Key management – refers to distribution of cryptographic keys, mechanisms used to bind identity to a key and generation ,maintenance and revoking of such keys.

Assume that authentication completed and identity assigned.

Problem is to propagate authentication to other principals and systems.

A key bound to identity Bob is really Bob’s key.

X -> Y :{m}k

Entity X sends Y a message m enciphered with key k.

K\_alice and K\_Bob – keys belonging to Alice and Bob

k-secret key (for symm cryptosystem)

e – public key

d – pvt key

a||b – a and b are concatenated.

Interchange key – An interchange key is a cryptographic key associated with with a principal for communication.

an "Interchange Key" is a cryptographic key used for securing data during transmission between two or more entities.

Types of interchange keys – symmetric and asymmetric keys

It plays a crucial role in ensuring the confidentiality and integrity of the data being exchanged.

It plays a crucial role in ensuring the confidentiality and integrity of the data being exchanged.

A session key is a cryptographic key associated with communication itself.

Alice has a cryptographic key used to exchange information with Bob. This key should not change over interactions with Bob.

If Alice communicates twice with Bob she does not want to use same key for enciphering the messages.

Reduces likelihood of eavesdropper breaking cipher.

She will generate a key for single session .Key enciphers data and is discarded when session ends. This is session key.

Session keys prevent forward searches.

Forward searches occur when set of plaintext messages small.

Encipher all plaintext messages using target’s public key. When ciphertext intercepted ,it is compared with these texts.

Key exchange

Goal – enable Alice to communicate secretly with Bob using a shared cryptographic key.

Symmetric cryptographic key exchange

Suppose Alice and Bob wishes to communicate.

How do they agree on a common key?

Alice sends key to Bob, Eve the eavesdropper will see it and can read traffic between them.

Rely on a trusted third party, Cathy.

Alice and Cathy share a secret key ,Bob and Cathy share a different secret key.

Goal – provide a secret key Alice and Bob share.

Alice -> Cathy :{request for session key to Bob }k\_Alice

Cathy ->Alice : {k session} k\_Alice ||{k\_session} k\_Bob

Alice -> Bob : {k session}k\_Bob

1. **Alice -> Cathy: {request for session key to Bob}k\_Alice**
   * Alice sends a message to Cathy requesting a session key that she can use to communicate securely with Bob.
   * This message is encrypted with Alice's secret key, **k\_Alice**, which she shares with Cathy. Only Cathy, who possesses the corresponding decryption key, can decrypt and read this message. This ensures that even if someone else intercepts this message, they won't be able to understand it.
2. **Cathy -> Alice: {k\_session}k\_Alice || {k\_session}k\_Bob**
   * Cathy responds to Alice's request by sending a message that contains the session key (**k\_session**) encrypted twice: once with Alice's secret key and once with Bob's secret key.
   * The first part of Cathy's message (**{k\_session}k\_Alice**) can be decrypted by Alice using her own secret key, allowing her to obtain the session key.
   * The second part (**{k\_session}k\_Bob**) is intended for Bob. Alice can't decrypt this part as she doesn't have Bob's secret key.
3. **Alice -> Bob: {k\_session}k\_Bob**
   * Alice forwards the part of Cathy's message that contains the session key encrypted with Bob's secret key to Bob.
   * Bob then uses his secret key to decrypt this message and obtain the session key.
   * Now, both Alice and Bob have the same session key (**k\_session**), which they can use to communicate securely.

This process ensures that Alice and Bob end up with a shared secret key without ever having transmitted it in plain text over the network.

It also ensures that even if an eavesdropper like Eve is listening to the communication, she cannot obtain the session key, as she does not have access to either Alice's or Bob's secret keys. This method leverages the trust that Alice and Bob have in Cathy and the secure key pairs each of them shares with her.

Public cryptographic key exchange

Alice uses Bob’s public key to encipher session key she generates.

Alice -> Bob : {k\_session}e\_Bob

Bob deciphers message using pvt key.

Eve can forge such a message

To fix

Sign the session key using private key

Alice ->Bob :{Alice ||{k\_session} d\_Alice} e\_Bob

Bob decrypts in 2 stages

using his pvt key and Alice public key.

