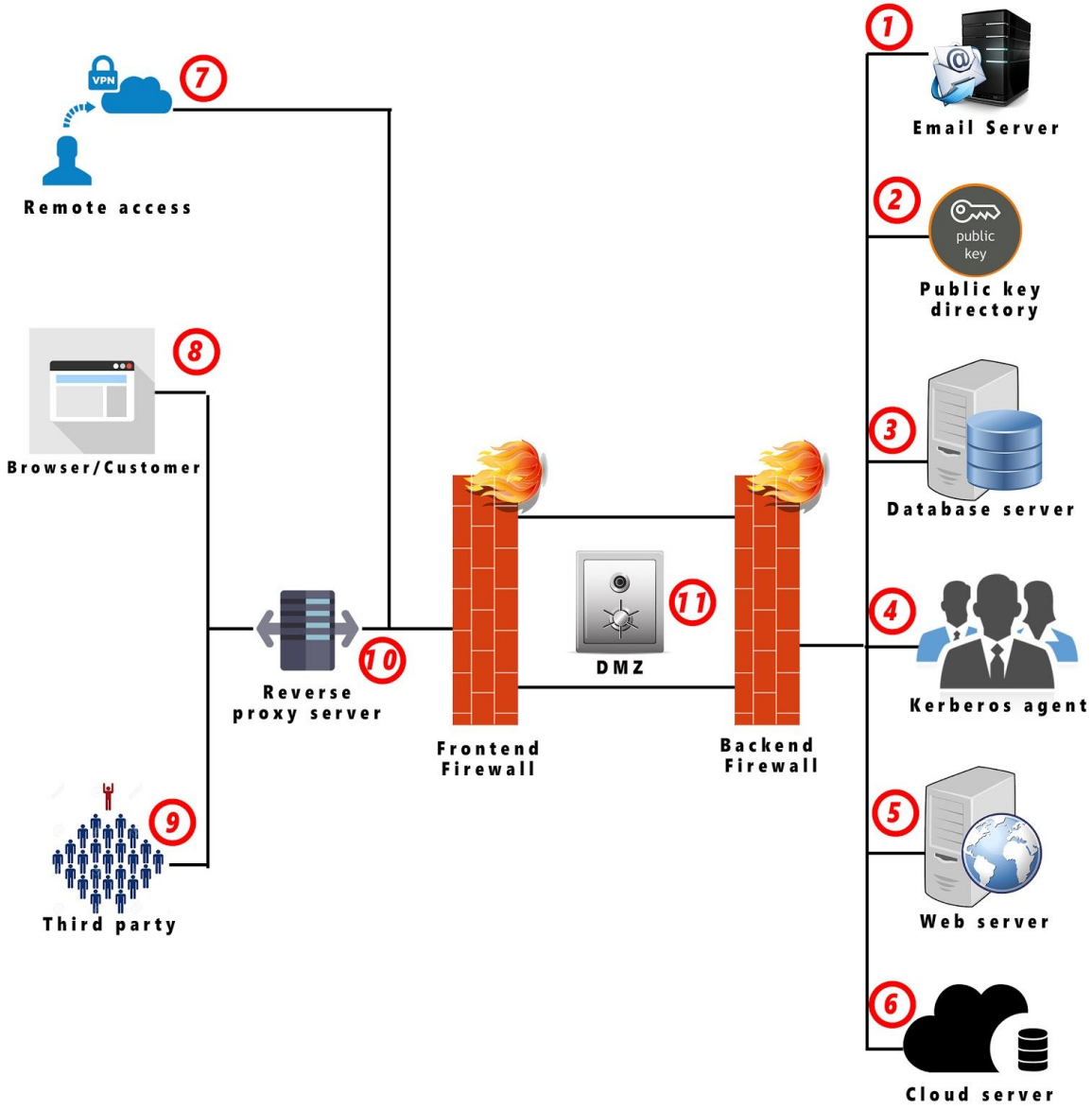
Cybersecurity Investigations and laws

CS 698

Final Assignment

Hawk eyes: **Adhithya Sivanesh (as3423), Gautam Pandey (gp88), Ketaki kakade (kk524)**

Taking into consideration the confidentiality, integrity and authenticity of data flowing in a network which could be vulnerable to attacks and which may compromise the integrity of a company we have come up with the following solutions. The paper also suggests security measures that should be taken in order to protect the different parts of the company.



