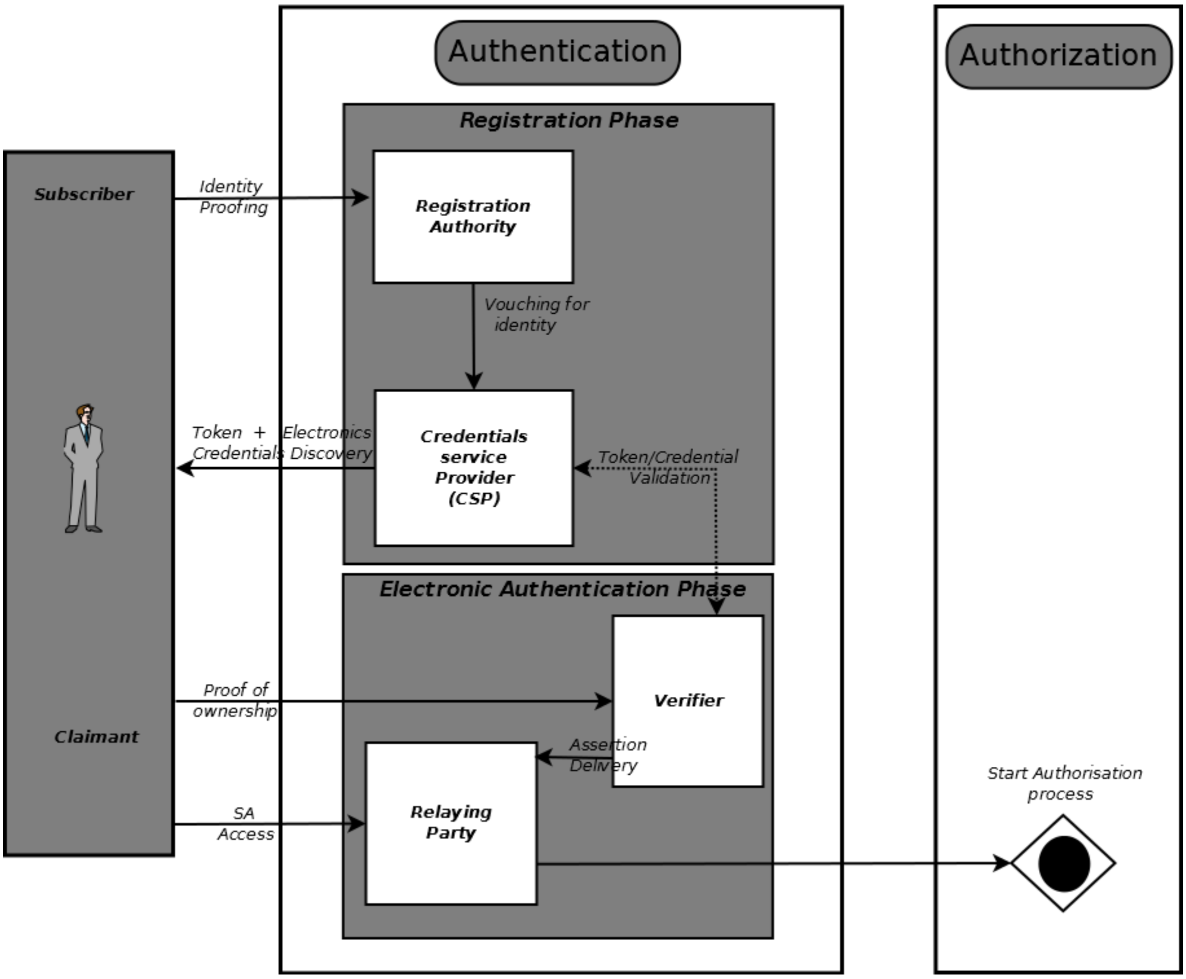

Figure 3.1: Authentication Process Model [stoa]

decision. Depending on this, the relaying party allows or denies access to the resources or a requested transaction.

3.2 Security Assurance Level

The assurance level is defined as the degree of confidence in the registration and authentication phase of an authentication process model. The assurance level is an indicator of the security strength of an information system.

The assurance levels are categorized as [omb];

Level 1: Little or no confidence in the asserted identity's validity.

Level 2: Some confidence in the asserted identity's validity.

Level 3: High confidence in the asserted identity's validity.

Level 4: Very high confidence in the asserted identity's validity.

The assurance levels are related to the potential impact of a security breach. Potential impact is the measurement of the loss due to the compromise on the security goals. The loss is defined as damage, reduction or deprivation of a desirable entity. Authenticity, availability, integrity, non-repudiation and confidentiality are the goals of an information system security. A compromise on the mentioned goals can result in financial loss, material loss, reputation loss or any other type of loss.

The assurance levels categorize the required security strength, depending on the potential impact of a security breach on an information system. The lower assurance levels ensure less security while higher levels ensure higher security.

Potential impact with respect to the severity of loss is classified as follows [fipb]:

No impact: The impact is considered not applicable, if a compromise on the security goals is supposed to result in no loss.

Low impact: The impact is considered low, if a compromise on the security goals is supposed to result in limited loss.

Moderate impact: The impact is moderate, if a compromise on the security goals is supposed to result in serious loss.

High impact: The impact is considered high, if a compromise on the security goals is supposed to result in severe loss.

Apart form defining the potential impact categories, it is important to define the areas where loss can take place. Following are the areas, where a security compromise might have a potential impact [omb]:

- Inconvenience or damage to reputation of the organization
- Financial loss
- Harm to organization's programs or public interests
- Unauthorized release of sensitive information
- Personal safety

- Civil or criminal violations.

A security compromise can impact any of the above mentioned areas. Also, the same impact on two different areas can result in different consequences and cannot be compared. For example, a low potential impact on personal safety might be more devastating than a moderate impact on financial assets.

Table 3.1 categorizes the assurance levels with respect to potential impact on different areas. It provides the guidelines to select an assurance level depending on the potential impact on different areas of an information system. It includes three different parameters; assurance levels, potential impact areas and risk impact levels.

The *low* impact level of inconvenience is covered by assurance level 1, while its *moderate* impact level is covered by assurance levels 2 and 3. Also, *high* impact level of inconvenience is covered by level 4. It simply means that assurance level 1 should be selected and implemented if a security breach results in *low* inconvenience. Similarly, assurance level 2 and 3 should be implemented if a security breach results in *moderate* inconvenience and assurance level 4 should be implemented in case of *high* inconvenience. The similar criteria is adopted for the rest of potential impact areas and an assurance level is selected.

In another example, the organization should choose assurance level 1, if a security breach results in *low* impact on first three potential impact areas and no impact on the rest of the potential impact areas. Level 1 should not be used if there is a *low* impact involved in categories like personal safety and unauthorized disclosure to sensitive information. It is because these categories have severe consequences and require higher security level.

3.3 Information Security Process

Now that I have mentioned details about assurance levels, it is time to go into details of information security process. Ensuring information security in organizations is a step by step process. E-governments and independent security bodies provide guidelines to select, implement and manage a security solution in organizations.

E-authentication Guidance for Federal Agencies [omb] provides guidelines for an information security process. The process starts by assessing risks involved in the information system. The identified risks are then mapped to the assurance levels and an assurance level is selected. The organizations then strive to achieve the assurance level by implementing suitable technologies. The information security authentication solution is then validated and monitored to recognize any change in the security needs. The information security steps are described below:

- Conduct a risk assessment to identify risks.
- Map identified risks to the applicable assurance level. Select an assurance level.
- Select technologies which promise to ensure the assurance level and identified risks.

$PotentialImpactAreas$	$Level1$	$Level2$	$Level3$	$Level4$
Inconvenience	Low Impact	Moderate Impact	Moderate Impact	High Impact
Damage to reputation of the organization	Low Impact	Moderate Impact	Moderate Impact	High Impact
Financial loss	Low Impact	Moderate Impact	Moderate Impact	High Impact
Harm to organization's programs or public interests	No Impact	Low Impact	Moderate Impact	High Impact
Unauthorized disclosure of sensitive information	No Impact	Low Impact	Moderate Impact	High Impact
Personal safety	No Impact	No Impact	Low Impact	High Impact
Civil or criminal violation	No Impact	Low Impact	Moderate Impact	High Impact

Table 3.1: Tolerable Impact in Different Areas with respect to Assurance Levels [omb]

- Validate the security system. Make sure it has achieved the expected results.
- Keep on assessing the security solution periodically and determine if some modifications are required.

The rest of the chapter sequentially covers these information security steps, by providing deep insight into the information security process.

3.4 Conduct Risk Assessment

Risk assessment is the first step of information security process. It lays ground for upcoming security related decisions and also plays a decisive role in technology selection.

Risk assessment is a part of the risk management. Risk management is performed to minimize the effect of risk and achieve higher security standards. The four components of risk management model are shown in figure 3.2. Following is a brief description of all four components with main focus on risk assessment.

3.4.1 Framing Risk

Risk framing creates context for the rest of the risk management process. It provides the basis for next risk management components. As shown in figure 3.2, it also takes feedback from the other three risk management components and can react accordingly.

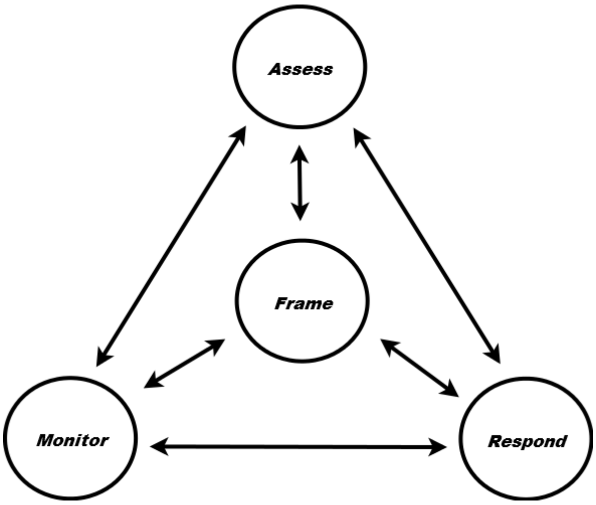

Figure 3.2: Risk Management Process [nisb]

Risk framing establishes an environment, in which risk based decisions can be made. It identifies certain assumptions and constraints which play important role in risk assessment, response and monitoring. Organizational decisions are made in risk framing and aspects like finance, business impact and legal issues are also addressed here.

Tolerance to risk and organizational priorities are also set during risk framing [nisd]. The organization can decide to tolerate the risk if the impact is not very high. For example, organization can decide to tolerate the risk and take no actions, if the financial loss due to a certain risk is less than a few hundred dollars. It is because mitigating the risk might be far more expensive than its actual impact.

3.4.2 Assessing Risk

Risk assessment identifies potential risks and their likelihood of occurrence [omb]. Context and constraints, identified by risk framing, are used to assess the risk or harm.

Following are the major tasks during risk assessment [nisb].

- Define assets and their values.
- Define threats involved and their vulnerabilities.
- Define likelihood of an attack.
- Define harm in case of a successful attack.

Figure 3.3 represents a model for risk assessment. The model is largely divided in two steps. Preparing for the risk assessment is the first step. It takes input from risk framing and uses it to define the goals of risk assessment. Organizational architecture and time frame to cope with the risk is also identified during this step. It is important to set an effective time, as the risk changes with time. Keeping time constraints into account is important for the risk assessment process.

Organizational assets and their values are also identified during the first step of risk assessment. While some assets have higher values and others are less valuable, it is important to identify and categorize them. The assets with higher value should be given more importance later during the risk mitigation stage. The asset value can be calculated by taking into account the outcome of its unavailability. "How the increase in unavailability time of a certain asset will affect the organization?"can further give a deep insight in the asset value calculation.