Cryptographic key infrastructure

Because symmetric cryptosystems use shared keys, not possible to bind identity to a key.

Instead 2 parties need to agree on shared key.

In public key cryptosystems – association between key and principal critical as it determines public key used to encipher messages for secrecy.

If binding erroneous someone else can read message.

Principal represented by a name

A yellow text on a white background

Description automatically generated

Certificate – token that binds identity to a cryptographic key.

Create a message containing representation of identity, corresponding public key and having a trusted authority sign it.

C\_Alice = { e\_Alice || Alice || t } d\_Cathy

Bob obtains Alice’s certificate. He knows Cathy’s public key, he can decipher certificate. He checks timestamp to determine when certificate was issued.

He can determine whether certificate can be trusted.

A principal is a unique entity. An identity specifies a principal.

Authentication binds principal to an identity.

Certificate signature chains

Cerificates – means of getting other people to trust our public key.

Once they do that can setup a public key infrastructure.

Everybody can happily communicate with everybody else securely.

If you want someone across world to trust public key, who should sign the public key.

Have somebody who trusts Alan or sits close to Alan while he generates his key pair.

Can also present them with passport so it really looks like me. Need trust for cryptography to work.

Alan goes to betty ,Betty does the signing. How far does this get us ? Not very far if Betty is not friend of Zoe.

Your description touches on several key concepts related to Public Key Infrastructure (PKI) and trust models in cryptography. Let's break down these concepts for clarity:

1. **Certificates**: These are digital documents that bind a public key to an individual's identity. Certificates are essential for others to trust a public key. They typically contain the public key, the identity of the key holder, and other metadata. They are signed by a trusted authority to assert the validity of the information.
2. **Public Key Infrastructure (PKI)**: This is a framework that provides services and standards to enable secure, encrypted communication and authentication over networks. PKI involves the use of certificates, digital signatures, and certificate authorities to establish and maintain a trustworthy networking environment.
3. **Trust Models in Cryptography**:
   * **Direct Trust**: This is where individuals trust each other directly. For instance, if you know someone personally and trust them, you might directly trust their public key without needing an intermediary.
   * **Third-Party Trust**: In most real-world scenarios, direct trust isn't feasible, especially over large networks or when communicating with strangers. This is where third-party trust comes in. A trusted third party, often a Certificate Authority (CA), verifies identities and issues certificates.
4. **Certificate Signing and Trust Chains**:
   * **Scenario**: Alan needs to prove his identity and his public key. He could go to someone who knows him (like Betty) to verify his identity and sign his public key.
   * **Limitations**: If Betty signs Alan's key, it only extends trust to those who trust Betty. If Zoe doesn't know or trust Betty, she won't trust Alan's key just because Betty signed it.
   * **Solution**: In larger systems, a hierarchical trust model is often used where a central CA (or a hierarchy of CAs) issues certificates. These CAs are widely trusted, and their signatures on certificates extend trust more broadly.
5. **Identity Verification**: When someone goes to get their public key signed (like Alan going to Betty), they often need to present proof of identity (like a passport). This is to ensure that the public key is indeed tied to the correct individual.
6. **Challenges in PKI**:
   * Trust in the CA: The entire system hinges on the trustworthiness of the CA. If the CA is compromised, the trust in the entire network can be undermined.
   * Distribution and Revocation: Distributing certificates and handling the revocation of compromised keys are also significant challenges in PKI systems.

In summary, PKI and certificates are crucial in establishing trust in digital communications. They involve verifying identities and associating public keys with those identities, typically through a trusted third party like a CA. The effectiveness of this system depends largely on the trustworthiness and management of the CAs and the certificates they issue.

Top of Form

**Trust in the Public Key**:

* **Essential for Verification**: The entire process hinges on the recipient trusting that the public key used for verification genuinely belongs to the sender. If there's doubt about the ownership of the public key, the integrity of the data cannot be assured.
* **Implications of Mistrust**: If the public key is not trusted, there's no guarantee that the data hasn't been intercepted and altered by a third party, with a new digital signature applied using a different private key. In such a case, the recipient might wrongly believe the data is authentic and unaltered.