The explanations for each of the block in the network diagram has been explained based on the numbering given in the diagram.

1]Email server

Email represents the ultimate in communication convenience but it is coupled with lack of security. When SMTP was developed – back when the internet was a much more trusting environment- security wasn't a major concern. AS a result, users today transmit confidential information across the internet without realizing just how vulnerable the data is. If you send your emails without encrypting them then you are basically sending them in plain text. If the mail happens to be intercepted when in route, it could be easily read by anyone. Efficient encryption systems were developed for this reason and can either produce an encrypted email or encrypt the sending of an email.

In order to encrypt the email yourself, there are useful programs such as the open source software PGP (Pretty Good Privacy), which uses asymmetric encryption. Whereas traditional methods for coding and decoding access the same key, asymmetric encryption uses two keys – a private key and a public key. The public key can be forwarded without encryption to the intended contact without worry since only the private key opens the protected emails. Conversely, the user must keep the public key of the contact in order to do their part in encrypting the email. Additionally, you can use the Transfer protocol SSL, or rather TLS, to encrypt the transfer of the emails.

The transfer protocol Transfer Layer Security (TLS) is the universal tool required for sending email content in a secure fashion. This tool is better known under it former name Secure Socket Layer (SSL). An email with SSL or TLS encryption is characterized by its content not being able to be decoded by third parties since they don't have access to key required for encryption. Therefore it doesn't matter whether the email is sent or retrieved through an email client, such as Outlook, or through a web browser. This is due to the encrypted emails being illegible to any snoopers during the whole transfer process between mail server and client (internet browser). The contents are then converted back into plaintext when received by the recipient.

Is encrypting a mail server enough?

Encrypting information is only one aspect of security, the other is knowing the identity of the person. If two people choose to communicate by email, how can they be sure that any of the communications were transmitted without being tampered with? Also, if a website owner wants to be sure that only a specific user gains access to secured information, how can this assurance be provided? The simple answer is Digital Certificates. Certificates on the mail servers are a pretty big deal because without one:

1. There's no way to identify that the mail server you're connecting to, is actually the correct mail server.

2. Any email sent between your browser or email client and the server are not encrypted and could be intercepted.

Without a certificate, you leave yourself open to a man-in-the-middle (MITM) attack, whereby malicious parties could insert themselves between you and your mail server to intercept and access your emails. *However,* while an SSL Certificate will protect your emails in transit to and from your server, it does nothing to protect your emails as they pass through other servers, which may not be have SSL. Additionally, securing your mail server doesn't protect the emails at rest.

Authentication & data integrity

The cryptography technology underlying S/MIME means that only the intended recipient of your email can actually read it. In addition to encryption, S/MIME enables you to add digital signatures to your emails too. This means not only can you protect your emails from falling into the wrong hands, but also:

1. Prove that your email actually came from you (i.e. is not a spoof or phishing email) – the digital signature is applied with your private key and verified with your public key, which are unique to you. Your identifying information is included in the signature, which most email clients display prominently.

2. Prevent changes to your email after it has been sent – when a digital signature is verified (in this case, when a recipient opens your email), a process takes place behind the scenes that compares the email contents at that moment to when the signature was applied. If the content doesn't match, an error will display so your recipient knows something is wrong and not to trust the contents of the email. Figure 1 shows example of a Digital certificate.