Second step of risk assessment starts by conducting the risk assessment. Threats and their sources are identified in the beginning. A threat is an event, having the tendency to impact organizational assets via unauthorized access to the information system [nisb]. Threats can be originated by different sources. They can be originated by a pre-planned and well-defined cyber-attack. Physical attacks can also result threats. An ignorant individual can create threats for an organization and human errors can also become the threat originators.

Once all the threats sources are recognized, it is time to identify system vulnerability. Vulnerability is another term for weakness. Any kind of weakness makes the system vulnerable. Security system weakness occurs due to ignoring a security measure or an improper/partial implementation of a security measure. Vulnerabilities may rise because of changing risks with time and not acting timely to those risks.

System vulnerability has a strong connection with the threats exposed to the system. The threats exploit the weaknesses in a system and try to attack the weaker part of it. Predisposing conditions affect the likelihood or severity of a threat and should also be identified. The predisposing conditions can increase or decrease the impact of a threat.

Once threats and system vulnerabilities are identified, it is time to determine the likelihood of threat occurrence. A probabilistic analysis is made to calculate the likelihood of threat. Historic data is normally required to perform an analysis. Organizations keep record of previous threats or collect this information from a trusted party. The accuracy of the collected information is important for an accurate analysis. If no data is available, assumptions can be made about the likelihood of threat occurrence.

Next step in risk assessment is to calculate the impact of a threat. Impact is defined as the magnitude of harm. The harm can occur due to the loss of availability, authenticity, integrity, or confidentiality of an information system.

After identifying the likelihood of threat occurrence and the related potential impacts, it is time to calculate the risk involved in an information system. Risk is a function of likelihood of threat occurrence and its impact.

$$Risk = Likelihood * Impact$$

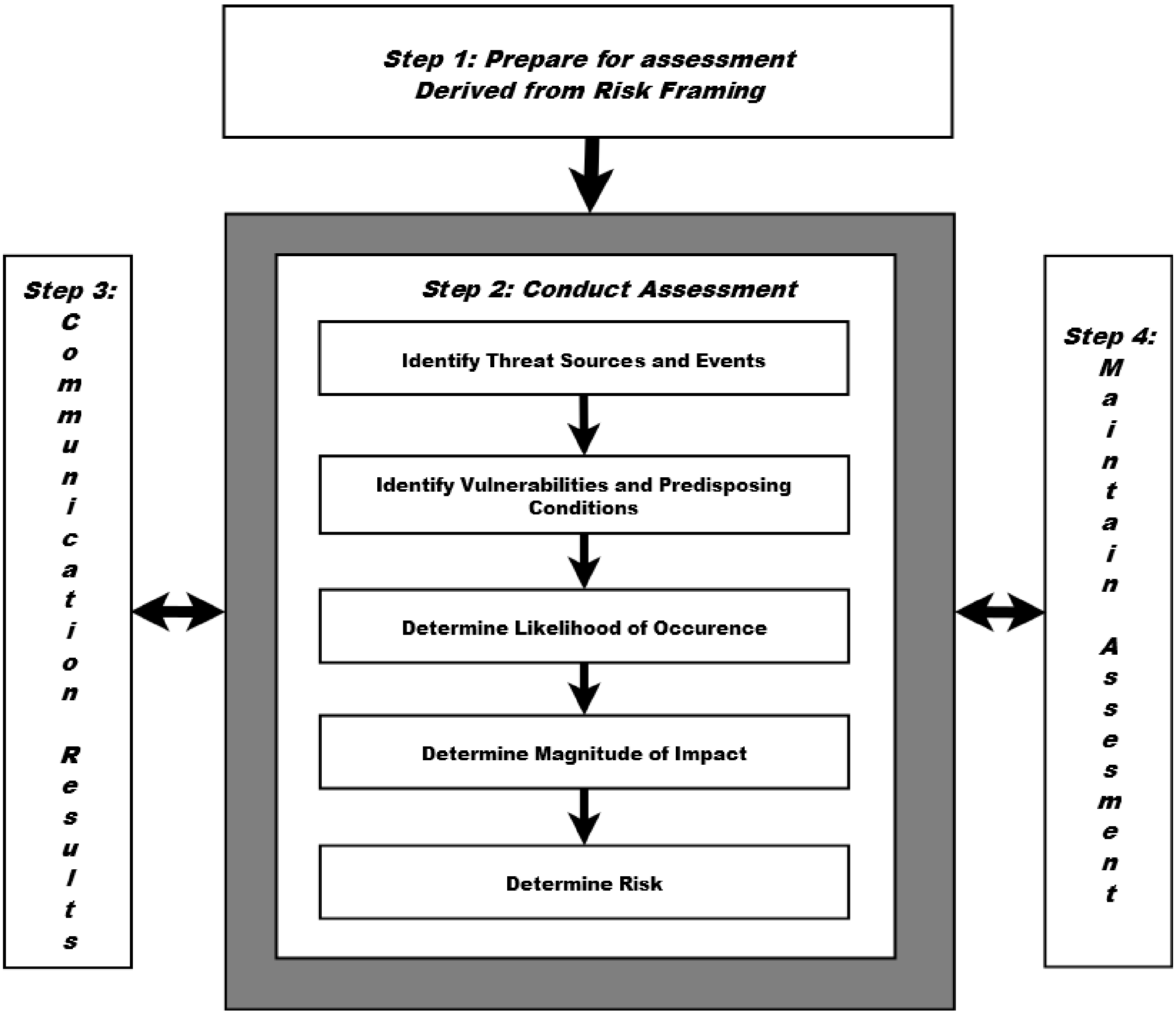
Step 1: Prepare for assessment
Derived from Risk Framing
Step 2: Conduct Assessment
Identify Threat Sources and Events
Identify Vulnerabilities and Predisposing Conditions
Determine Likelihood of Occurence
Determine Magnitude of Impact
Determine Risk
Step 3: Communication Results
Step 4: Maintain Assesment

Figure 3.3: Risk Assessment Process[nisb]

Security Identity Across Borders Linked (STORK) is an European electronic identity interoperability platform. It provides a standard template for risk analysis [stob] as shown in table 3.2, 3.3 and 3.4. The standard defines 5 levels of likelihood of threat occurrence and 3 impact levels. Likelihood of threats is partitioned in levels depending upon their probability of occurrence. The likelihood levels belong to a certain range of threat occurrence probability.

The STORK standard template further combines different impact levels and likelihood levels to categorize the risk. Risk is categorized into 5 levels and is a function of threat likelihood and corresponding impact. Table 3.4 shows that a threat with high impact (H) and low chance of occurrence (L) will result in medium (M) risk. Similarly, a threat with low impact (L) and very high likelihood of occurrence (HH) can impose a high risk (H).

LikelihoodofThreatOccurrence	*Description*
Very High (HH)	Likelihood of threat occurrence is more than 80%
High (H)	Likelihood of threat occurrence is more than 60% and less than 80%
Medium (M)	Likelihood of threat occurrence is more that 40% and less that 60%
Low (L)	Likelihood of threat occurrence is more than 20% and less that 40%
Very Low (LL)	Likelihood of threat occurrence is less than 20%

Table 3.2: Likelihood of Threat Occurrence [stob]

ImpactLevel	*Description*
High (H)	Impact can highly effect the information system
Medium (M)	Impact has medium effect on the information system
Low (L)	Impact has low effect on the information system

Table 3.3: Impact Levels [stob]

		Impact		
		H	M	L
Chance	HH	HH	HH	H
	H	HH	H	M
	M	H	M	L
	L	M	L	LL
	LL	L	LL	LL

Table 3.4: Risk Calculation [stob]

Once risk assessment is complete, the results are communicated within the organization. The purpose is to synchronize the security task force and make coherent decisions throughout the organization.

3.4.3 Responding to Risk

Risk assessment provides input to this step. By this time, organizations are well aware of the type and level of the exposed risk. They can decide on what to do with it. There are multiple options available to respond to risk [nisd]:

Accept risk: Risk can be accepted if it is within the tolerance range. This option is normally opted if risk assessment yields low risk values.

Avoid risk: Risk avoidance becomes inevitable if the risk level is higher than the acceptance level. Risk must be avoided in this case. Threat mitigating action are performed and security measures are implemented to avoid risk.

Reduce risk: Risk reduction is also a valid option to respond to risk. It is an attempt to reduce risk level. It is a compromise between avoiding and accepting risk. Only that portion of risk which is above the tolerance threshold is mitigated.

Share or Transfer risk: In this case, the organization can outsource the risk liability to another party and concentrate on its core business process.

Combination of above: Combination of more than one of the above mentioned options is used to respond to risk.

Necessary actions must be taken if organizations want to avoid or mitigate risk. Apart from structural decisions, organizations must choose the right technology. Selecting the right technology is discussed in detail in section 3.6.

3.4.4 Monitor Risk

As risk changes over time, it is necessary to monitor it from time to time. The frequency of risk monitoring depends on the rate of change of risk. Risk monitoring verifies the implementation of security measures and confirms their effectiveness. Risk monitoring can also provide input to risk framing, risk assessment and risk monitoring.

3.5 Map Identified Risks to the Applicable Assurance Level

So far I have identified the risks and also described the assurance levels. Now, I map the identified risks to the assurance levels. Once the mapping is complete, an assurance level is selected accordingly. The assurance level chosen at this stage plays fundamental role in selecting the technologies for the authentication process. Only those technologies are selected and implemented, which meet the selected assurance level.

First of all, I categorize the identified risks in section 3.4 to the risk categories of table 3.1. The risk categories from table 3.1 are mentioned below:

1. Inconvenience
2. Damage to reputation of the organization
3. Financial loss
4. Harm to organization's programs or public interests
5. Unauthorized release of sensitive information
6. Personal safety
7. Civil or criminal violations

Once risk categorization is complete, I find the impact profile of each category. The impact profile of each risk category is mapped to table 3.1 assurance levels and one assurance level is selected. Here, it is important to mention that, the assurance level selected should be the one capable of assuring the highest impact level. For example, consider a scenario where first six risk categories of table 3.1 hold low value but the last risk category (civil or criminal violation) holds moderate value. The assurance level 3 must be selected in this case, even if the first six risk categories have low impact.

A Scenario: In this section, I assume a scenario to explain the identified risks mapping to select an appropriate assurance level.

Assume the data stored in the cloud belongs to a bank. The bank decides to store the record of money transactions in a cloud. Data related to the remote authentication of bank's customers (customer password, TAN list) is not stored in the cloud. A security breach in the cloud or exposure of bank's data to unauthorized users can cause high impact on inconvenience and harm to bank's reputation. At the same time, it might not result in high financial loss. Impact on different risk categories is this scenario is mentioned in table 3.5.

It is important to mention here that the impact profile levels for risk categories in table 3.5 are mentioned to illustrate the concept. One can disagree with the writer's point of view on the impact profile levels.

As it is seen form table 3.5, the impact level for inconvenience, damage to reputation and unauthorized release of sensitive information is high. The high impact for these risk categories is mapped to assurance level 4 of table 3.1. The cloud provider must select

RiskCategories	*ImpactLevelonBankData*
1. Inconvenience	High
2. Damage to reputation of the organization	High
3. Financial loss	Low
4. Harm to organization's programs or public interests	Moderate
5. Unauthorized release of sensitive information	High
6. Personal safety	Not applicable
7. Civil or criminal violations	Low

Table 3.5: Bank Data Impact Profile

assurance level 4 if he wants to store the bank's data in his cloud. Also, the cloud provider has to implement the authentication technologies which can meet the assurance level 4.

In the SCCM scenario, the selected assurance level depends on the cloud content. The two-factor authentication was implemented as the IBM customer requirement and the assurance level decision was also implicitly made by the IBM customer. The content stored in the cloud decided the impact profile of different risk categories. The assurance level 3 was selected after the complete analysis of IBM cloud content.

3.6 Select Technology to Meet the Assurance Level

The appropriate authentication technology is selected to meet the required assurance level. This thesis strives to achieve assurance level 3 by using two-factor authentication. The security risks are avoided and mitigated using a strong two-factor authentication system.