Trusting a public key is fundamental in cryptography, especially in secure communications and data encryption. Here's why trust in a public key is essential:

For data integrity, confidentiality

1. **Authentication**: Trusting a public key ensures that you are communicating with the intended person or entity. In digital communications, you often need to be certain that the party on the other end is who they claim to be. A trusted public key verifies the identity of the communicator, thereby preventing impersonation or "man-in-the-middle" attacks.
2. **Data Integrity**: When data is signed using a private key, its corresponding public key can be used to verify that the data hasn't been tampered with. This process relies on the trust that the public key genuinely belongs to the sender. If you don't trust the public key, you can't be certain that the data is intact and hasn't been altered during transmission.
3. **Confidentiality**: In encryption, a sender might encrypt data using the recipient's public key. Trust in this public key is crucial because if the public key doesn't actually belong to the intended recipient, someone else could decrypt and access the data. This is particularly important for sensitive information.
4. **Non-repudiation**: Digital signatures provide non-repudiation, meaning the signer cannot deny the authenticity of their signature on a document or a message sent. This is only reliable if the public key used to verify the signature is trusted to belong to the signer.
5. **Secure Transactions**: In financial transactions or any exchange involving sensitive information, trusting public keys is vital. It ensures that the transaction is secure and both parties are confident in the exchange's integrity.
6. **Trust Ecosystem**: In larger networks and systems, like the internet, trust in public keys allows for a broader trust ecosystem. Secure websites, email encryption, digital signatures on software, and many more applications rely on this trust. It's the basis of a secure and functioning digital world.

In summary, trusting a public key is a cornerstone of secure digital communications. It enables authentication, ensures data integrity and confidentiality, provides non-repudiation, and is essential for secure transactions. Without trust in public keys, the risks of data breaches, fraud, and malicious activities increase significantly.

PGP stands for "Pretty Good Privacy." It is an encryption program that provides cryptographic privacy and authentication for data communication. PGP is widely used for securing emails, files, and entire disk partitions and to increase the security of email communications. Here's an overview of how PGP works and its key features:

1. **Encryption and Decryption**:
   * **Asymmetric Encryption**: PGP uses asymmetric encryption where a message is encrypted with the recipient's public key and can only be decrypted by the corresponding private key, which is kept secret by the recipient.
   * **Symmetric Encryption for Messages**: For efficiency, PGP often encrypts the actual message using symmetric encryption with a one-time-use secret key. It then encrypts this secret key with the recipient's public key.
2. **Digital Signatures**:
   * PGP allows users to sign their messages with their private key. Recipients can then use the sender's public key to verify the signature, ensuring the message's authenticity and integrity.
3. **Web of Trust**:
   * Unlike traditional Public Key Infrastructure (PKI) models that rely on a central authority to verify public key ownership, PGP uses a decentralized trust model known as the "Web of Trust."
   * In the Web of Trust, individual users can sign others' public keys, vouching for their authenticity. This creates a network of trust relationships where the authenticity of a public key can be ascertained through its endorsements by other trusted users.
4. **Key Management**:
   * PGP involves careful management of public and private keys. Users typically keep their private key secured with a passphrase and distribute their public key to those they wish to communicate with securely.
5. **Applications**:
   * **Secure Email**: PGP is most commonly used to secure email communications. Emails are encrypted with the recipient's public key and can only be decrypted by the recipient with their private key.
   * **File Encryption**: PGP can also encrypt files and documents, ensuring that only authorized individuals can access the content.
6. **Compatibility and Standards**:
   * PGP has been standardized in the form of OpenPGP, which is an open standard for encryption software. This ensures compatibility across different implementations of PGP.

In summary, PGP is a powerful tool for securing digital communications. Its use of encryption, digital signatures, and a decentralized trust model makes it a robust solution for ensuring privacy, authenticity, and integrity in the digital world

Six degrees of separation

Only ever 6 steps of acquaintance between one individual in the world and another individual in the world.

Certificate authority – entity that issues certificates.

PGP = go to other people that trust me and ask them to sign my key.