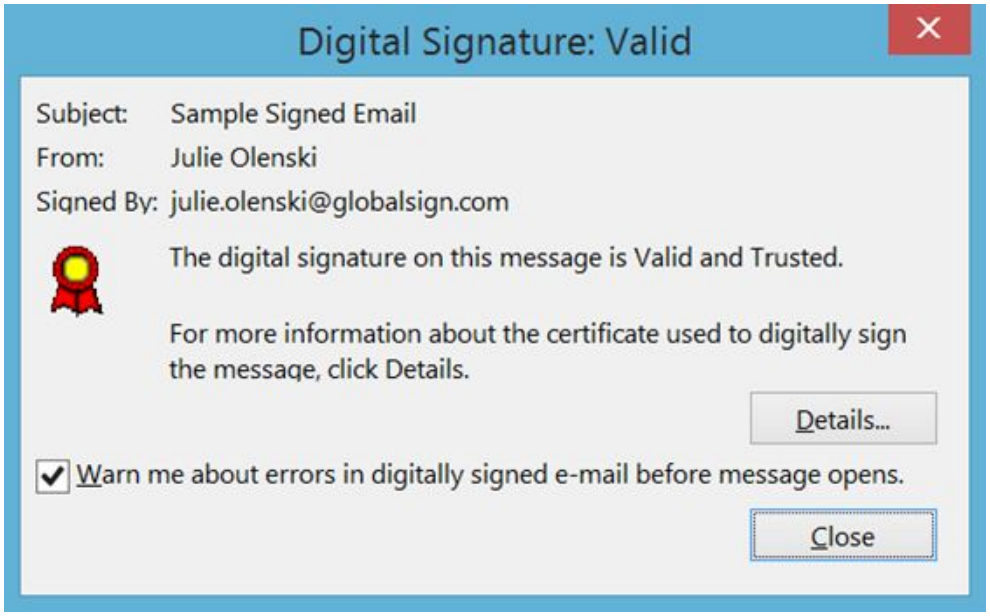


Figure 1

2]**Public key directory:**

The public key directory is a central repository for exchanging information required for authentication. The authentication involved is ensuring the authenticity of the users and integrity of their corresponding messages. The systems needs to be placed in a secure environment. This is usually done by creating several segmented and separated environments by VLAN's (Virtual Local Area Networks). These VLAN's can only be accessed by routers and firewalls.
The data stored in the public key directory are usually in an encrypted form. The usually used encryptions standards are as follows:
1) Triple DES: Triple DES uses three individual keys with 56 bits each. The total key length adds up to 168 bits, but experts would argue that 112-bits in key strength is more like it.
2) Blowfish: It is a symmetric cipher that splits messages into blocks of 64 bits and encrypts them individually. It is known for its speed and effectiveness, and is also freely available in the public domain.
3) Twofish: Keys used in this algorithm may be up to 256 bits in length and as a symmetric technique, only one key is needed. Twofish is regarded as one of the fastest of its kind, and ideal for use in both hardware and software environments. Its free availability makes it even more famous.
4) AES: The Advanced Encryption Standard (AES) is the algorithm trusted as the standard by the U.S. Government and numerous organizations. Although it is extremely efficient in 128-bit form, AES also uses keys of 192 and 256 bits for heavy duty encryption purposes.
On top of these encryption standards, the organization could also make use of **honey encryption**. Honey Encryption is a security tool that makes it difficult for an attacker who is carrying out a brute force attack to know if he has correctly guessed a password or encryption key.

3]**Database servers**

Database server is the term used to refer to the back-end system of a database application using client/server architecture. The back-end, sometimes called a database server, performs tasks such as data analysis, storage, data manipulation, archiving, and other non-user specific tasks.Database encryption can generally be defined as a process that uses an algorithm to transform data stored in a database into "cipher text" that is incomprehensible without first being decrypted. There are various types of database encryption methods provided by database security companies. To select proper encryption method for your DBMS, you first know what types of database encryption methods are available.

