

# Web Service Security



# WIRE-LEVEL SECURITY

---

- Security begins at the transport or wire level; that is, with basic protocols that govern communications between a web service, whether SOAP-based or ReST-style, and its clients.
- Security at this level typically provides three services.
  - First, the client and service need transport-level assurance that each is communicating with the other rather than with some impostor.
  - Second, the data sent from one side to the other needs to be encrypted strongly enough so that an interceptor cannot decrypt the data and thus gain access to the secrets carried therein.
  - Third, each side needs assurance that the received message is the same as the sent message. This chapter covers the basics of wire-level security with code examples.

# USER AUTHENTICATION AND AUTHORIZATION

---

- Web services provide clients with access to resources.
- If a resource is secured, then a client needs the appropriate credentials to gain access.
- The credentials are presented and verified through a process that usually has two phases.
  - In the first phase (known as user authentication), a client (user) presents information such as a username together with a credential such as a password. If the credential is not accepted, access to the requested resource is denied.
  - The second phase (known as role authorization), which is optional, consists of fine-tuning the authenticated user's access rights.
    - For example, a stock-picking web service might provide all paying customers with a username and password, but the service might divide the customers into categories, for instance, regular and premier. Access to certain resources might be restricted to premier clients.

# WS - SECURITY

---

- WS-Security, or WSS for short, is a collection of protocols that specify how different levels of security can be enforced on messaging in SOAP-based web services.
  - For example, WSS specifies how digital signatures and encryption information can be inserted into SOAP headers.
  - SOAP-based services are designed to be transport-neutral. Accordingly, WSS is meant to provide comprehensive end-to-end security regardless of the underlying transport.

# SECURITY BASICS

---

- Message digest



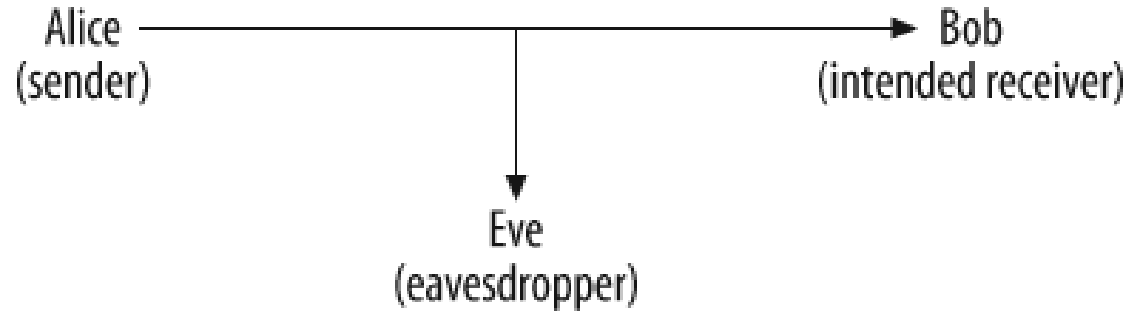
*HMAC*

*(Hash Message Authentication Code)*

# HTTPS BASICS

---

- HTTPS is the most popular among the secure versions of HTTP.
- HTTPS provides three critical security services over and above the transport services that HTTP provides.



# HTTPS BASICS

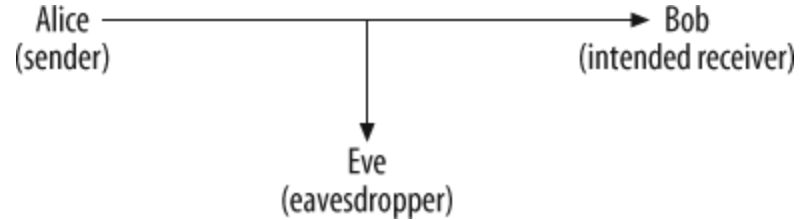
---

- Alice needs to send a secret message to Bob.
- Eve, however, may be eavesdropping.
- Eve may even try to dupe Alice and Bob into believing that they are communicating with one another when, in fact, each is communicating instead with Eve.
  - This variation is known as the *MITM* (Man In The Middle) attack.

# HTTPS BASICS

---

- For secure communications, Alice and Bob thus need these three services:
  - Peer authentication
  - Confidentiality
  - Integrity



# PEER AUTHENTICATION

---

- Alice needs Bob to authenticate himself so that she is sure about who is on the receiving end before she sends the secret message.
- Bob, too, needs Alice to authenticate herself so that he knows that the secret message is from her rather than an impostor such as Eve.
- This step also is described as mutual authentication or mutual challenge.

# CONFIDENTIALITY

---

- Once Alice and Bob have authenticated each other, Alice needs to encrypt the secret message in such a way that only Bob can decrypt it.
- Even if Eve intercepts the encrypted message, she should not be able to decrypt the message because doing so requires enormous computational power or incredibly good luck.