An authentication model has been discussed in the beginning of this chapter. It is important to mention that the selected assurance level should be met through out the authentication process. Both the authentication phase and the registration phase should be implemented properly to meet the selected assurance level [nisa]. Technologies involved in an authentication process include the authentication token used, the token management mechanism and the communication protocol supported by the claimant and the verifier to transfer the token.

Following three sections describe in detail the process of selecting the appropriate technology for the two-factor authentication solution. Section 3.7 targets on selecting the token used for authentication and section 3.8 examines the public key infrastructure (PKI). Section 3.8 gives insight into the transport layer security; a PKI enabled service.

3.7 Authentication Token

An token contains a secret to be used in the authentication process [nisa]. A token can be something a claimant knows, something a claimant has or something a claimant is. It can be in a soft or hard form. It can be a password, a hardware token or a digital certificate. In short any entity which can help the claimant to authenticate itself is considered as a token.

3.7.1 Token Model

An authentication token model is shown in figure 3.4 [nisa]. An authentication token has two optional inputs and one output. Different components of a token model are explained below.

Token Secret: Each token has a secret which distinguishes it from other tokens. Token secret can be a cryptographic key, a secret seed to generate random one-time passwords or a grid combination. In trivial scenarios, the secret can be the token itself. For example, a password is a token and a secret at the same time.

Token Input Data: It is an optional field. Token input might be required to generate the output. It can be provided explicitly by the user or is an implicit feature of a token. Token input data is implicit for tokens like passwords.

Token Activation Data: It is an optional field. Token activation data might be required to activate a token. Examples are a PIN used to activate a smart card or a hardware token.

Token Output: Token output is used to authenticate the claimant. It is a function of token secret and token input data. While token secret remains constant for longer duration, the input data changes more frequently.

3.7.2 Single-Factor Tokens vs Multi-Factor Tokens

Single-factor tokens cover one of the three authentication factors (knowledge factor, possession factor, characteristics factor). Single-factor tokens do not require another authentication factor for their activation. Two or more than two single-factor tokens must be combined together to provide a two-factor authentication solution.

Multi-factor tokens cover more than one authentication factors. A multi-factor token belongs to an authentication factor and requires an extra factor for activation. The token will not produce output without this extra authentication factor. A multi-factor token can be single-handedly used in a two-factor authentication solution. A hardware token generator, requiring a password for its activation, is an example of multi-factor token.

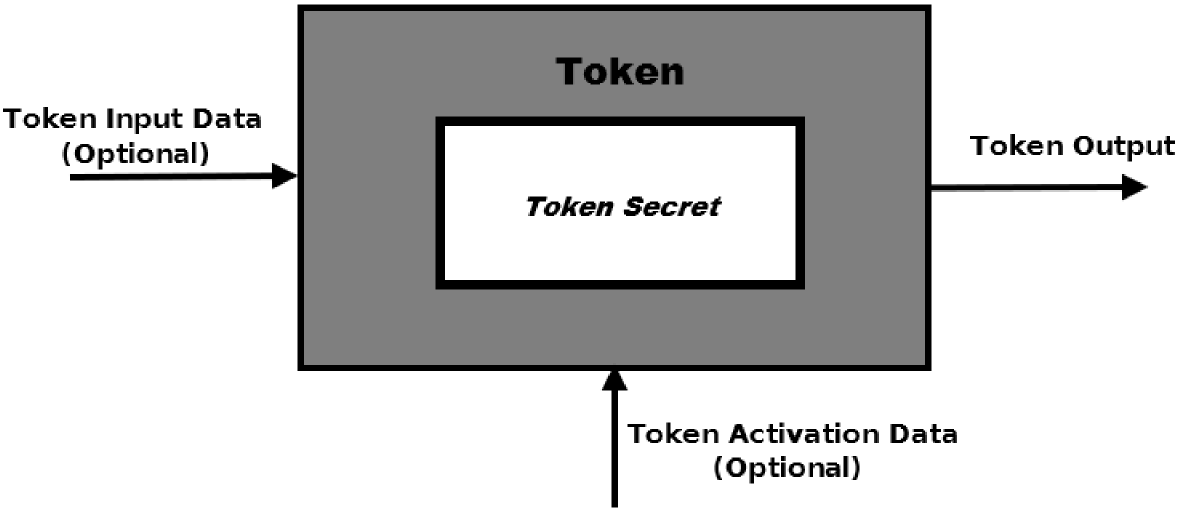

Figure 3.4: Authentication Token Model

3.7.3 Hardware Tokens vs Software Tokens

Tokens can be further classified into hardware tokens and software tokens. Hardware tokens are hardware devices having storage capacity and optionally an integrated chip to do some computation. They can store a cryptographic key or can compute one-time passwords using their computational capability.

Software tokens are software implementations of hardware tokens [egia]. They can optionally use the computational capability of the platform they are installed on. Typical examples of software tokens are soft cryptographic keys and software to compute one-time password.

3.7.4 Token Technologies Under The Hood

This section explains the token technologies in detail. These technologies are important to mention as all the authentication tokens are based on them. One or more of these technologies are combined to construct an authentication token.

Shared Secret

A shared secret is an old and common authentication technique. It belongs to the knowledge factor of authentication. The claimant and the verifier share a secret with each other and the secret is known only to both of them. A shared secret is commonly used with a user ID during the authentication.

The claimant is supposed to memorize the secret and is obliged to not share the secret with others. The verifier stores the secrets in encrypted form to protect it from unauthorized access. Examples of shared secret are password, PIN and a personal question.

Passwords are widely used for authentication but the fact is, they do not resist most of the security threats mentioned in section 2.2. High entropy passwords can partially resist the password discovery attacks and shoulder surfing attacks.

Digital Certificate

The claimant and the verifier use digital certificates to authenticate themselves against each other. Digital certificates do not only act as authentication tokens but also facilitate security goals like encryption, data integrity and non-repudiation.

Digital certificates can be self-signed or signed by a certificate authority. Both the mechanisms provide the same security level. Only difference is that, all self-signed certificates must be stored on the verifier side.

The certificates are resistant to the security risks of the shared secrets. While a certificate might be stolen or copied, it can not be known or revealed to attackers. Attacks like shoulder surfing, replay and brute-force are ineffective on the certificates.

One-Time Passwords (OTP)

One-time passwords are generated by an algorithm. They are different from typical passwords as they do not repeat themselves. Every next password generated will be different from the previous ones. The password generating algorithm uses an underlying secret as a seed. The seed is shared between the claimant and the verifier. During the authentication process, the claimant and the verifier generate the one-time password individually. The user sends the one-time password to the verifier which compares it with the locally generated password. The claimant is authenticated if both the passwords match.

One-time passwords are already very common. Messages sent on mobile phone for authentication are one-time passwords. Some banks provide TAN number lists containing one-time passwords. Scratch cards also contain one-time passwords.

A software developer can use algorithms to generate one-time passwords. The password generating software can be embedded in a hardware. Java Smart cards, USB dongles, and mobile phones can run the password generating software to generate one-time passwords. Hardware tokens generating one-time passwords are also available in market.

One-time passwords technology has certainly increased the customer acceptance of two-factor authentication. They can be distributed in software, hardware and even in paper form and open new opportunities for two-factor authentication solutions.

One-time password algorithms come in two flavours:

HMAC-Based one-time password algorithm (HOTP): HOTP is an event based password generating algorithm. The passwords are generated based on Hashed Message Authentication Code (HMAC) [rfca]. The claimant and the verifier agree on a counter value in the beginning. The counter is incremented each time the event occurs i.e. a password is generated. The user and the verifier keep track of the counter value. The difference in counter value at both sides is known as window size [rfca]. If the password generated at claimant's side is out of the window size, it will be rejected from the verifier. The claimant has to synchronize the counter with the verifier in this case.

A fixed window size is agreed upon in the beginning. The narrower the window size is, the less is the opportunity for an attacker to successfully perform a brute-force attack. It is because, as the attacker makes a wrong guess of password and tries to authenticate against the verifier, the verifier will increment its counter. Shortly, the counter will be out of the window size and the whole authentication process will be shut down. Now, even if the attacker makes a right guess, the verifier will reject this password as it is out of the window size.

HOTP is a dynamic and effective algorithm for generating one-time passwords. On the other hand, it also has some drawbacks. Passwords generated with HOTP algorithm are not time sensitive. Also, HOTP authentication solution can suffer form denial of service attacks. An attacker can deliberately make repeated wrong attempts to log on. The window size will be reached soon and the actual users will not be able to log on. The claimant will have to synchronize the counter with the verifier again.

Time-Based one-time password algorithm (TOTP): Time-based one-time password algorithm is an extension of HOTP algorithm to support the time-based moving factor [rfcc]. It is an improved version of the previous algorithm.

HOTP had one major drawback; the passwords generated by HOTP are not time sensitive. They have no validity time limit and can be used any time later. This increases the security risk as the password generating token can be shared with an attacker for a short period of time. The attacker can note down the password and uses it later to authenticate itself. As the passwords do not expire, the attacker has full freedom to use the stolen password.

Time-based one-time password algorithm prevents this situation. The passwords generated by TOTP algorithm are valid only for short period of time. The passwords expire after the validity period and cannot be used for authentication. The password validity time can be set in the algorithm and is normally set to 30 to 60 seconds.

Biometrics

Biometrics authentication use physical or behavioural characteristics of the claimant. Physical characteristics include fingerprints, iris recognition, face recognition etc. Behavioural characteristics include rate and flow of movements like keyboard typing patterns.

Introducing a new user in the biometrics authentication system is called enrolment [ffi]. The samples of the user are taken in this process. Samples can belong to physical or behavioural characteristics and are converted into mathematical model templates. The mathematical model templates are stored in a database for comparison during the authentication phase.

Biometrics authentication requires specific hardware and software installed on the user′s computer. The hardware reader makes a live scan of the pre-decided biometrics characteristic during the authentication. The scan results are then transferred to the verifier which decides for the acceptance or rejection of the scan results. The scan results can change from time to time as the scans are made in different environments at different times. Biometrics authentication includes a certain amount of tolerance to accommodate this difference.

Biometrics are commonly used during live authentication. On the other hand, using biometrics for remote authentication is not widely used because of certain drawbacks. Biometrics are converted in the bit format during the scan process. An attacker can copy the bit format and use it later. Another drawback of biometrics is, they can not be revoked. They have life long constant values. Authentication tokens like passwords and certificates can be changed or revoked. But it is not the case for biometrics.

Tamper Evidence, Detection and Response

Tamper evidence, detection and response play important role in the security of authentication tokens. The terminology is defined below.

Tamper Evidence: A token is considered to be tamper evident if there is an external indication that an attempt has been made to compromise the physical security of the token [fipa]. Tamper evidence is the first step in increasing the token security. If there is no tamper evidence on a compromised token, it can be used inattentively for a long time. A lot of research has been done to make hardware tokens tamper evident. Software tokens on the other hand are less known to be tamper evident.

Tamper Detection: Tamper detection is automatic determination by a token that an attempt has been made to compromise its physical security [fipa].

Tamper Response: Tamper response is the automatic action taken by the token when a temper detection has occurred [fipa]. The token can take actions to destroy its secret information or to shut down the output generation until activated again.

Technologies		Example Tokens
Shared Secrets		Password, PIN, Secret Questions
One-time passwords		TAN Lists, Messages sent on telephone, Scratch cards
Digital Certificates		Self-signed, Issued by a certificate authority
Software tokens	One-time passwords	Mobile devices with one-time password generating software installed. Windows desktop applications are also available to generate one-time passwords.
	Digital Certificates	Cryptographic token placed on a hard drive or imported in a web browser.
Hardware tokens	One-time passwords	Separate hardware tokens generating one-time passwords.
	Digital Certificates	USB tokens, Smart Cards
Biometrics		Fingerprint recognition, Retina recognition, Face recognition etc.