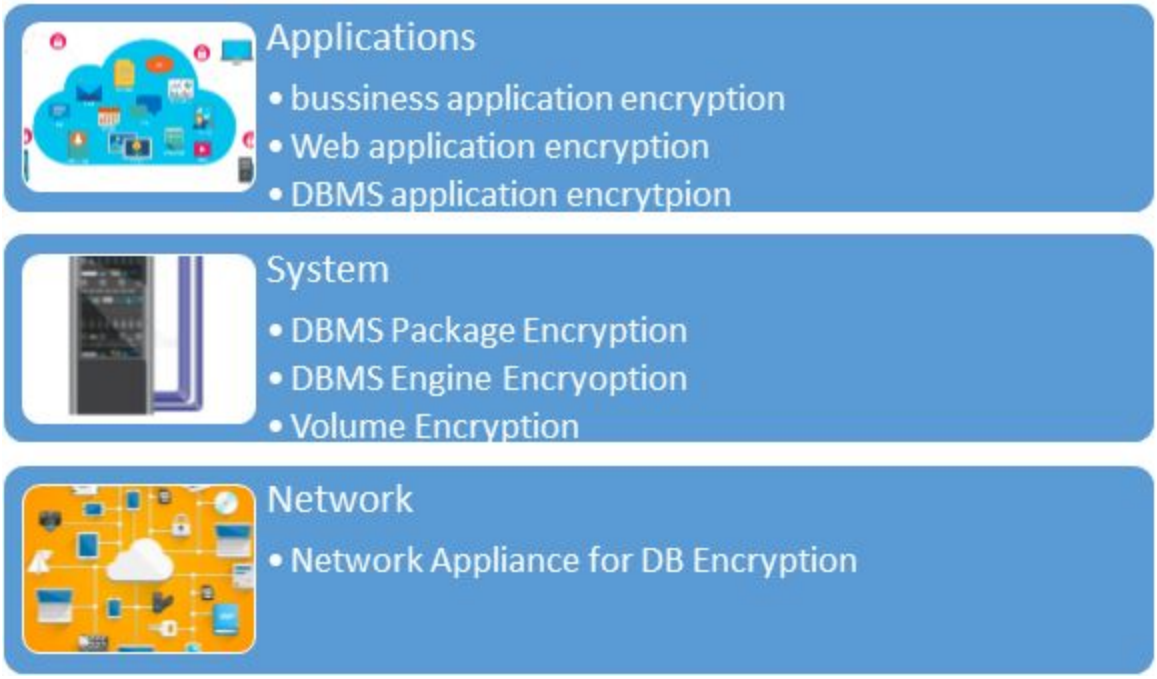


Figure 2

1. **Business Application Encryption (BA)**

This type of DB encryption is applying encryption/decryption API on the application server or business application server. There are no restrictions of applicable DBMS, and the method does not cause strain to the DBMS. However, this method can be time-consuming because every query related to encryption data must be modified.

2. **DBMS Application Encryption (DA)**

This type of DB encryption performs encryption/decryption by applying a DBMS product module as a form of API. This method also has no restrictions of applicable DBMS and can be applied to various DBMS (same as the BA method above). The need of query modification is also similar. However, this method is differentiated from the BA method because the encryption/decryption function can be used in the DBMS administration tool.

3. **DBMS Package Encryption (DP)**

This type of DB encryption performs encryption/decryption by installing a product module on DBMS. Since the module is installed on the DBMS as a form of package, this method can support indexing of encrypted columns and can be implemented easily without additional query modifications. Also, audit functions through GUI and access control can be provided as an

integrated security function. This type of database encryption is regarded as the ‘Plug-In Method”

4. **DBMS Engine Encryption (DE)**

This type is the most evolved form of DB encryption methods. Since encryption/decryption is performed at the engine-level of DBMS, it is most easily implemented and fast encryption/decryption performance is possible. This method requires engine-level modification, therefore it can only be provided in a few cases. First, it can be provided by DBMS vendors. Second, it can be provided through open source DBMS such as MySQL or MariaDB. Third, it can be provided by collaboration between database encryption companies and database companies. In case of solutions provided by DBMS vendors, such as Oracle’s TDE or SQL Server TDE, it only provides encryption functions and no access control or audit functions. In this case, you need to purchase a separate package to apply integrated security. This type of database encryption is regarded as the ‘Transparent Data Encryption Method’.

