

Introduction to public-key cryptography

Part 2: applications of public-key cryptography

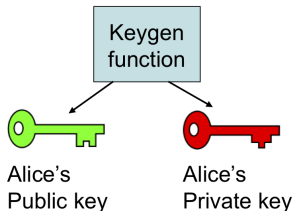
Jean-Sébastien Coron

University of Luxembourg

- Part 1: basic constructions (previous lecture)
 - History
 - Classical cryptography: block-ciphers, hash functions
 - Public-key cryptography: RSA encryption and RSA signatures, DH key exchange
- Part 2: applications of public-key cryptography (this lecture)
 - Security models
 - How to encrypt and sign securely with RSA. OAEP and PSS.
 - Public-key infrastructure. Certificates, HTTPS protocol.

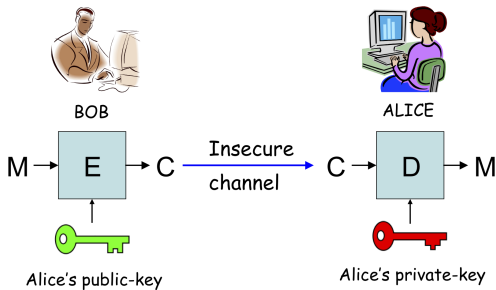
Public-key cryptography

- Invented by Diffie and Hellman in 1976. Revolutionized the field.
- Each user now has two keys
 - A public key
 - A private key
 - Should be hard to compute the private key from the public key.
- Enables:
 - Asymmetric encryption
 - Digital signatures
 - Key exchange, identification, and many other protocols.



Public-key encryption

- Public-key encryption (or asymmetric encryption)
 - Solves the key distribution issue



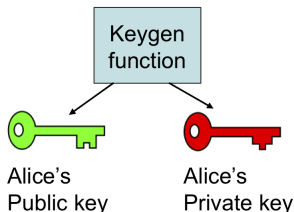
- Key generation:

- Generate two large distinct primes p and q of same bit-size $k/2$, where k is a parameter.
- Compute $n = p \cdot q$ and $\phi = (p - 1)(q - 1)$.
- Select a random integer e such that $\gcd(e, \phi) = 1$
- Compute the unique integer d such that

$$e \cdot d \equiv 1 \pmod{\phi}$$

using the extended Euclidean algorithm.

- The public key is (n, e) .
- The private key is d .



- Encryption with public-key (n, e)
 - Given a message $m \in [0, n - 1]$ and the recipient's public-key (n, e) , compute the ciphertext:

$$c = m^e \bmod n$$

- Decryption with private-key d
 - Given a ciphertext c , to recover m , compute:

$$m = c^d \bmod n$$

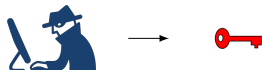
- Security is based on the hardness of factorization
 - Given $n = p \cdot q$, no known efficient algorithm to recover the primes p and q .
 - Public modulus n must be large enough: at least 1024 bits. 2048 bits is better.

Security models

- To be rigorous when speaking about security, one must specify: the attacker's goal and the attacker's power.

- The attacker's goal

- Does he need to recover the private key ?



- or only decrypt a particular ciphertext (or less) ?

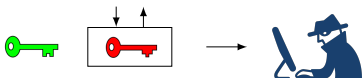


- The attacker's power

- Does he get only the user's public-key ?

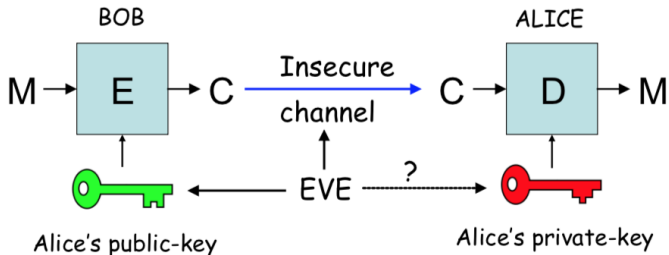


- or more ?



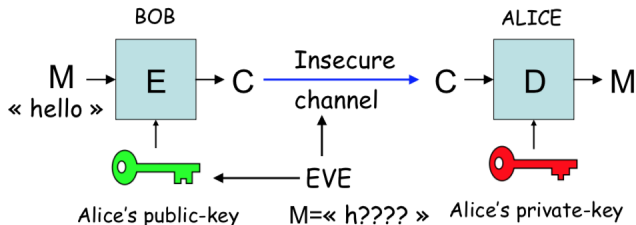
Attacker's goal

- One may think that the adversary's goal is always to recover the private key.
 - complete break
 - may be too ambitious in practice



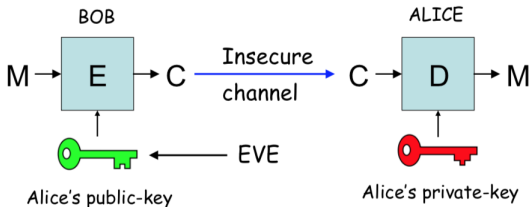
Attacker's goal

- More modest goal: being able to decrypt one ciphertext.
 - or recover some information about a plaintext (for example, the first character)