PGP

1. **Personal Verification**:
   * When you create a digital identity (like a PGP key pair), you have your own public and private keys. Your public key needs to be trusted by others to be useful in secure communications.
   * To build trust in your public key, you go to people who already know and trust you. These individuals can be friends, colleagues, or acquaintances who also use PGP.
2. **Signing Each Other's Public Keys**:
   * These people, after verifying your identity (often in person or through another trusted method), sign your public key with their private key. This is akin to vouching for your identity and the authenticity of your public key.
   * In return, you can do the same for their public keys if you trust them. This mutual verification contributes to a network of trusted connections.

A digital signature is a cryptographic mechanism used to verify the authenticity and integrity of a message, software, or digital document. It's akin to a handwritten signature or a stamped seal, but it's much more secure and designed for the digital world. Here's a breakdown of how it works and why it's important:

1. **Creation of a Digital Signature**:
   * **Use of Asymmetric Cryptography**: Digital signatures use asymmetric cryptography, which involves a pair of keys: a private key and a public key. The private key, which is kept secret by the signer, is used to create the signature. The public key, which is available to anyone, is used to verify the signature.
   * **Signing Process**: When a document or message is signed digitally, an algorithm (like RSA, DSA, or ECDSA) takes the data in the message and the signer's private key to generate a unique digital signature.
2. **Verification of a Digital Signature**:
   * **Public Key**: To verify a digital signature, the recipient uses the signer's public key. The verification process ensures that the signature was created using the corresponding private key and that the message or document hasn't been altered since it was signed.
   * **Authenticity and Integrity**: If the signature is valid, it confirms the identity of the signer and ensures that the content of the message or document has not been tampered with since it was signed.
3. **Non-Repudiation**:
   * A digital signature also provides non-repudiation. This means the signer cannot later deny having signed the document, as only their private key could have created that unique signature.
4. **Security Considerations**:
   * Digital signatures are secure, but their security depends on the secrecy of the private key. If a private key is compromised, the security of the digital signature is compromised too.
   * The strength of the digital signature also depends on the cryptographic algorithm used and the length of the keys.
5. **Applications**:
   * Digital signatures are widely used in various applications, including secure email, software distribution, financial transactions, and legal documents.
   * They are legally binding in many jurisdictions, much like traditional handwritten signatures.

In summary, digital signatures provide a secure and verifiable way of confirming the authenticity and integrity of digital information. They play a crucial role in modern digital communications, ensuring secure transactions, authentic documents, and trustworthy software distribution.

Top of Form

1. **Digital Signatures and Data Integrity**:
   * **Signing Data**: When a sender wants to ensure the integrity of their data, they use their private key to create a digital signature. This signature is unique to both the data and the private key.
   * **Sending Data and Signature**: The sender then transmits the data along with the digital signature.
2. **Verification Process**:
   * **Using the Public Key**: Upon receiving the data and the signature, the recipient uses the sender's public key to verify the signature.
   * **Integrity Check**: The verification process involves a cryptographic algorithm that checks if the received signature matches the data. If the signature is valid, it proves that the data was indeed signed by the owner of the private key and that it has not been altered since it was signed.
3. **Trust in the Public Key**:
   * **Essential for Verification**: The entire process hinges on the recipient trusting that the public key used for verification genuinely belongs to the sender. If there's doubt about the ownership of the public key, the integrity of the data cannot be assured.
   * **Implications of Mistrust**: If the public key is not trusted, there's no guarantee that the data hasn't been intercepted and altered by a third party, with a new digital signature applied using a different private key. In such a case, the recipient might wrongly believe the data is authentic and unaltered.
4. **Why Trust Matters**:
   * **Authenticity**: Trust in the public key also implies trust in the authenticity of the sender. A verified signature not only assures that the data is intact but also that it came from the legitimate sender.
   * **Security Protocols**: In many security protocols, establishing the trustworthiness of public keys is crucial. This is often managed through certificate authorities in Public Key Infrastructures (PKI), where certificates are used to validate the ownership of public keys.