Types of encryption in Databases

1. **Transparent/External database encryption**

Transparent data encryption (TDE) is used to encrypt an entire database, which therefore involves encrypting “data at rest”. TDE ensures that data on physical storage media cannot be read by malicious individuals that may have intention to harm them. The most important attribute of TDE is transparency. The transparent element of TDE has to do with the fact that TDE encrypts on "the page level", which essentially means that data is encrypted when stored and decrypted when it is called into the system's memory**.** The contents of the database are encrypted using a symmetric key that is often referred to as a "database encryption key".

2. **Column-level encryption**

A typical relational database is divided into tables that are divided into columns that each have rows of data. Whilst TDE usually encrypts an entire database, column-level encryption allows for individual columns within a database to be encrypted. It is important to establish that the granularity of column-level encryption causes specific strengths and weaknesses to arise when compared to encrypting an entire database. Firstly, the ability to encrypt individual columns allows for column-level encryption to be significantly more flexible when compared to encryption systems that encrypt an entire database such as TDE. Secondly, it is possible to use an entirely unique and separate encryption key for each column within a database. This effectively increases the difficulty of generating rainbow tables which thus implies that the data stored within each column is less likely to be lost or leaked. The main disadvantage associated with column-level database encryption is speed, or a loss thereof.

3. **Encrypting File System (EFS)**

EFS can encrypt data that is not part of a database system, which implies that the scope of encryption for EFS is much wider when compared to a system such as TDE that is only capable of encrypting database files. Whilst EFS does widen the scope of encryption, it also decreases database performance and can cause administration issues as system administrators require operating system access to use EFS. Due to the issues concerning performance, EFS is not typically used in database applications that require frequent database input and output. In order to offset the performance issues it is often recommended that EFS systems be used in environments with few users.

Hashing

Hashing is used in database systems as a method to protect sensitive data such as passwords; however it is also used to improve the efficiency of database referencing. Inputted data is manipulated by a hashing algorithm. The hashing algorithm converts the inputted data into a string of fixed length that can then be stored in a database. Hashing systems have two crucially important characteristics that will now be outlined. Firstly, hashes are "unique and repeatable. Secondly, hashing algorithms are not reversible. In the context of database encryption, hashing is often used in password systems. When a user first creates their password it is run through a hashing algorithm and saved as a hash. When the user logs back into the website, the password that they enter is run through the hashing algorithm and is then compared to the stored hash. Given the fact that hashes are unique, if both hashes match then it is said that the user inputted the correct password. One example of a popular hash function is SHA (Secure Hash Algorithm).

Salting

One issue that arises when using hashing for password management in the context of database encryption is the fact that a malicious user could potentially use an Input to Hash table rainbow table for the specific hashing algorithm that the system uses. This would effectively allow the individual to decrypt the hash and thus have access to stored passwords. A solution for this issue is to 'salt' the hash. Salting is the process of encrypting more than just the password in a database. The more information that is added to a string that is to be hashed, the more difficult it becomes to collate rainbow tables. This increase in the complexity of a hash means that it is far more difficult and thus less likely for rainbow tables to be generated. This naturally implies that the threat of sensitive data loss is minimized through salting hashes. Without salts, an attacker who is cracking many passwords at the same time only needs to hash each password guess once, and compare it to all the hashes. However, with salts, each password will likely have a different salt; so each guess would have to be hashed separately and compared for each salt, which is considerably slower than comparing the same single hash to every password.

4] **Kerberos agent**

A Kerberos agent is a third party agent that is often used when there are multiple number of users and their corresponding public keys are to be exchanged without any external intervention. The Kerberos agent uses tickets to authenticate users to services in TCP/IP networks over insecure channels. It uses a trusted third-party (TTP). It is based on symmetric key cryptography. It allows the end users to authenticate themselves on an open (unprotected) network. It uses a Key Distribution Center (KDC) to share a secret key (master key) with each end user (this is a long-term key). The commonly used encryption standards are Data Encryption Standard (DES) encryption algorithm (with 56-bit keys) or Advanced Encryption Standard (AES).