Table 3.6: Token Technologies and Related Examples

3.7.5 Token Technologies and Token Examples

The above mentioned technologies are used to generate different types of authentication tokens. Different combinations of token technologies give rise to different tokens. Table 3.6 provides some examples of well-known tokens with respect to the technologies used.

3.8 Public Key Infrastructure (PKI)

PKI is a framework that consists of encryption mechanisms, security policies and utilities generating, storing and managing keys. It is a combination of hardware, software, policies and procedures for managing certificates and keys [SC02].

Public Key Infrastructure is integrated in the SCCM two-factor authentication solution as it can ensure certain information security properties. PKI provides confidentiality by encrypting data, integrity using hash functions and authentication by providing the certificate to the PKI clients. Non-repudiation can also be achieved by using the private key to sign the outgoing messages.

Below I take a closer look at different PKI components and explain their responsibilities during the authentication process.

3.8.1 Certificate Authority (CA)

Certificate authority is the major component of PKI. It is similar to a notary [nisc] and provides the same services. Certificate authority performs following functionalities:

- Issues certificates
- Revokes certificates
- Formulates Certificate Policy and Certificate Practice Statement

The certificate authority (CA) issues certificates to the PKI clients. Each CA holds a certificate of its own and a corresponding private key. The CA uses its private key for signing the issued certificates and documents like certificate revocation list (CRL) etc. The CA certificate is used by PKI clients to confirm the validity of the CA signatures. The CA certificate is self-signed or is signed by some other CA up in the hierarchy.

3.8.2 Registration Authority

Registration authority provides assistance to the certificate authority. Registration authority verifies the PKI clients identity and asserts if the client is entitled to hold a certificate.

Registration authority receives the request for new certificate and interrogates the requester's identity. Registration authority validates the requester's identity and sends the certificate request to the CA. The CA issues the certificate and sends it back to the RA. The RA receives the certificate and passes it on to the certificate requester.

Registration authority should take full care in asserting requester's identity and credentials. The identity assertion can be the weak link in the security chain. The complete security process is compromised, if the PKI client's identity is not validated properly.

The registration authority is used in a scenario when CA receives very large number of certificate requests. Registration authority provides scalability to the PKI system. Different registration authorities can be spread on different geographical locations to assist the certificate authority.

3.8.3 PKI Client

PKI client is an entity (an individual or an organization) that uses the PKI services. The PKI client can be a certificate holder or a relying party, as explained below.

The *certificate holder* obtains a certificate from PKI and uses it for its authentication. It makes a request against the CA to issue a certificate and is called a certificate requester. The certificate requester might generate the public private key pair locally and sends the public key to CA to issue a certificate. If it does not have the capability to generate the public private key pair, it can ask the CA to generate the key pair for it. The certificate requester also provides the

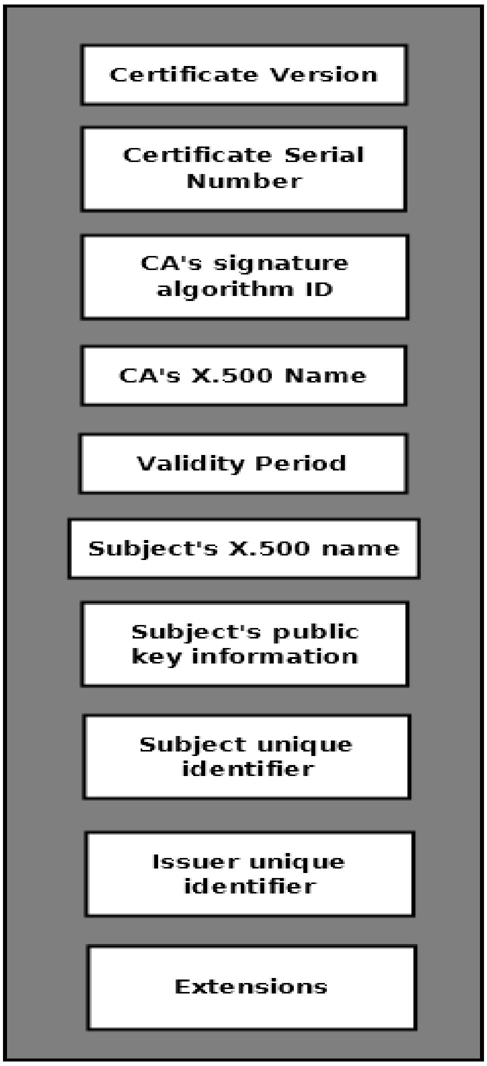
Certificate Version
Certificate Serial Number
CA's signature algorithm ID
CA's X.500 Name
Validity Period
Subject's X.500 name
Subject's public key information
Subject unique identifier
Issuer unique identifier
Extensions

Figure 3.5: Public Key Certificate [SC02]

extra information required by the CA for the identity validation. Once provided with the certificate, it is responsibility of the certificate holder to protect the corresponding private key.

The *relying party* relies on the PKI services to vouch for the authenticity of the certificate presented to it by the certificate holder. The relying party keeps the certificate of the certificate authority in its local trust store to assert the validity of the certificate.

Both the web client and the web server act as certificate holder and relying party in the SCCM two-factor authentication solution. Both hold a certificate and also rely on the CA to authenticate each other's certificate.

3.8.4 Public Key Certificate

Certificate authority generates the digital certificates. The digital certificates are kept by individuals and organizations and are used as authentication tokens.

A public key certificate contains the information about the certificate owner. This information is normally stored as Distinguished Name (DN) attribute and includes the fields like name, organization, organizational unit and country of the certificate owner. The public key certificate has a corresponding private key which is owned by the certificate owner.

The certificate is digitally signed by the CA and thus any information contained in the certificate can not be changed. The digital signatures make sure that the public key and the DN contained in the certificate are not altered seamlessly. Any attempt to change the certificate credentials will be detected, as the digital signatures of the certificate will change.

X.509 version 3 certificate and its essential components are shown in figure 3.5. The digital certificate is one of the two authentication tokens used in the SCCM two-factor authentication implementation. The certificate along with the corresponding private key is imported in the web browser.

3.8.5 PKI Repositories and Certificate Distribution

PKI stores certificates in a special repository. The repository is typically an implementation of the X.500 standard [nisc]. The repository is used to store certificates and distribute them later. The certificates can be distributed either manually or using LDAP protocol [SC02]. The PKI repository is also updated on regular basis to remove the revoked or expired certificates and add the new ones.

3.9 Transport Layer Security

Transport Layer Security (TLS) is a PKI enabled service. It is a cryptographic protocol. The cryptographic protocol can provide key agreement, encryption, authenticity and non-repudiation. TLS uses different cipher suites. The cipher suite is a term commonly used to represent the combination of encryption, authentication and a message authentication code (MAC) algorithm. The client and server decide on using the particular cipher suite before the TLS session is established.

$$CipherSuite = Encryption + HashFunction + Authentication/KeyExchange$$

TLS protocol is made up of two layers[HK12] as shown in figure 3.6. The first layer is a part of application layer and includes handshake protocol, change Cypher Spec protocol

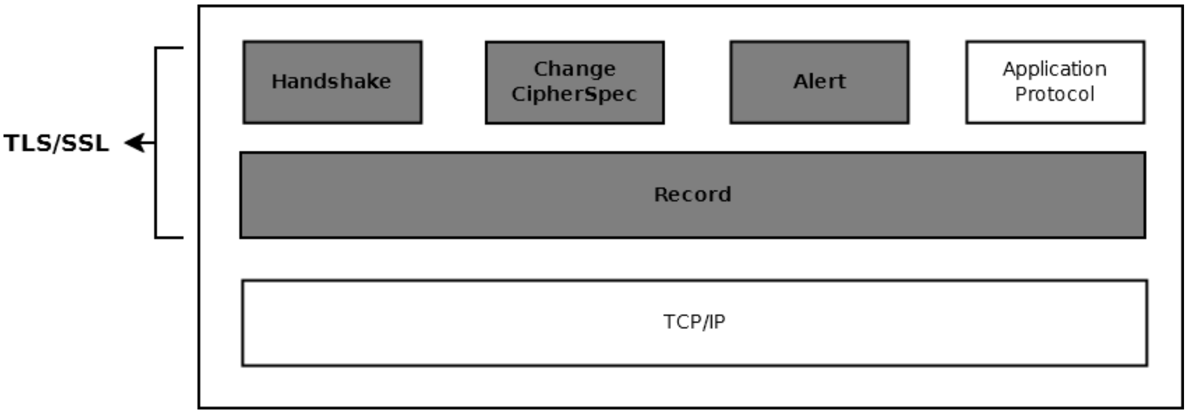

Figure 3.6: TLS/SSL Protocol Architecture

and alert protocol. The second layer consists of Record protocol. Below I explain the TLS protocol in detail.

3.9.1 Handshake Protocol

Handshake protocol initiates the TLS communication between the client and server. Its different steps are shown in figure 3.7. I explain below the step by step details of TLS handshake between the client and server.

Step 1: The client sends the "ClientHello" message to the server. It also sends the supported cipher suite, a random number (nonce) and the highest supported TLS version.

Step 2: The server receives the message from the client and looks into the details. It chooses the best possible cipher suite and TLS version, which is supported by both client and server. The server sends a "ServerHello" message along with a random number (nonce), the chosen cipher suite and TLS version.

Step 3: The server sends its certificate to the client. The certificate is validated by the client. It is important to mention here that certificate validation is not part of TLS protocol and is performed locally by the client. The certificate validation is done by the web browser on client-side and by the web server on server-side.

Step 4: Server sends a "CertificateRequest" message to the client, indicating that client also needs to provide its certificate.

Step 5: The server sends the "ServerHelloDone" message to the client. The message indicates that the server has completed the handshake negotiation.

Step 6: The client sends its certificate to the server.

Step 7: The client generates the "PreMasterSecret". It is used by client and server later to derive the "Master Key". The client encrypts the pre-master secret with the server's public key, which it has already received included in the server certificate. The encryption ensures that only server can access the pre-master secret.

Step 8: It is time for the client to prove the possession of the private key. The client uses its private key to create a digital signature over the previously sent messages. These messages along with the digital signature are sent to the server. Server detects the validity of the digital signature using the public key extracted from the client certificate.

At this time, both the client and server generate the Master key and the session key. Both the client and server use the pre-master secret and shared random numbers to derive these keys.

Step 9: The client sends the "ChangeCypherSpec" to the server. This message indicates that all the messages in future will be encrypted and hashed using the newly generated session key. The asymmetric key encryption is not used any further. Symmetric key encryption is adopted from this point on because of its higher efficiency.

Step 10: The client also sends the "Finish"message to the server. The server tries to decrypt the message and verify the hash message using the symmetric key. If it fails to do so, the handshake process is terminated by the server.

Step 11: The server also sends the "ChangeCypherSpec" to the client indicating the use of symmetric keys in future.

Step 12: The server sends the "Finish"message to the client. The client tries to decrypt the message and validates the message authentication code (MAC). The client aborts the handshake if it can not decrypt or validate the "Finish"message.

3.9.2 Change Cypher Spec Protocol

This protocol is used to change the encryption mode of the session between the client and server. The client or server can demand to change the encryption key. The new encryption key can be computed by using the information exchanged during the handshake.