Public Key Infrastructure (PKI) is a framework used to create, manage, distribute, use, store, and revoke digital certificates and manage public-key encryption. It's a system that allows for secure electronic transfer of information over various types of networks. Here's a more detailed explanation:

1. **Key Components of PKI**:
   * **Digital Certificates**: These are electronic documents that use a digital signature to bind a public key with an identity — this could be a person, organization, or device. The certificate includes information about the key, identity, and the digital signature of an entity that has verified the key's authenticity.
   * **Certificate Authority (CA)**: This is a trusted entity that issues and manages digital certificates. CA verifies the identity of the certificate requestor (whether a person, organization, or device) and then issues a certificate to ensure their authenticity.
   * **Registration Authority (RA)**: Sometimes, before a CA issues a certificate, a RA assists in the identity verification process. The RA is responsible for accepting requests for digital certificates and authenticating the entity making the request.
   * **Public and Private Keys**: PKI employs asymmetric cryptography which involves two keys: a public key (known to others) and a private key (kept secret by the owner). The public key is embedded in the certificate and can be freely distributed, while the private key is used to decrypt messages or sign digital signatures.
2. **Functions of PKI**:
   * **Authentication**: PKI verifies the identity of entities and ensures that the public key contained in the certificate belongs to the entity it claims to be.
   * **Confidentiality**: By using a recipient's public key to encrypt data, PKI ensures that only the recipient can decrypt and read it with their private key.
   * **Data Integrity**: PKI ensures that data has not been altered or tampered with in transit through digital signatures.
   * **Non-Repudiation**: Digital signatures also provide non-repudiation, which means a sender cannot deny sending a message or signing a document.
3. **How PKI Works**:
   * When a user or system needs to securely communicate with another, they request the recipient's digital certificate.
   * This certificate is verified against the CA to ensure its validity.
   * Data is then encrypted with the recipient's public key, ensuring that only the recipient can decrypt it using their private key.
   * Digital signatures are used to ensure the authenticity and integrity of the message.
4. **Certificate Lifecycle Management**:
   * Certificates have a validity period and need to be renewed or revoked if circumstances change (e.g., a user leaves an organization, or a key is compromised).
   * PKI manages the entire lifecycle of a certificate, including renewal and revocation.
5. **Applications of PKI**:
   * PKI is widely used in various applications such as securing web communications (HTTPS), securing email (S/MIME), authenticating users to systems, digital signing of documents, and more.

In summary, PKI is essential for establishing secure, trustworthy digital communications and transactions. It provides a framework for managing digital identities and securing data through the use of digital certificates and cryptographic key pairs.

Question- Which of the following is true of hierarchical public key infrastructures? Select one or more: a) All CAs are ordered into a single hierarchy. b) The root certificate authority must be trustworthy for a certificate hierarchy to function. c) Root certificate authorities are governmental organisations d) It is the prime strategy of the PGP system

To determine which of these statements about hierarchical public key infrastructures (PKI) is true, let's examine each one:

a) **All CAs are ordered into a single hierarchy.**

* This statement is False. In a hierarchical PKI model, Certificate Authorities (CAs) are indeed organized in a hierarchy, but it doesn't necessarily mean there is only a single, universal hierarchy. Different organizations or networks can have their own hierarchical PKI systems, with a top-level CA and several subordinate CAs.

b) **The root certificate authority must be trustworthy for a certificate hierarchy to function.**

* This statement is true. The trustworthiness of the entire PKI system hinges on the trust placed in the root CA. If the root CA is compromised or not trustworthy, the integrity of all certificates issued within its hierarchy can be questioned.

c) **Root certificate authorities are governmental organizations.**

* This statement is false. Root CAs can be operated by various entities, including private companies, non-profit organizations, or governments. The nature of the organization running a root CA can vary widely, and many root CAs are not affiliated with governmental entities.

d) **It is the prime strategy of the PGP system.**

* This statement is false. PGP (Pretty Good Privacy) uses a different model known as the "Web of Trust," which is not hierarchical. In the Web of Trust, individual users can sign each other's public keys, establishing trust in a decentralized manner, as opposed to the top-down approach of a hierarchical PKI.