5]**Web server**

Web servers rely upon strong encryption to protect the data sent between users and the Web server. In the absence of strong encryption, any such communications are vulnerable to eavesdropping and modification. This threat could potentially undermine the confidentiality and integrity of financial transactions or other sensitive data that is exchanged with end users. There are two steps to ensuring strong encryption is being used to protect Web communications. One requires the use of a secure cryptographic protocol, and the other requires that the selected protocol make use of strong cipher algorithms. The cryptographic protocol describes how the Web user and server set-up communications and exchange encryption keys while the cipher algorithm specifies the mathematical operations used to encrypt and decrypt data.

There are two main cryptographic protocols in use on the Web today; the Secure Sockets Layer (SSL) and Transport Layer Security (TLS). When configuring the protocols used on a Web server, an organization should choose to support both TLS and SSL version 3. Earlier versions of SSL have critical vulnerabilities and should not be used. Now that you are using SSL the web server can start serving files via HTTPS. Using HTTPS doesn't just mean that your traffic is encrypted—encryption is only half of the story and it's useless without Authentication. What good is it to encrypt something between two parties if you can't be sure of the identity of the person to whom you're talking? Consequently, being able to serve HTTPS traffic means you must possess a cryptographic certificate attesting to your identity. Acquiring such a certificate requires you to identify to one of the many Certificate Authority,CAs.

Considerations in choosing a Web server include how well it works with the operating system and other servers, its ability to handle server-side programming, security characteristics, and the particular publishing, search engine and site building tools that come with it. The most leading web servers available today are Apache HTTP server, Internet Information Services(IIS),

Lighttpd, Sun Java system web server, Jigsaw server. Figure 3 shows a website without a valid certificate or that the certificate is from an untrusted third party.

Figure 3.

6] **Cloud Server:**

There are two main aspects when we talk about encrypting data on the cloud.
At-rest data -Data at rest generally refers to data stored in storage-typically on disks or cloud servers. This is one of the most important components of securing the cloud architecture. Data stored in the cloud needs to be updated and stored in storage very frequently and most businesses using the cloud rely on this data being secure on the cloud. There are a variety of encryption techniques used by different clouds, most common are AES, RSA, DES with some of the front runners like google cloud, microsoft azure and amazon AWS are using these.

Data in-transit- The use of cloud means that users have the ease of access to the data stored and can retrieve process and store this data on the cloud frequently and readily. This means that there is going to be a lot of data in transit almost always. This data is vulnerable to a man in the middle attack if not encrypted using sophisticated encryption techniques. SSL and TLS are most commonly used.

The diagram below gives a brief overview of how AWS a popular cloud platform encrypts data.