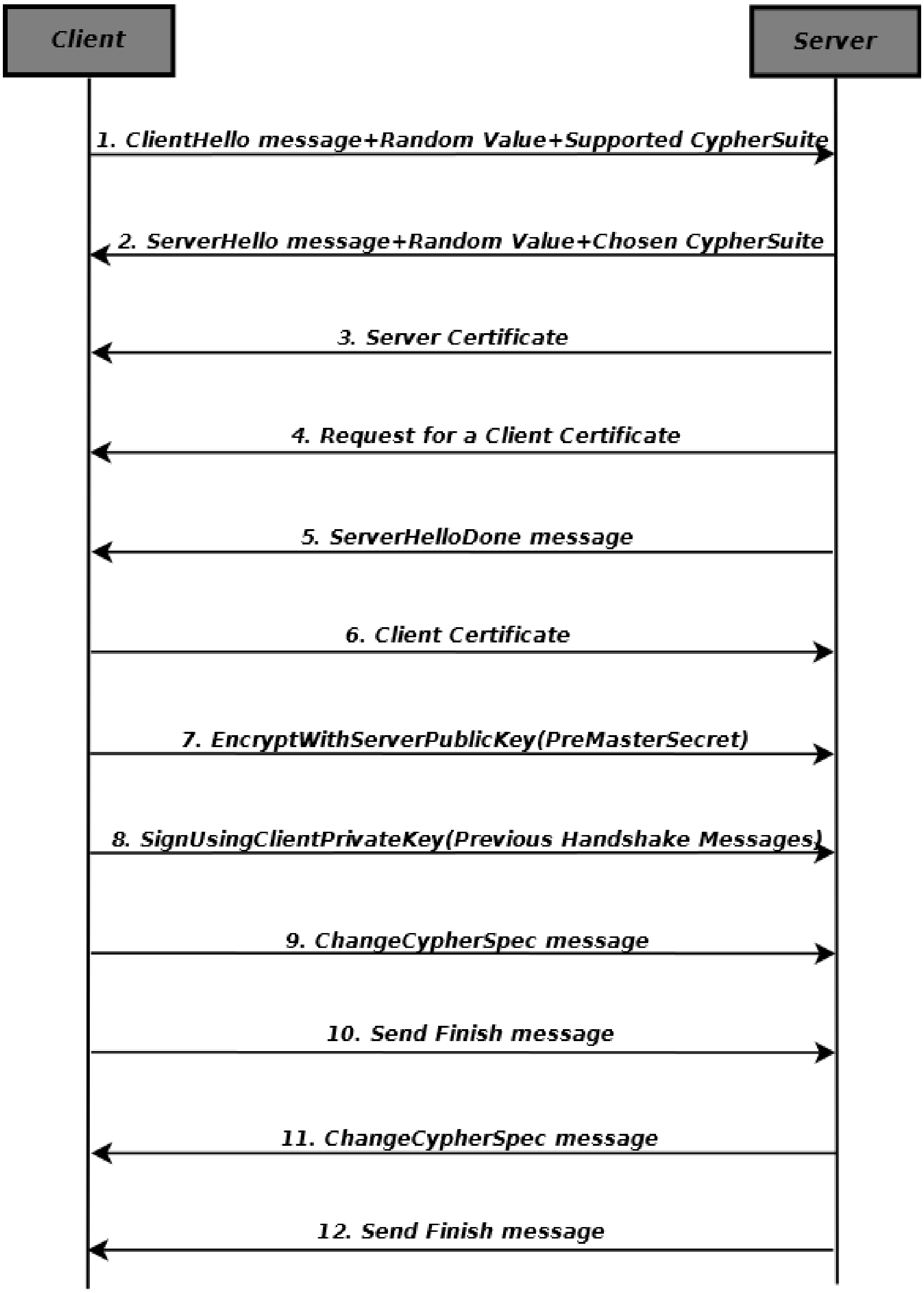
Client
Server
1. ClientHello message+Random Value+Supported CypherSuite
2. ServerHello message+Random Value+Chosen CypherSuite
3. Server Certificate
4. Request for a Client Certificate
5. ServerHelloDone message
6. Client Certificate
7. EncryptWithServerPublicKey(PreMasterSecret)
8. SignUsingClientPrivateKey(Previous Handshake Messages)
9. ChangeCypherSpec message
10. Send Finish message
11. ChangeCypherSpec message
12. Send Finish message

Figure 3.7: TLS/SSL Handshake Process

3.9.3 Alert Protocol

Alert protocol is used to communicate a fatal error between the client and server. It is normally used before terminating the connection and conveys the reason of connection termination.

3.9.4 Record Protocol

As specified in RFC 2246, the record protocol might perform four functions [rfcb]:

- Fragment the data coming from the application into manageable blocks and assemble incoming data to pass up to the application layer.
- Compress the outgoing data and decompress incoming data.
- Apply a message authentication code (MAC) or a hash to the outgoing data and use the MAC to verify incoming data.
- Encrypt the hashed data and decrypt the incoming hashed data.

3.10 SCCM Information Security Paradigm

Smart Cloud Content Management (SCCM) is a SaaS content management service. The content stored in the IBM content management solution might be confidential and requires higher security measures to be implemented. The information security paradigm is based on confidentiality, integrity, authentication, availability and non-repudiation. An authentication and security solution is integrated in the IBM cloud service to achieve the information security goals. PKI framework is used to ensure information security goals in SCCM. Below I go into further details of each security goal and observe how PKI and PKI enabled services can ensure it.

3.10.1 Ensuring Confidentiality

Confidentiality is to prevent unauthorized disclosure of information. Encryption is the key to ensure confidentiality. During the encryption process, the plaintext is transformed into encrypted text. The encrypted text can be decrypted using the secret key. The sender and receiver know the secret key and thus only they can decrypt the encrypted message. The secret key used for encryption is either symmetric or asymmetric, as explained below.

Symmetric Key Encryption

The sender and receive share a single secret key and use it to encrypt and decrypt the message. A new secret key is normally calculated for each communication session. The symmetric key encryption is relatively efficient as compared to asymmetric key encryption. On the other hand it is difficult for the sender and receiver to agree upon a shared secret. DES, 3DES and AES are commonly used symmetric key encryption algorithms.

Asymmetric Key Encryption

It is also known as public key encryption. Two different keys are used, one for encrypting the message and the other one for decrypting it. The key pair consists of a public key and a private key. The public key is distributed freely and is not considered as a secret. The private key is owned by an entity and is never shared with others. The message is encrypted with public key and can only be decrypted using the private key. Also, it is not possible to calculate the private key with the help of public key.

Asymmetric encryption is slow and inefficient as compared to symmetric encryption. The major benefit of asymmetric encryption is that there is no need to agree upon a shared secret between the client and server. RSA is a well known algorithm for public key cryptography.

A common practice is to use the asymmetric encryption in the beginning of the communication session to calculate the symmetric key and then switch to the symmetric key encryption to enhance efficiency.

3.10.2 Ensuring Data Integrity

Integrity makes sure that the data and transactions are not altered or modified. Cryptographic hash functions provide integrity. Hash functions take the outgoing message as an input and calculate a smaller chunk of data, known as the message digest. The message digest is a kind of fingerprints of the actual message. Hash functions have some unique properties which make them a perfect choice for data integrity. Hash functions are one-way algorithms. It is infeasible to calculate the message itself, with the knowledge of the message digest. Also, it is not possible to intentionally generate the same message digest with two different messages.

The sender calculates the small message digest, appends it with the original message and transmits it to the receiver. The receiver receives the message and the message digest, calculates the message digest using the message received and compares both the message digests. If both message digests match, the data integrity is ensured.

It is important to mention here that data integrity is important even if encryption is used. While using encryption, the message transmitted is not clear to the attacker. But it is still possible for an attacker to change some random bits of the encrypted message. The randomly changed bits can alter the message. Sending the message digest along with the actual message makes sure that the data is not intentionally or accidentally altered.

3.10.3 Ensuring Authentication

Ensuring authentication is to verify the user is who it claims to be. Authentication is a two way process. The client should be able to authenticate the server and vice versa.

Authentication is done in two steps:

1. Validating the certificate.
2. Extracting the certificate credentials and comparing them with the user registry attributes.

It is important to note that certificate validation plays a partial role in authentication. Authentication also requires extracting the certificate credentials and comparing them with the LDAP user credentials.

Authentication is ensured, as the client and the server hold their respective certificates and private keys. The certificate itself is not a secret and simply possessing a certificate is not sufficient for authentication. Authentication is possible with the combination of a certificate and the corresponding private key. Also, certificates can not be altered, as they are signed by the certificate authority's (CA) private key. Thus, the certificate and corresponding private key are bound to each other and both work together to ensure authentication.

3.10.4 Ensuring Availability

Availability is the main concern of a successful authentication solution. It is also important for customer satisfaction.

To provide higher availability, the users should always have access to the authentication tokens and if required, the hardware to activate the tokens. To ensure availability, the computing systems should be up and running to authenticate the users and the communication channels should function correctly.

3.10.5 Ensuring Non-repudiation

Non-repudiation ensures that the sender can not deny the message sent. Standard TLS protocol does not provide non-repudiation. Digital signatures are used to ensure non-repudiation. The user signs the message using its private key and sends it to the other parties. Only the user has control over his private key and thus can not deny the digital signatures.

Non-repudiation is ensured only if the private key is kept solely by the user. The user has to generate the public private key pair locally and sends the public key to the certificate authority for certificate generation. It is hard to ensure non-repudiation if the user does not have the capability to generate the public private key pair and depends on the certificate authority to create the pair for it. More than one parties know the private key in this case and the user can deny the validity of the digital signatures.

4 Two-Factor Authentication Solution for SCCM

WebSphere application server (WAS) is a middleware component. WebSphere application server security is based on Java Authentication and Authorization Service (JAAS) framework. WebSphere manages and runs the SCCM content management and search services and is an integrating point for the new incoming services. SCCM stores the user credentials in an LDAP user registry. Figure 4.1 shows the interaction between the web client and SCCM components during authentication process. The user provides authentication tokens via the web browser to the IBM HTTP Server. The IBM HTTP Server passes these credentials further to the WebSphere application server. WebSphere interacts with the user registry to assert the validity and authenticity of the authentication tokens and allows or denies access to the web client.

WAS provides a single authentication point for all the deployed applications. WebSphere application server provides Single Sign On (SSO) authentication using the LTPA token. WebSphere sends the LTPA token embedded in the session cookie to the web client. Further, LTPA token is cached on the server side. It is used to ensure SSO and prevents multiple authentication requests from the same user.

4.1 WebSphere Authentication Mechanisms

WebSphere application server provides different authentication mechanisms. Below I discuss four well-known authentication mechanisms of WebSphere application server [CS09].

HTTP Basic Authentication

A built in pop up is displayed to the user. The user enters the user name and password. The password is sent to the application server in base64 encoding.

Form-based Authentication

Form-based authentication allows the developers to customize the login page. The customized login page is presented to the user. It enters the user name and password and authenticates itself against the application server.