# INTEGRITY

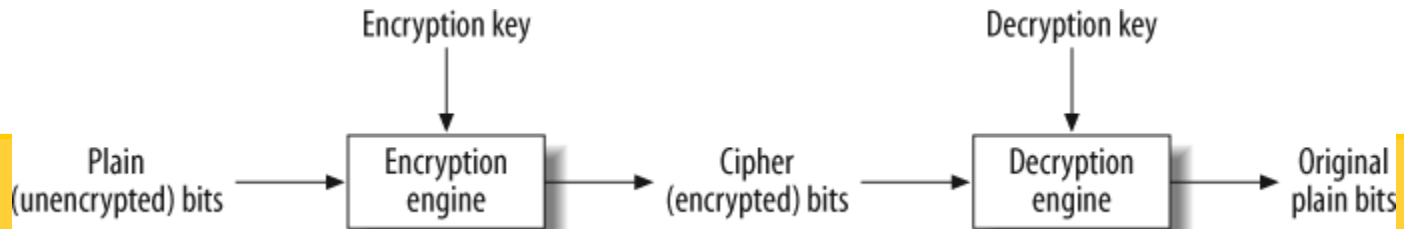
---

- The message that Alice sends should be identical to the one that Bob receives.
- If not, an error condition should be raised.
- The received message might differ from the sent one for various reasons; for instance, noise in the communications channel or deliberate tampering by Eve.
- Any difference between the sent and the received message should be detected.

# SYMMETRIC AND ASYMMETRIC ENCRYPTION/DECRYPTION

---

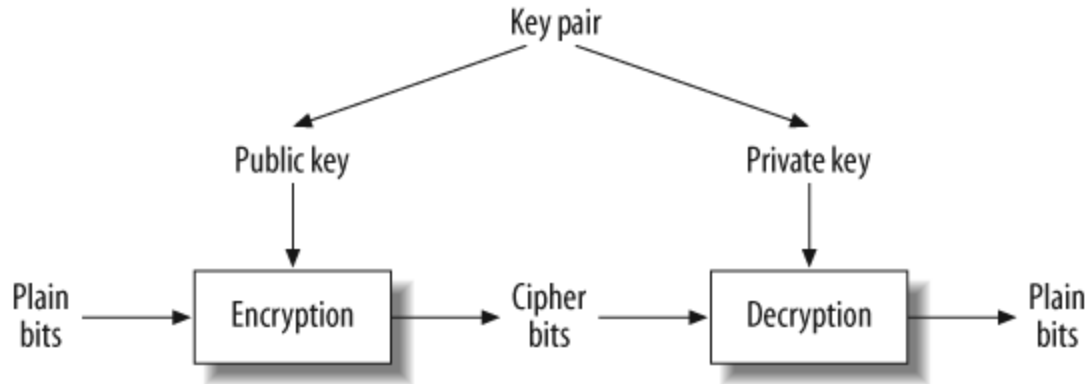
- Under either approach, the bits to be encrypted (plain bits) are one input to an encryption engine and an encryption key is the other input.
- The encrypted bits are the cipher bits.
- In the symmetric approach, the same key (a.k.a. the secret key or the single key) is used to encrypt and decrypt.



- 
- The symmetric approach has the advantage of being relatively fast, but the disadvantage of what is known as the key distribution problem.
    - How is the secret key itself to be distributed to the sender and the receiver?

# ASYMMETRIC APPROACH

---



# ASYMMETRIC APPROACH

---

- In the asymmetric approach, the starting point is a key pair, which consists of a private key and a public key.
- As the names suggest, the private key should not be distributed but safeguarded by whoever generated the key pair.
- The public key can be distributed freely and publicly.
- If message bits are encrypted with the public key, they can be decrypted only with the private key, and vice-versa.
- The asymmetric approach solves the key distribution problem, but asymmetric encryption and decryption are roughly a thousand times slower than their symmetric counterparts.

- 
- HTTPS handles peer authentication through the exchange of digital certificates.
  - In many cases, however, it is only the client that challenges the server.

- 
- For authentication and confidentiality, HTTPS relies on digital certificates, which are widely used in public key cryptography precisely because the exchange of secret keys is so difficult among many users.
  - An X.509 certificate is a public key certificate that serves as an identity certificate by binding the public key from a key pair to an identity such as a person (for instance, Alice) or an organization (for instance, Bob's employer).
  - The certificate contains the digital signature of a CA such as a VeriSign, although certificates can be self-signed for testing purposes.
  - In signing a digital certificate, a CA endorses the certificate and thereby verifies that the certificate's public key is bound to a particular identity.
    - For instance, VeriSign signs Alice's certificate and thereby verifies that the certificate's public key belongs to Alice's key pair.

---

Demo

# **CERTIFICATE INFORMATION**