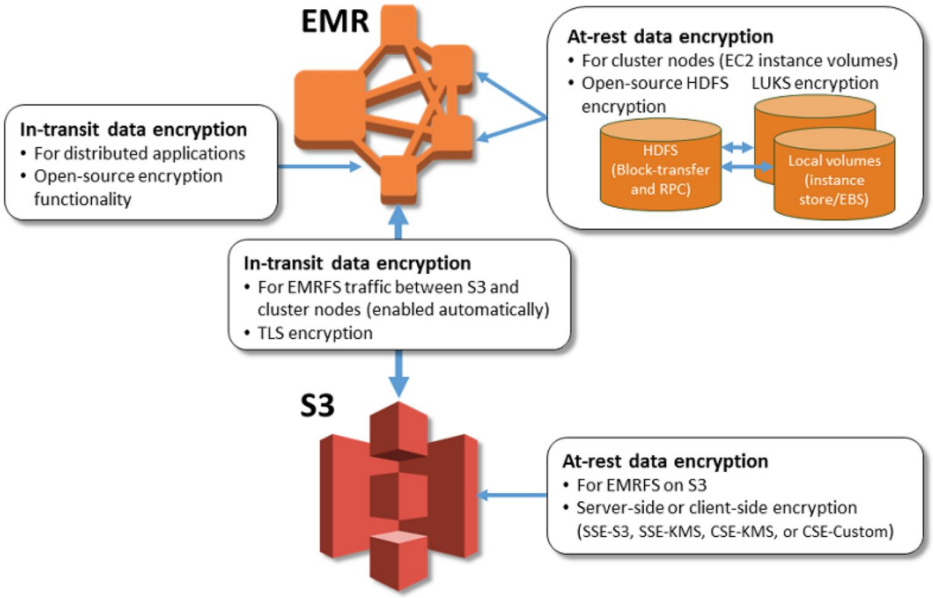


Figure 4.

Users have the option to add to the layers of encryption that are existent and configure the security groups.

In addition asymmetric keys are also used and public and private keys are required to verify the confidentiality as well as the authenticity of the data in transit to and from cloud.

7]Remote access through VPN

A remote-access VPN allows individual users to establish secure connections with a remote computer network. Those users can access the secure resources on that network as if they were directly plugged into the network's servers. An example of a company that needs a remote-access VPN is a large firm with hundreds of salespeople in the field. Another name for this type of VPN is virtual private dial-up network (VPDN), acknowledging that in its earliest form, a remote-access VPN required dialing in to a server using an analog telephone system. There are two components required in a remote-access VPN. The first is a network access server (NAS, usually pronounced "nazz" conversationally), also called a media gateway or a remote-access server (RAS). A NAS might be a dedicated server, or it might be one of multiple software applications running on a shared server. It's a NAS that a user connects to from the Internet in order to use a VPN. The NAS requires that user to provide valid credentials to sign in to the VPN. To authenticate the user's credentials, the NAS uses either its own authentication process or a separate authentication server running on the network.

An alternative for VPN is Remote Desktop Protocol (RDP), Software that allows you to connect to your work computer from your home computer and have access to all of your programs, files, and network resources—just as though you were in front of your computer at work.

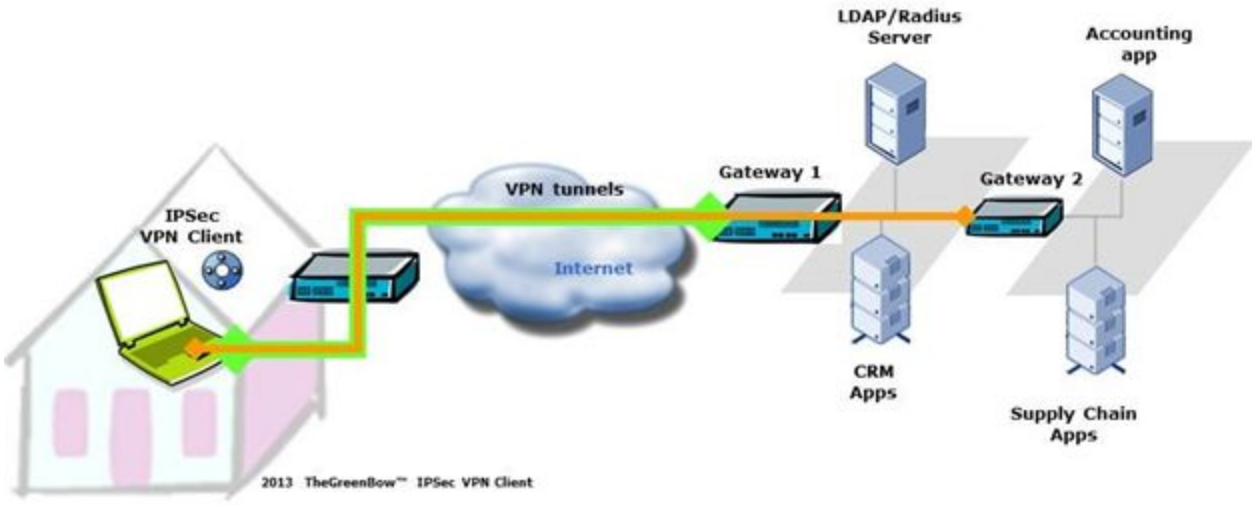


Figure 5.

