CSE 410 Fall 2025
Privacy-Enhancing Technologies

Marina Blanton

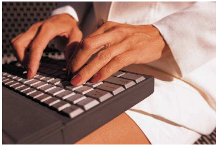
Department of Computer Science and Engineering
University at Buffalo

Lecture 8: Protecting Data in Transit II

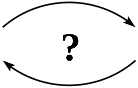
Proteching Data in Transit

- We are discussing Diffie-Hellman key exchange
- Implementing authenticated key exchange requires additional tools
  - signatures
  - certificates
- This will allow us to achieve key establishment with strong security guarantees

Public Keys and Trust



Alice
public key pk_A
secret key sk_A



Bob
public key pk_B
secret key sk_B

- If we want to use public-key cryptography, we are facing the key distribution problem
  - how/where are public keys stored?
  - how do I obtain someone's public key?
  - how can Bob know or "trust" that pk_A is indeed Alice's public key?

Public-Key Certificates

Distribution of public keys can be done

- by public announcement
  - a user distributes her key to recipients or broadcasts to community
- through a publicly available directory
  - can obtain greater security by registering keys with a public directory

Neither approach protects against forgeries

Digital certificates are used to address this problem

- a certificate binds identity (and/or other information) to a public key

Public-Key Certificates

- Assume there is a central authority *CA* with a known public key pk_{CA}
- CA produces certificate for Bob as $\mathsf{cert}_B = \mathsf{sig}_{CA}(\mathsf{pk}_B||\text{Bob})$
- Bob distributes $(\mathsf{pk}_B, \mathsf{cert}_B)$
- Alice can verify that her copy of Bob's key is genuine
- This technique is used in many applications
  - TLS/SSL, ssh, email, IPsec, etc.

Public-Key Certificates

Fundamentally, a certificate is a binding between a public key and identity

In practice, it contains other fields

- e.g., the type of signing algorithm, expiration date, etc.

The underlying assumption is that the user has an authentic copy of the CA's public key

Alternatively, there can be a sequence of certificates (chain of trust) from a key the user trusts to the key being verified

Key Types

We can distinguish between two fundamental types of keys:

- long-term keys
  - they are set up in advance and stored securely
  - they could be private keys corresponding to the user's public keys
  - they could be secret keys shares with another party such as the TA
- short-lived session keys
  - a session key is used for a particular session and is discarded at the end of it
  - session keys are normally secret keys for a symmetric cryptography

Key Exchange Security

Recall that we want the parties to authenticate during a key agreement protocol

- this is called an authenticated key exchange

Other considerations are also important in real life applications

- suppose that a session key is exposed
  - we prefer to see no impact on the security of the long-lived key
- suppose that an attacker gets a hold of your long-lived key
  - ideally this should not compromise the security of past session keys
  - this property is called perfect forward secrecy

Diffie-Hellman Key Exchange

Authenticated Diffie-Hellman key exchange

- each user U has a private signing key sk_U and the corresponding public verification key pk_U
- there is a trusted authority TA that signs keys
- user U holds a certificate $\mathsf{cert}(U)$ issued by the TA

$$\mathsf{cert}(U) = (U, pk_U, \sigma_{TA}(U, pk_U))$$

- the protocol is also known as station-to-station key agreement
- it combines the key exchange with a mutual authentication scheme

Authenticated Diffie-Hellman Key Exchange (simplified)

- public parameters are as before (G, q, g)
- Alice chooses random a, computes $x_A = g^a$, and sends $\mathsf{cert}(A)$ and x_A to Bob
- Bob chooses random b, computes

$$x_B = g^b,\ k = (x_A)^b = g^{ab} \text{ , and } \ y_B = \sigma_B(A||x_B||x_A)$$

  and sends $\mathsf{cert}(B)$, x_B, and y_B to Alice
- Alice verifies y_B; if the signature is valid, she computes

$$k = (x_B)^a = g^{ab} \text{ and } \ y_A = \sigma_A(B||x_A||x_B)$$

  and sends y_A to Bob
- Bob verifies y_A; if the signature is valid, he accepts

Diffie-Hellman Key Exchange

Security of authenticated Diffie-Hellman

- the man-in-the-middle attack on DH key exchange no longer works
- what happens now is:

Alice		Mallory		Bob
	g^a →		$g^{a'}$ →	
	← $g^{b'}, \sigma_B(A\|\|g^{b'}\|\|g^a)$		← $g^b, \sigma_B(A\|\|g^b\|\|g^{a'})$	
	$\sigma_A(B\|\|g^a\|\|g^{b'})$ →		$\sigma_A(B\|\|g^{a'}\|\|g^b)$ →	

- Mallory cannot forge Alice's and Bob's signature, so she cannot be successful

Diffie-Hellman Key Exchange

Security of authenticated Diffie-Hellman

- this protocol is a secure mutual identification scheme
  - this can be proven using the security definitions for mutual authentication
- if an adversary is active, this will be detected by the participants
- if the adversary is passive, both parties will accept with the same key
  - the adversary cannot compute any information about the key under appropriate hardness assumptions

Diffie-Hellman Key Exchange

- We might want to consider possible influence that different sessions can have on each other in real life usage
- We'll next look at security under a known session key attack
  - Mallory observes several sessions with different users (which can involve Mallory as well) of her choice
  - Mallory is able to compromise session keys associated with some of the observed sessions of her choice
  - Mallory is then asked to recover the key for a challenge session

Diffie-Hellman Key Exchange

Known session key attack on authenticated Diffie-Hellman

- this key exchange protocol is secure against the known session key attack
- intuition:
  - the values g^a, g^b are chosen anew for each session
  - they are not related to previous sessions or the long-term keys of the participants
  - it is computationally infeasible, given g^a and g^b, to compute any information about g^{ab}

Diffie-Hellman Key Exchange

Perfect forward secrecy

- this property means that compromise of long-term key does not compromise past session keys
- suppose Mallory records sessions between Alice and Bob and somehow gets a hold of Alice's secret signing key
- this property requires that Mallory cannot recover session keys for Alice's expired session
  - an expired session is a session for which Alice erased all information used to generate the session key k
- where does authenticated Diffie-Hellman stand?

Diffie-Hellman Key Exchange

Perfect forward secrecy

- in authenticated Diffie-Hellman protocol, session keys are independent of long-term keys
- it achieves perfect forward secrecy

Diffie-Hellman Key Exchange

Perfect forward secrecy

- in authenticated Diffie-Hellman protocol, session keys are independent of long-term keys
- it achieves perfect forward secrecy

We arrive at the following conclusion:

- authenticated Diffie-Hellman key agreement is
  - an authenticated key agreement scheme
  - secure against known session key attacks and
  - achieving perfect forward secrecy
- now this is a standard security requirement for key exchange protocols

Key Exchange Security

Contrast the above with the following scenario

- suppose we use public-key encryption
- Bob has a public-private key pair $(\mathsf{pk}_B, \mathsf{sk}_B)$ and makes pk_B available to others
- Alice chooses a symmetric key k and sends it encrypted to Bob: $\mathsf{Enc}_{\mathsf{pk}_B}(k)$
- this was used in practice, e.g., in TLS

Can we achieve the same security properties?

Key Exchange Security

Contrast the above with the following scenario

- suppose we use public-key encryption
- Bob has a public-private key pair $(\mathsf{pk}_B, \mathsf{sk}_B)$ and makes pk_B available to others
- Alice chooses a symmetric key k and sends it encrypted to Bob: $\mathsf{Enc}_{\mathsf{pk}_B}(k)$
- this was used in practice, e.g., in TLS

Can we achieve the same security properties?

- authentication is possible
- what about perfect forward secrecy?

Deniability

Another property that can be important for secure communication is that for a conversation to be deniable

- when we converse in person, we don't expect what we say to hold as evidence against us in court
- we want similar guarantees for digital communication
- this is at odds with the need to authenticate to establish a secure communication channel
- signatures specifically are traced back to you and provide evidence of your participation

Deniability

To achieve deniability, we want to use a mix of public-key and symmetric tools to guarantee that

- authentication is still achieved during key establishment
- a conversation partner is unable to provably attribute messages to you afterwards

Asymmetric (public-key) tools are important for authentication

Symmetric tools (e.g., MACs) are important for proving knowledge of certain information and don't point to the message origin

Summary

- Key distribution is divided into
  - key predistribution
  - session key distribution
  - key agreement, which we discussed
- There are many key exchange protocols, many of which are based on the Diffie-Hellman key exchange
- The properties that are essential
  - secure mutual authentication
  - secure key computation
  - resilience to known session key attack
  - perfect forward secrecy
- Deniability can be important as well