So, from the options given, (b) "The root certificate authority must be trustworthy for a certificate hierarchy to function" is true, and the others have inaccuracies or are based on different systems than hierarchical PKI.Top of Form

So, we have the idea of certificates.

Certificates are our means of getting other people to be able to trust our public key.

Once they can do that, we can start the whole ball rolling with a whole public key infrastructure

thing.

Everybody can happily communicate with everybody else securely.

But if I want to instill trust in my public key, if I want somebody across the world like

Zoe to be able to trust my public key, who should I get to sign my public key?

Well, I guess the best thing is to have somebody who trusts me, or even if they don't trust

me, somebody who can sit close to me while I generate my key pair, who sees how I take

my private key and put it away, and hands over my public key physically to them.

At the same time, I might present them with my passport so they really can check that

it's me, that it looks like me, or something like that.

Or I have to go to somebody who's known me all their lives and really does trust me.

So, all of a sudden, we've got a weird situation with trust going on in order to get cryptography

to work.

So, I would go to one of my friends.

I guess that would be Betty.

And Betty does the signing.

How far does that get us?

Well, hardly very far at all, in case Betty is not a friend of Zoe's.

We might hope that Betty got her public key signed by Carl, and that Carl, in his turn,

got his signed by Diane, and that Diane...

And if we're lucky, all of those people might at some point lead to somebody who has been

signed by Zoe already.

Yngve, I guess.

So, it's quite conceivable that we can make this work if we can find chains of signatures

on certificates that lead us from one individual who needs to trust another individual, and

vice versa.

Indeed, this is the basic scheme that was introduced in something that we call

Pretty Good Privacy.

I say privacy because I'm English.

You'll hear a lot of international English-speaking people say privacy.

Pretty good privacy, or pretty good privacy.

Yeah.

This is now getting to be a very well-established system for encrypting things, but not only

encrypting things, for establishing the infrastructure that is needed in order to trust keys.

I'm calling it an infrastructure.

And now you understand why we get names like PKI, because this stands for the idea of public

key infrastructures.

Ways and means of putting all of these things together in order to allow trust in different

people's public keys.

So it can be done more or less like this.

In practice, of course, maybe I wouldn't just get Betty to sign my key.

I could go to all of my friends, all the people who trust me, and vice versa.

We can all get together.

There have been things called key signing parties, where these crypto nerds got together

early on.

And they got together and shook hands with each other and said, Hello Betty, meet my

friend Carl.

I've known him for...

And in that way, you can get to learn different people through different people's contacts.

And the more people that sign your public key, in a way, allows you to build more and

more trust in it.

And there were systems based around the idea that the more people who have signed it, the

more a key could be trusted.

Which is a nice idea.

OK?

Otherwise, I'd like to remind you, or introduce to you, the concept of six degrees of separation.

I don't know how well-known it is, to be honest, but I know there has been a movie called

Six Degrees of Separation, with Will Smith starring.

The premise behind six degrees of separation is that, according to statistics on populations

and things, there are only ever six steps of acquaintance between one individual in

the world and any other individual in the world.

That's the hypothesis.

You can take some tribe out in Borneo that's hardly ever met anybody, is the idea, but

somebody will have met them, and then somebody who's met them will have met somebody else,

and that somebody else will have met somebody.

And once you've gone to six degrees of separation, you've expanded the combinations until you

actually get everybody in the world.

So if that's a principle that we can count on, then in theory at least, we might be able

to find chains of links between A and Z that are only six steps long.

And suddenly it sounds like it might be workable.

If you can find the chains.

There's a game you can play to demonstrate this, if you're into it.

When you sit next to somebody on a long flight that you've never met before, if you've really

got nothing else to do, you can strike up a conversation with this person and see if

you can find if there's one common friend between you through discussing.

I guess that goes like, well, I used to go to school at this place, and then I went to

university here, and my family has been very involved in sailing.

And so you find all the possible circles of connections and see if the person next to

you can connect to some common acquaintance.

As proof of concept, I guess.

But yeah, that is the basic principle behind the ideas, the original ideas of how PGP would

work is through these links of certificates and signings to establish trust.

In the next video, I'm going to show you another possibly more common way of establishing trust

between certificates that you might recognise more readily.