8] **Browser/ Customer:**

The goal here is to provide confidentiality and integrity for HTTPS traffic.
HTTPS(Hypertext Transfer Protocol Secure) is a communication protocol for secure communication that makes use of Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL). TLS/SSL is initialized at the session layer, then works at the presentation layer. Most web browsers incorporate functionality of the OSI layers 5, 6 and 7. When referring to HTTPS it will be an implementation of SSL/TLS in the context of the HTTP protocol. SSL/TLS will then be implemented in the browsers (and web server) to provide security of the in data traffic.

9] **Third Party:**

It is very common for businesses and applications in the modern day to use a variety of third party applications to store and process data. It is very important that the data being handled by the third parties be secured as it is intellectual property that is very valuable to the organization.

There are a number of measures which can be use to protect and secure your data when third party applications come into play.

Review third-party apps access to API scopes
This means that you must only share the pieces of data or core services that are required by the third party and grant the appropriate access levels to third party agents. This is typically done using oauth tokens.

Create and maintain a whitelist of trusted third party vendors -
It is very important to maintain a list of authorized third party apps that may need access to your data or core services. This solves two purposes, ease of access to third parties as well as protection against those who are not on the list and may have malicious intents. You must also regularly review and update this list to add new authorized users or delete the users who no longer require access.

10] **Reverse proxy server:**

A reverse proxy server is a type of proxy server that typically sits behind the firewall in a private network and directs client requests to the appropriate backend server. It provides an additional level of abstraction and control to ensure the smooth flow of network traffic between clients and servers. The good thing about using a reverse proxy is it provides extra level of security and anonymity by intercepting requests headed for the backend servers, a reverse proxy server protects their identities and acts as an additional defense against security attacks. It also ensures that multiple servers can be accessed from a single record locator or URL regardless of the structure of your local area network. Since requests could come from various devices, or third party organizations, it may or may not be in an encrypted form and often they may also involve confidential data that must not be disclosed. So it is a good strategy to re-encrypt the backend connection. You don't need to buy another certificate. You can use self-signed certificates for the backend. So the organization could use an upstream ssl to secure traffic to upstream servers.

11] **DMZ:**

DMZ stands for Demilitarized zone. here are various ways to design a network with a DMZ. The two most common methods are with a single or dual firewalls. These architectures can be expanded to create very complex architectures depending on the network requirements. A more secure approach is to use two firewalls to create a DMZ. The first firewall also called the perimeter firewall is configured to allow traffic destined to the DMZ only. The second or internal firewall only allows traffic from the DMZ to the internal network. This is considered more secure since two devices would need to be compromised before an attacker could access the internal LAN. As a DMZ segments a network, security controls can be tuned specifically for each segment. For example a network intrusion detection and prevention system located in a

DMZ that only contains as Web server can block all traffic except HTTP and HTTPS requests on ports 80 and 443.

Conclusion and the Future of Encryption

Expert observers are hopeful that a new method called Honey Encryption will deter hackers by serving up fake data for every incorrect guess of the key code. This unique approach not only slows attackers down, but potentially buries the correct key in a haystack of false hopes. Then there are emerging methods like quantum key distribution, which shares keys embedded in photons over fiber optic, that might have viability now and many years into the future as well.

Whether it's protecting your email communications or stored data, some type of encryption should be included in your lineup of security tools. Successful attacks on victims like Target show that it's not 100 percent bulletproof, but without it, you're offering up convenient access to your data.

Cyber attacks are constantly evolving, so security specialists must stay busy in the lab finding out new schemes to keep them at bay!