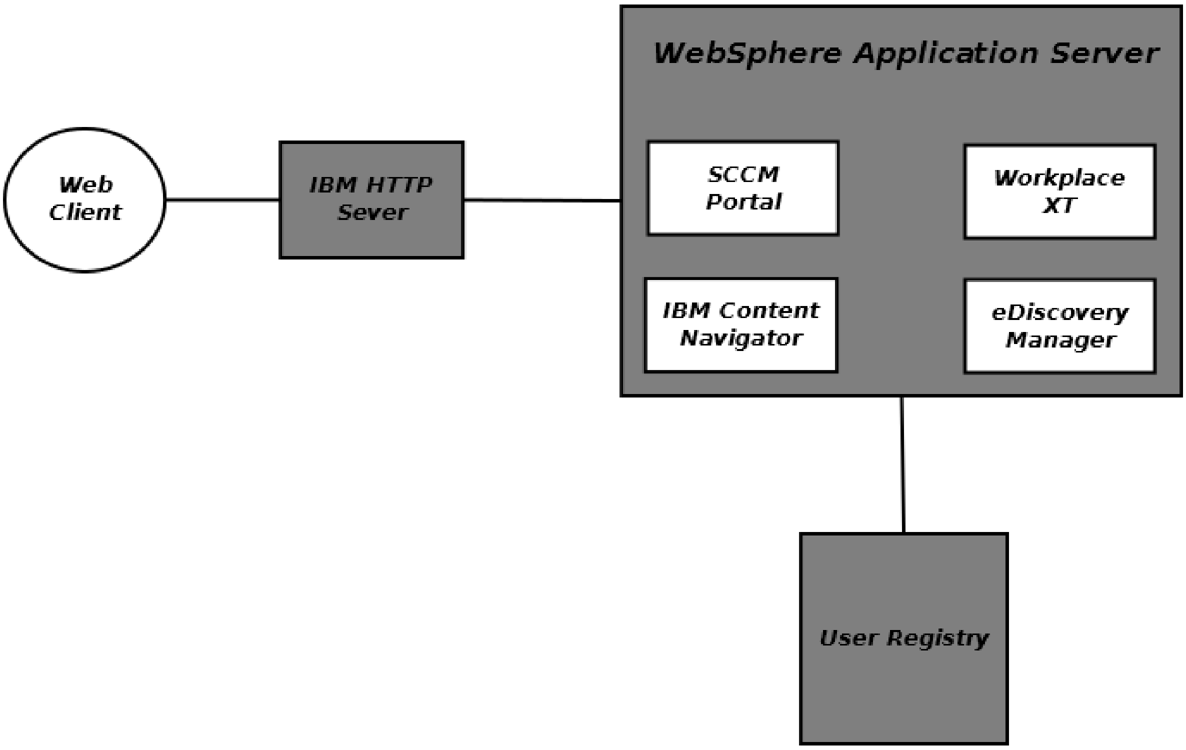

Figure 4.1: SCCM Components

Client Certificate Authentication

This mechanism requires digital certificates for authentication. The digital certificate is used as an authentication token like user names and passwords. Both the client and server can use their respective certificates to authenticate each other.

Customized JAAS Authentication

WAS also provides the possibility to include custom login modules. Java authentication and authorization service (JAAS) is a Java implementation of Pluggable Login Modules (PAM). The login modules can be plugged into the pre-existing WebSphere login modules. JAAS should be used if the authentication requirements are different than the authentication mechanisms mentioned above. One example can be the use of time-based one-time passwords (TOTP) for client authentication. One has to implement its own JAAS custom login module to provide TOTP authentication.

4.2 Configuration Details

IBM HTTP Server (IHS) works as a relay between the web client and the application server. It receives messages from web client and sends them directly to the WebSphere application server.

IHS stores the signer certificates in its keystore and uses them later for client certificate validation. The signer certificate is a certificate authority (CA) certificate or a self-signed certificate. If the SCCM clients hold the self-signed certificates, all the corresponding client certificates must be included in the web server keystore. For the SCCM production system, the client certificates are issued by a certificate authority (CA) and only the CA certificate is added in the web server keystore. The IBM Key Management tool (ikeyman) can be used to add or remove CA certificate from web server keystore.

Multiple virtual hosts can be defined in the IBM HTTP server configuration. Different virtual hosts can support different security policies and authentication settings. While one virtual host requires certificate authentication for one website, the other virtual host might require password authentication for the other website. It is possible only, if the web sites are registered with different IP addresses in the DNS. It is recommended to set a global security policy if the web server holds only one IP address. I configure the web server to support single security policy on global scale, as all the SCCM cloud applications require the same level of security.

The TLS handshake does not mandate the web client to present a certificate. I configure the web server to explicitly require the client certificate during TLS handshake. Web server provides “SSLClientAuth” variable to set the client certificate requirement during TLS handshake. The variable can be set to values “None”, “Optional” or “Required”. I set this value to “Required” to force the web client to provide its certificate. If the client fails to present a valid certificate, the SSL handshake will fail.

The “SSLClientAuth” variable value can also be set to "Optional". While using the optional value, the client is first asked to provide its certificate. If it fails to do so, the WebSphere authentication can choose some other pre-configured authentication mechanism like one-time password authentication.

The SCCM web portal application runs in the web container of WebSphere. The IBM HTTP server plug-in receives the data from the web client and passes it on to the web container. A secure channel must be established between the plug-in and the web container. The plug-in exchanges its certificate with the web container to establish a secure channel. It will allow only authenticated plug-ins to communicate with the web container.

4.3 Implementation Details

4.3.1 Certificate Validation

Both the client and server exchange their certificates with each other and validate the exchanged certificates. The certificate validation involves following steps [SC02]:

- Verify the certificate is issued by a trusted certificate authority (CA). The trusted certificate authority is the one whose certificate is present in the local trust store.
- Verify the validity of the signatures present on the certificate.
- Confirm that certificate is not expired and is not revoked.

4.3.2 WebSphere Log-in Modules

WebSphere application server uses the “WEB-INBOUND” login configuration to control security of incoming web clients. The login configuration is a combination of different login modules, which are invoked during the authentication process. One can add or remove certain login modules in the login configuration to change the authentication process. It is also possible to include custom login modules to achieve certain authentication mechanism not supported by WebSphere application server.

Following two login modules are always included for the inbound web communication authentication.

1. com.ibm.ws.security.server.lm.ltpaLoginModule
2. com.ibm.ws.security.server.lm.wsMapDefaultInboundLoginModule

The two modules generate the LTPA SSO cookie on successful authentication. The Lightweight Third Party Authentication (LTPA) is used to ensure inter-operable authentication between distributed, multiple application servers and machines. SSO is a subset of LTPA and if enabled, an LTPA token is generated and embedded in the cookie. The cookie is sent back to the web client through the HTTP response. An LTPA token is time sensitive and expires after a specific time period. Therefore, it is important that the distributed machines participating in the authentication process are time-synchronized so that LTPA tokens do not expire too early or too late.

4.3.3 Certificate Mapping and Log-in Filters

Login filters are the servlet filters. The servlet filters run before the corresponding servlet and are used for pre-authentication. The login filters are invoked, depending on their order mentioned in the deployment descriptor. The servlet filtering is chained i.e. on completion, a filter hands over the control to the next one.

I use two servlet filters in my implementation. The first filter runs after the client provides its certificate and before the login page is presented to it. The filter extracts attributes from the client certificate and uses these attributes to make an LDAP query. The second filter runs after the client provides its user name and before the form-based authentication is performed by WebSphere. This filter maps the provided user name to the corresponding certificate owner.

As discussed before, certificate validation is only one part of the client certificate authentication. The certificate owner must also be matched to a valid LDAP entity. Also, the two authentication factors (client certificate and user name password) must be bound together. If the two factors are not bound together, it is possible to use first factor from one user and the second factor from another user to authenticate successfully. Login filters provide both functionalities. They map the client certificate to a specific LDAP entity and also bind the two authentication factors together.

During my implementation, the client certificate DN and the LDAP entity DN did not match. I had to find a solution to map the two distinguished names (DN). There existed no consistent mapping between the certificate DN and the LDAP entity DN. Given the client certificate, one can not search for the certificate owner in the LDAP registry. I had to bind them together to provide two-factor authentication. To bind the two authentication tokens, I store one of the certificate DN attributes in an additional LDAP entity attribute. For example:

$$LDAP''description''attribute = ClientcertificateCN$$

The "description" attribute in this example is also made unique throughout the LDAP, so that no two users can accidentally have the same attribute values. Now, I can use the client certificate to search for the corresponding LDAP entity in LDAP and bind the two authentication tokens together.

Filters do following tasks in the implementation:

- Extract user credentials (DN etc.) from the client certificate.
- Use the certificate DN attribute (e.g. common name CN) to search for the corresponding LDAP entity and store it locally.
- Receive the user name and password from the SCCM web portal after the user has submitted these values. Match the user name entered on the login web portal with the locally stored LDAP entity attribute.

In this way, filters help authenticating the client certificates and also bind the two authentication tokens together.

4.3.4 Step by Step Authentication Process

This section explains the complete authentication process step by step. The numbering below is related to the figure 4.2. The authentication process starts when the client tries to access the login page or the secured resources on the application server. Further authentication process continues as under:

1. The web server sends a list of trusted certificate authorities to the web browser. The CA certificates have already been added in the web server keystore before the authentication starts. The web server also sends its certificate to the web client.
2. The web browser prompts the user to select a certificate, which he has already imported in the web browser. The client selects one of the imported certificates and sends it to the web server. This certificate acts as the first authentication token.
3. If required, the web server retrieves the CA certificate from the keystore and validates the client certificate.
4. On successful validation of the certificate, control goes to one of the servlet filters. This servlet filter is configured to run before the login page is presented to the web client.
5. The servlet filter extracts certificate credentials (DN etc.) and use them to search for the particular user against the LDAP user registry. The certificate credentials used to search the LDAP must be unique and returns only one entry.
6. The successful LDAP search returns an LDAP entity, which is stored locally by the servlet filter.
7. The login page is handed over to the web server.
8. The web server relays the login page to the web client.
9. The web client enters its user name and password on the page and submits.
10. Control goes back to another servlet filter. This servlet filter is configured to run before the "j_security_check" servlet. The filter compares the user name submitted on the login page with the locally stored LDAP entity. If the user name matches with the LDAP entity attribute, the control goes on to the WebSphere application server form-based authentication modules.
11. The "j_security_check" servlet and the login modules run and an LTPA SSO cookie is generated.
12. The LTPA SSO cookie is handed over to the web client via web server.