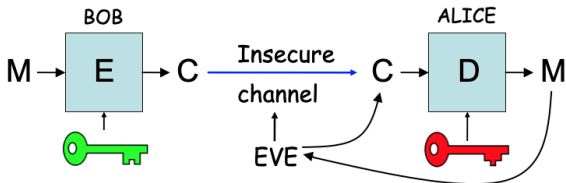
Attack model

- Specify the power of the attacker
- Public-key only attack: the attacker gets only the public-key
 - Called chosen plaintext attack (CPA) because the adversary can encrypt any plaintext of his choice.
 - Weakest adversary



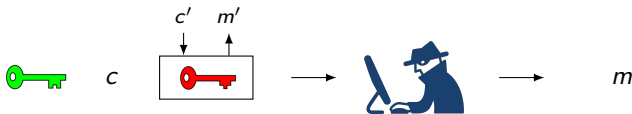
Chosen ciphertext attack (CCA)

- Most powerful attack
- The attacker can obtain decryption of messages of his choice
- May be realistic in practice
 - attacker gets access to a decryption machine
 - encryption algorithm used in a more complex protocol in which users can obtain decryption of chosen ciphertexts.



Chosen ciphertext attack against textbook RSA

- Chosen-ciphertext attack:
 - Given ciphertext c to be decrypted
 - Generate a random r
 - Ask for the decryption of the random looking ciphertext $c' = c \cdot r^e \pmod n$
 - One gets $m' = (c')^d = c^d \cdot (r^e)^d = c^d \cdot r = m \cdot r \pmod n$
 - This enables to compute $m = m'/r \pmod n$



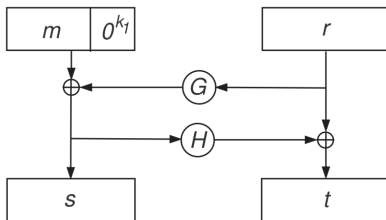
- Conclusion: do not use textbook RSA encryption !

Strongest security notion for public-key encryption

- Indistinguishability under adaptive chosen ciphertext attack (IND-CCA2)
 - Formalized in 1991 by Rackoff et Simon
 - A ciphertext should give no information about the corresponding plaintext, even under an adaptive chosen-ciphertext attack.
 - Has become standard security notion for encryption.



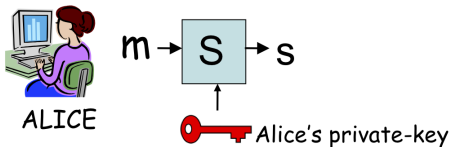
- OAEP (Bellare and Rogaway, E'94)
 - IND-CCA2, assuming that RSA is hard to invert.
 - PKCS #1 v2.1



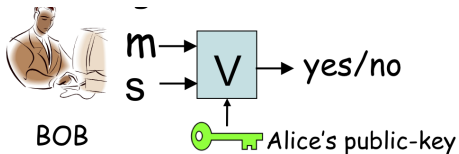
$$c = (s||t)^e \bmod N$$

Digital signatures

- A digital signature σ is a bit string that depends on the message m and the user's public-key pk
 - Only Alice can sign a message m using her private-key sk



- Anybody can verify Alice's signature of the message m given her public-key pk



The RSA signature scheme

- Key generation :
 - Public modulus: $N = p \cdot q$ where p and q are large primes.
 - Public exponent : e
 - Private exponent: d , such that $d \cdot e = 1 \pmod{\phi(N)}$
- To sign a message m , the signer computes :
 - $s = m^d \bmod N$
 - Only the signer can sign the message.
- To verify the signature, one checks that:
 - $m = s^e \bmod N$
 - Anybody can verify the signature

Attacks against textbook RSA signatures

- Given $\sigma_1 = (m_1)^d \bmod N$ and $\sigma_2 = (m_2)^d \bmod N$:
 - one can compute the signature of $m_1 \cdot m_2$ without knowing d :

$$\sigma = (m_1 \cdot m_2)^d = (m_1)^d \cdot (m_2)^d = \sigma_1 \cdot \sigma_2 \pmod{N}$$

- One cannot use plain RSA signature
 - Hash-and-sign paradigm: the message is first hashed

$$m \longrightarrow H(m) \longrightarrow 1001 \dots 0101 \parallel H(m)$$



$$\sigma = (1001 \dots 0101 \parallel H(m))^d \pmod{N}$$

Attack scenario for signature schemes

- We must specify the adversary's goal and the adversary's power.

- Adversary's goal

- Controlled forgery: the adversary can produce the signature of any message

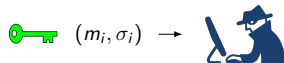


- Existential forgery: the adversary can produce the signature of a (possibly meaningless) message

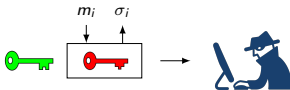


- Adversary's power

- Known message attack: the adversary obtains a set of pairs message/signature



- Chosen message attack: the adversary can obtain the signature of any message of his choice, adaptively.