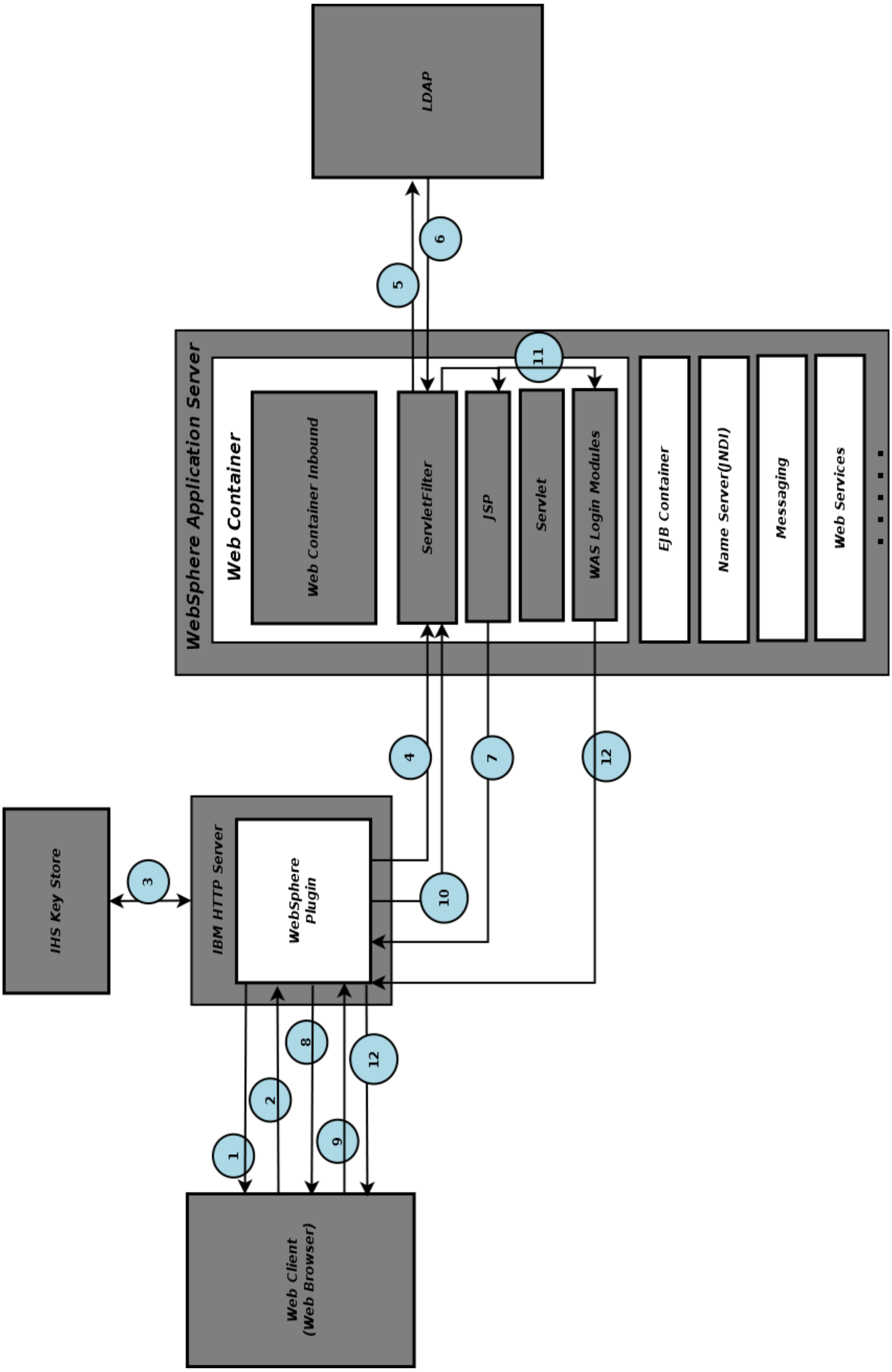

Figure 4.2: Two-Factor Authentication Implementation Details

4.3.5 Block and Redirect Login Requests for SaaS services not Supporting 2-FA

SCCM combines different SaaS services in its content management solution. These services are deployed on WebSphere application server as different EAR files. These services do not support two-factor authentication. They offer their own web portals for authentication. Prior to the two-factor authentication solution, one could use these web portals for authentication rather than authenticating through the SCCM portal. I blocked the access to the web portals offered by the SaaS services not supporting two-factor authentication. It is because the source code for these services can not be changed and thus one can not include the login filters discussed in section 4.3.3. The login filters are required to bind the two authentication tokens together.

In the implementation, I make sure that the SaaS services can be accessed only after the client is authenticated through the SCCM web portal. The opted solution is explained below:

1. Block web portal URLs of the services not supporting two-factor authentication.
2. Redirect the web portal requests for these services to the SCCM portal login page. I block the URLs by changing the deployment descriptors of respective services. Also, I include a new custom web portal for the corresponding products. The custom web portal redirects the incoming authentication requests to the SCCM portal log-in page.

This solution makes sure that the web clients complete the authentication process through the SCCM web portal. Once authenticated, they can also use other SaaS services not supporting two-factor authentication.

4.3.6 Rewriting HTTP requests to HTTPS requests

The two-factor authentication solution must redirect the incoming HTTP requests to the HTTPS requests. The SCCM web portal is presented to the clients, once their certificate is validated and the certificate validation can occur only if the client and server communicate over TLS channel.

Suppose, the client and server do not communicate over a TLS channel. The SCCM web portal will be presented to the client without requiring and validating the client certificate. Once the client enters its user name and password credentials and tries to authenticate, the system will go in an unstable state.

I configure IBM HTTP Server to redirect all the incoming authentication requests on port 80 to port 443. The IHS redirection engine receives the incoming requests and investigates the URL. If the URL contains HTTP keyword, the redirection engine rewrites it to HTTPS and sends it forward to IHS.

4.3.7 Increase Cryptographic Strength

IBM HTTP Server can demand stronger encryption for highly secure content. One can achieve higher security by configuring the IBM HTTP server to use stronger cryptographic ciphers. Only those web clients are given access, who can support the required cipher suite. It is recommended to restrict the encryption strength to 128 bits or higher. IHS uses "SSLCipherSpec" variable to specify the supported cryptographic cipher suites. Also, it can support different cryptographic cipher suites for different virtual hosts, directory contents or websites.

4.3.8 Logging Authentication Details

Logging the authentication events can increase post-authentication security. One can log the authentication data and use it later to observe abnormal authentication behaviour or a security breach. IHS supports detailed logging of the authentication data in different formats.

I log the authentication data e.g. the web client details, the cryptographic cypher used and the date and time of authentication. IBM Http Server provides variables which can be used in its configuration file to log the data. These variables include "HTTPS", "SSL_CLIENT_DN", "SSL_CIPHER", "HTTPS_KEYSIZE", "SSL_PROTOCOL_VERSION" and many more.

4.3.9 Certificate Revocation

Authentication token revocation is the mandatory part of each authentication system. This section explains the client certificate revocation. The client certificate must be revoked in following conditions:

- Employee holding the certificate leaves the company.
- The client certificate is compromised.
- The certificate holder credentials are changed.
- The certificate authority (CA) certificate is compromised.

Certificate Revocation List (CRL) is the list of revoked certificates. When the certificate authority issues the certificate to the PKI client, it sets the CRL Distribution Point (CDP) parameter in the certificate. The CDP parameter contains the information about the location of the certificate revocation list (CRL).

The certificate authority also publishes the certificate revocation list (CRL) to the CRL distribution point (CDP). The certificate authority signs the CRL before publishing it to the CRL distribution point. The digital signatures make sure that the CRL is not modified.

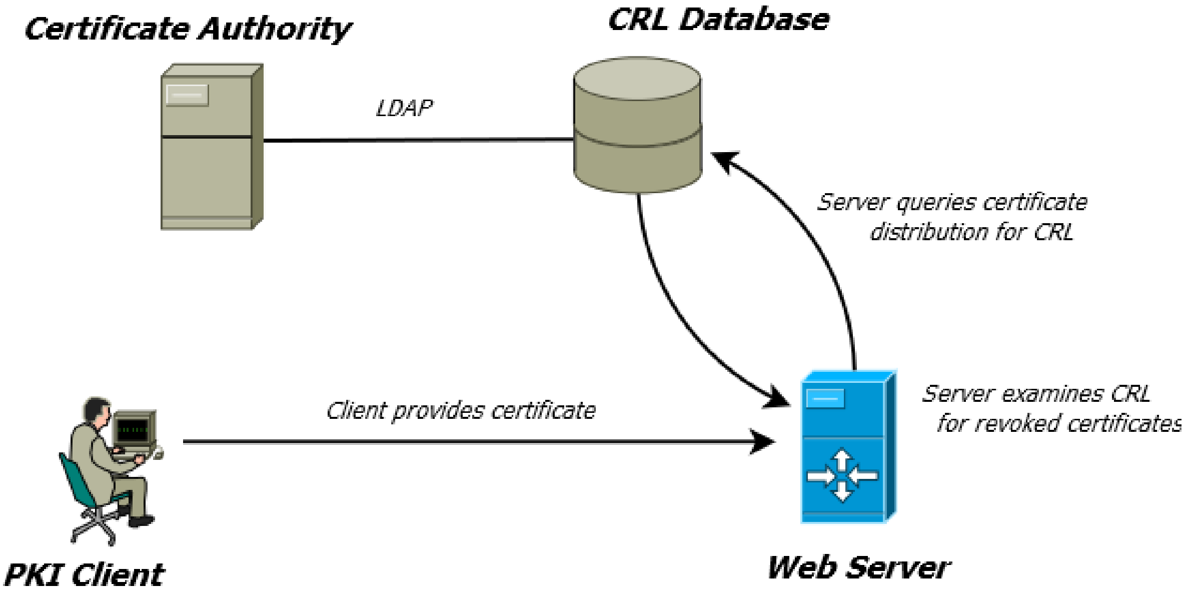
Certificate Authority
CRL Database
LDAP
Server queries certificate distribution for CRL
Client provides certificate
Server examines CRL for revoked certificates
PKI Client
Web Server

Figure 4.3: Certificate Revocation Process

During the authentication process, the PKI client presents its certificate to the web server. The web server looks into the client certificate and extracts the CRL distribution point information. The web server contacts the CDP and fetches the certificate revocation list (CRL). It checks the validity of the CRL and looks into it for the client certificate. If the client certificate is found in the CRL, it has been revoked and the client access is denied.

Fig 4.3 shows the complete certificate revocation process. It shows the sequential process of certificate validation during the authentication process. The steps are explained below:

Step 1: The web client presents its certificate to the web server during the authentication process.

Step 2: The web server receives the certificate. It extracts the CDP information from the certificate and requests the CRL distribution repository for the certificate revocation list (CRL).

Step 3: The CRL distribution repository provides the signed CRL to the web server. The web server verifies the non-repudiation and integrity of the CRL.

Step 4: The web server looks for the web client certificate in the CRL. The client certificate is rejected if its certificate is found in the CRL.

It is important to mention that the expired certificates are not included in the CRL. The web server explicitly checks for the certificate validity period and rejects the certificate if it is already expired. The expiration of the certificate is not related to its revocation.

5 Evaluation and Future Work

5.1 Two-Factor Authentication Solution Evaluation

This chapter evaluates the SCCM two-factor authentication solution. The two-factor authentication solution is evaluated with respect to different security threats. Security threats can originate different security risks and negatively impact the information system.

The authentication process is a combination of authentication tokens, repositories to store the tokens and a communication channel to transfer these tokens. The authentication tokens play an important role in the authentication process and must be resistant to a variety of security threats. The SCCM two-factor authentication solution uses two authentication tokens to authenticate the client. These tokens are the password and the client certificate. The communication channel transfers the tokens between the client and the server and should be resistant to the authentication process vulnerabilities.

Below I analyse the strength of the proposed two-factor authentication solution and argue its resistance against different security threats and vulnerabilities.

5.1.1 Token Theft

Token theft is related to the loss of a physical token and is possible if the attacker has physical access to the authentication token.

The SCCM two-factor authentication solution uses a digital certificate in its implementation. The digital certificate resists the theft attacks as it requires an activation password. Each time the user wants to import the certificate in the web browser, he must provide the activation password. The certificate activation password is known only to the certificate owner. The attacker does not know the activation password and a stolen certificate can not be misused.

5.1.2 Token Duplication

If the attacker can not steal the password, he might try to duplicate it. Common duplication attacks are copying a password written on paper or making an exact copy of an authentication token like digital certificate.

Passwords are not recommended to be written down. Also, the SCCM authentication solution changes the client passwords after each three months. It makes sure that the passwords older than three months can not cause a security risk.

Now consider the second authentication token i.e. the digital certificate. The certificates are digitally signed by the certificate authority. To make an exact copy of a digital certificate, the attacker must have access to the private key of the certificate authority to sign the certificate. The private key of a certificate authority is well secured and getting access to it is hardly possible.

5.1.3 Token Discovery

Token discovery is related to the knowledge of the token secret to an unauthorized person. The discovery attacks can be categorized in the offline and online discovery attacks.

The SCCM authentication process enforces the clients to choose strong passwords. The strong passwords have high entropy and are resistant to the offline brute force attacks. Also, changing passwords after each 90 days reduces the risk of token discovery attacks.

An authentication process is resistant to online discovery if the attacker cannot get access to the authentication token by repeated online attempts. In SCCM two-factor authentication solution, online guessing of the password is not possible until the attacker has access to the client certificate and the corresponding private key. Without the client certificate, the attacker has no access to the SCCM login web portal and can not make password authentication attempts.

As far as the digital certificate is concerned, the private key of the client is highly resistant to the token discovery attacks. It is because the private key is never communicated over the internet and it is also not possible to guess the private key with the help of the corresponding public key.

5.1.4 Replay Attacks

Replay attacks are possible if the attacker can authenticate itself by using the previous authentication message.

The SCCM two-factor authentication solution uses TLS communication protocol to transfer messages between the client and the server. The protocol provides resistance against replay attacks. TLS uses nonce and challenge during the handshake and it is not possible to use the same authentication messages again.

5.1.5 Eavesdropping

To resist the eavesdropping, an authentication process makes sure the token or information sent on the communication channel remains secret. Eavesdropping attacks are resisted by sending the messages in encrypted form.

The SCCM two-factor authentication solution thoroughly encrypts the messages exchanged between the client and the server. The passwords are also sent in encrypted form. During the handshake, both the client and server encrypt messages using each other's public keys. Meanwhile they agree upon a symmetric key. Once the handshake is complete, they start encrypting all the messages using the symmetric key.

On the other hand, the PKI private keys are not sent online, so there is no chance of eavesdroppers to get access to a private key. The public keys are sent over the internet, but getting access to the public keys is not sufficient for successful TLS handshake.

5.1.6 Phishing

Phishing is performed to reveal the token by tricking the client into thinking the attacker is the verifier.

During the SCCM authentication process, the server presents its certificate to the client web browser. The web browser checks the validity of the server certificate. It will refuse the connection if the server certificate is not valid. The web browser will also prompt the user about the invalid certificate.

The SCCM two-factor authentication solution warns the user of the invalid server certificate and a possible phishing attack. It is the responsibility of the user to pay attention to the warning and avoid any possible security threats.

5.1.7 Session Hijacking

Session hijacking is resisted by not allowing an intruder to participate in the session.

A session identifier (session ID) is agreed upon by the client and server before the session starts. In SCCM two-factor authentication system, the client and server agree upon the session ID during the TLS handshake. The session ID is encrypted using the asymmetric encryption keys before transmission. An attacker can not decrypt the session ID, as he has no access to the private keys of the client and server.

Also, the session identifier is a short living and highly random value and guessing it using brute force techniques is not a feasible task.

5.1.8 Man-in-the-middle Attack

The SCCM two-factor authentication process provides strong resistance to the man-in-the-middle attack. An authentication process provides strong resistance to the man-in-the-middle attack if it does not allow the user to reveal its secrets to an attacker, disguised as the verifier.

The SCCM two-factor authentication solution implements two-way certificate authentication. The client and the server use their public private key pairs for authentication and even an ignorant client can not reveal a secret to the attacker masquerading as the verifier.

5.1.9 Social Engineering

Social engineering attacks establish a certain level of trust to persuade the user to reveal the token. The attacker prompts the victim to provide his authentication tokens. Social engineering is normally done during a telephonic or personal conversation.

The SCCM two-factor authentication solution uses two authentication tokens simultaneously and provides resistance against the social engineering attacks. A tricked user might orally reveal the password to the attacker but can not orally reveal the certificate token. For example, the attacker can not trick the user to reveal the certificate during the telephone conversation. The certificate must be transferred physically and requires extra effort.

5.2 Future Work

This thesis explains the process of choosing a SaaS based two-factor authentication solution and looked into its implementation details. The solution is designed according to the security needs of the enterprise world. It provides a strong two-factor authentication solution, which is capable to resist most of the well known security threats. The SCCM two-factor authentication solution provides the opportunities to further improve the enterprise security. New security mechanisms can be implemented and integrated in the existing two-factor authentication solution. I discuss below some future work related to this thesis.

Currently, the SCCM two-factor authentication solution does not store the client certificates in the cloud. The next step should be to store the client certificates in the cloud premises. Advantages of managing the client certificates in the cloud are multi-fold. Mapping the client certificate credentials to an LDAP entity, as described in section 4.3.3, is not required any more. One can simply compare the incoming certificate with the certificate stored in the LDAP to assert its validity.

The SaaS services can send encrypted emails to the clients if the client certificates are stored in the cloud LDAP. Secure Emailing (S/MIME) will improve the enterprise security. One has to keep record of all the revoked and expired public private key pairs to provide email encryption service. All the expired and revoked key pairs will have to be stored in the cloud. It is because decrypting the old emails might be required by the business or legal requirements. If an email is encrypted using an expired or revoked public key certificate, one can decrypt it only with the corresponding private key. It is possible only if the corresponding public private key pairs are stored in the cloud.

Amazon has recently announced a Cloud Hardware Security Module (Cloud HSM) [1]. It is a hardware appliance to store the encryption keys (both public and private) in cloud to meet business, regulatory and contractual requirements of data security. It enables database encryption, digital rights management and other information security goals like authenticity, encryption and digital signing.

As discussed before, the SCCM authentication solution does not provide non-repudiation of data exchanged. Non-repudiation means the sender can not deny the messages sent by him. It is possible only if the sender signs the messages using its private key. The private key should be owned only by the signer to ensure non-repudiation. Two client certificates are required to provide non-repudiation if a private key is already stored in the cloud. The private key stored in the cloud can not be used to provide non-repudiation.

Automatic addition and removal of the two-factor authentication module can provide high flexibility to the SaaS services. It will provide the flexibility to the IBM cloud customers to decide for the authentication solution of their own choice.

[1] Amazon Cloud HSM

6 Conclusion

This thesis provides insight in different aspects of information security. It analyses the security requirements of a set of real customers from the financial services market segment and in a specific case from an insurance company in Europe. It proposes a two-factor authentication solution to enhance the security of the information systems. It further provides guidelines to implement a two-factor authentication solution and implements a prototype for it.

The thesis thoroughly analysed the security risks related to content management systems. It analyses different security threats and their potential impact on an information system in the cloud. The risk to an information system is a function of the probability of threat occurrence and its potential impact. The risks exposed to the information system can be mitigated by a strong authentication mechanism. Many e-governments and business standards provide guidelines on the strong authentication. They mandate use of two-factor authentication and concentrate on different aspects of authentication like authentication token, token management and communication protocols.

The approach discussed in this thesis uses password and client certificate to authenticate the user. The two tokens belong to two different authentication factors. Authentication data can be modified, altered and recorded during the communication and therefore a communication protocol must be able to provide resistance to the security threats. The proposed two-factor authentication mechanism uses TLS communication protocol. TLS is a Public Key Infrastructure (PKI) enabled service. Public key infrastructure plays the key role in providing information security properties. Two-way authentication is performed, by mandating both the client and server to present their certificates. The TLS handshake fails if the client and server do not trust each other or fail to provide the digital certificate.

Information security is based on aspects like availability, authenticity, confidentiality, data integrity and non-repudiation. The two-factor authentication solution, proposed in this thesis, has the capacity to provide the required information security. It provides authenticity using certificates as an authentication token. Confidentiality is ensured by encrypting the data sent on the wire. Also, a hash is sent along with the outgoing messages to provide data integrity.

The two-factor authentication solution leveraged the security level of the content management cloud applications. In order to prove the applicability of the suggested approach, I have integrated a two-factor authentication solution in the existing SaaS model. Through a series of test cases, I have proved that the solution was able to resist all the major security threats described during the thesis.

Bibliography

[AB13] A. M. Axel Buecker. Protecting Data Assets by Deploying a Multi-Factor Authentication Solution with End-to-End Encryption. Technical report, IBM, 2013. (Cited on page 21)

[All04] A. Allan. Passwords Are Near the Breaking Point. Technical report, Gartner, December 2004. URL http://www.donotspam.de/dokumente/gartner_passwords_breakpoint.pdf. (Cited on pages 6 and 10)

[CS09] L. B. S. G. C. E. F. G. S. J. U. V. S. Z. Carla Sadtler, Fabio Albertoni. *WebSphere Application Server V7.0 Security Guide*. IBM, 2009. (Cited on page 53)

[egia] Guidance on Multi-factor Authentication. (Cited on pages 10, 21 and 37)

[egib] Password Standard; State Services Commission New Zealand. URL http://authentication.webstandards.govt.nz/assets/password-standard.pdf. (Cited on page 8)

[ffi] Authentication in an Internet Banking Environment. URL http://www.ffiec.gov/pdf/authentication_guidance.pdf. (Cited on page 40)

[fipa] Security Requirements For Cryptographic Modules. Federal Information Processing Standards Publication 140-2. URL http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf. (Cited on page 40)

[fipb] Standards for Security Categorization of Federal Information and Information Systems. FIPS PUB 199. URL http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf. (Cited on page 26)

[Gen08] F. Gens. IT Cloud Services User Survey, pt.2: Top benefits & challenges. Technical report, IDC, 2008. URL http://blogs.idc.com/ie/?p=210. (Cited on pages 6, 13 and 15)

[hip] Understanding the Security & Privacy Rules associated with the HITECH and HIPAA Acts. Topic: Multifactor Authentication. URL http://www.secureit.com/resources/WP_HIPAA_HITECH_CMS_final_072811.pdf. (Cited on page 21)

[HK12] D. S. C. Heena Kharche. Building Trust In Cloud Using Public Key Infrastructure A step towards cloud trust. *International Journal of Advanced Computer Science and Applications*, Vol. 3:26–31, 2012. (Cited on page 44)

[ibt] Internet Banking and Technology Risk Management Guidelines Version 3.0. (Cited on page 21)

[MZ10] W. X. W. Q. A. Z. Minqi Zhou, Rong Zhang. Security and Privacy in Cloud Computing: A Survey. In *2010 Sixth International Conference on Semantics, Knowledge and Grids*. 2010. (Cited on page 13)

[nisa] Electronic Authentication Guideline. Recommendations of the National Institute of Standards and Technology. NIST Special Publication 800-63. URL http://csrc.nist.gov/publications/nistpubs/800-63-1/SP-800-63-1.pdf. (Cited on pages 10, 23, 35 and 36)

[nisb] Guide for Conducting Risk Assessments. NIST Special Publication 800-30. URL http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800_30_r1.pdf. (Cited on pages 6, 29, 30 and 31)

[nisc] Introduction to Public Key Technology and the Federal PKI Infrastructure. SP 800-32. (Cited on pages 42 and 44)

[nisd] Managing Information Security Risk. Organization, Mission, and Information System View. NIST Special Publication 800-39. URL http://csrc.nist.gov/publications/nistpubs/800-39/SP800-39-final.pdf. (Cited on pages 29 and 33)

[omb] E-Authentication Guidance for Federal Agencies. Memorandum to the Heads of all Departments and Agencies. URL http://www.whitehouse.gov/sites/default/files/omb/memoranda/fy04/m04-04.pdf. (Cited on pages 6, 25, 26, 27, 28 and 29)

[pci] Payment Card Industry Data Security Standard Requirements and Security Assessment Procedures. (Cited on page 21)

[rfca] HOTP: An HMAC-Based One-Time Password Algorithm. RFC 4226. URL http://www.ietf.org/rfc/rfc4226.txt. (Cited on page 39)

[rfcb] The TLS Protocol. RFC 2246. URL http://www.ietf.org/rfc/rfc2246.txt. (Cited on page 48)

[rfcc] TOTP: Time-Based One-Time Password Algorithm. RFC 6238. URL http://www.ietf.org/rfc/rfc6238.txt. (Cited on page 39)

[SC02] W. H. N. Suranjan Choudhury, Kartik Bhatnagar. *Public Key Infrastructure Implementation and Design*. Wiley, 2002. (Cited on pages 6, 41, 43, 44 and 56)

[SS11] V. S. S. Sengupta, V. Kaulgud. Cloud Computing Security–Trends and Research Directions. *2011 IEEE World Congress on Services*, pp. 524 – 531, 2011. (Cited on pages 3, 7 and 13)

[stoa] Framework Mapping of Technical/Organisational Issues to a Quality Scheme STORK D2.1. (Cited on pages 6, 24 and 25)

[stob] Quality Authenticator Scheme STORK D2.3. (Cited on pages 6, 31 and 32)

[WM10] T. D. K. D. Wanli Ma, Campbell J. Password Entropy and Password Quality. In *Network and System Security (NSS), 2010 4th International Conference*. Sept. 2010. (Cited on page 9)

[YJ04] A. R. G. A. Yan J., Blackwell A. Password memorability and security: empirical results. In *Security & Privacy, IEEE (Volume:2 , Issue: 5)*. Sept.-Oct. 2004. (Cited on page 9)

Declaration

All the work contained within this thesis, except where otherwise acknowledged, was solely the effort of the author. At no stage was any collaboration entered into with any other party.

(Umair Ashraf)