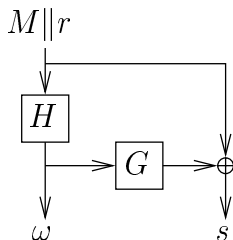
Strongest security notion for signature scheme

- Combines weakest goal with strongest adversary
- Existential unforgeability under an adaptive chosen message attack
 - Defined by Goldwasser, Micali and Rivest in 1988
 - It must be infeasible for an attacker to forge the signature of a message, even if he can obtain signatures of messages of his choice.



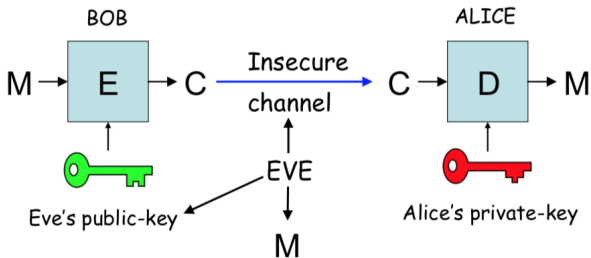
The PSS signature scheme

- PSS (Bellare and Rogaway, Eurocrypt'96)
 - IEEE P1363a and PKCS#1 v2.1.
 - 2 variants: PSS and PSS-R (message recovery)
 - Provably secure against chosen-message attacks, in the random oracle model.
 - PSS-R: $\mu(M, r) = \omega \| s$, $\sigma = \mu(M, r)^d \bmod N$



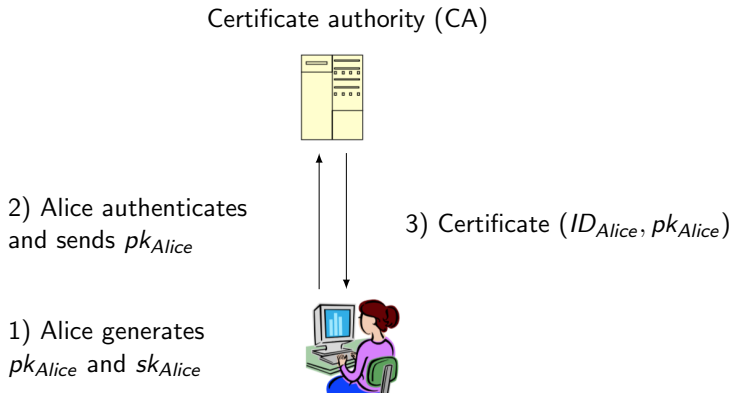
Public-key infrastructure

- Public-keys need to be authenticated
 - Bob needs to be sure that the public-key belongs to Alice.
 - Otherwise, impersonation attack



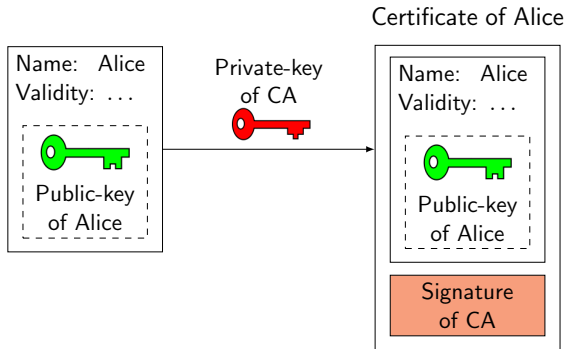
Public-key Infrastructure (PKI)

- A central authority binds public-keys to identities.
 - Public-key is stored in a certificate provided by the central authority
 - Used to prevent impersonation attack

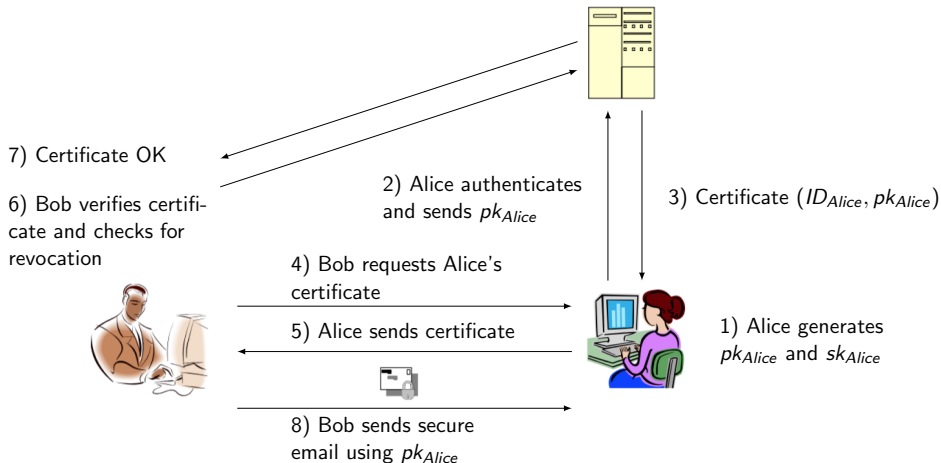


Public-key certificate

- The signature of the certificate authority (CA) binds together a public-key with an identity in the certificate.
 - Proves ownership of a public-key
 - Bob can be sure that the public-key belongs to Alice by checking the signature using the CA public-key.
 - The CA is trusted by all participants.
 - Most common certificate standard: X.509

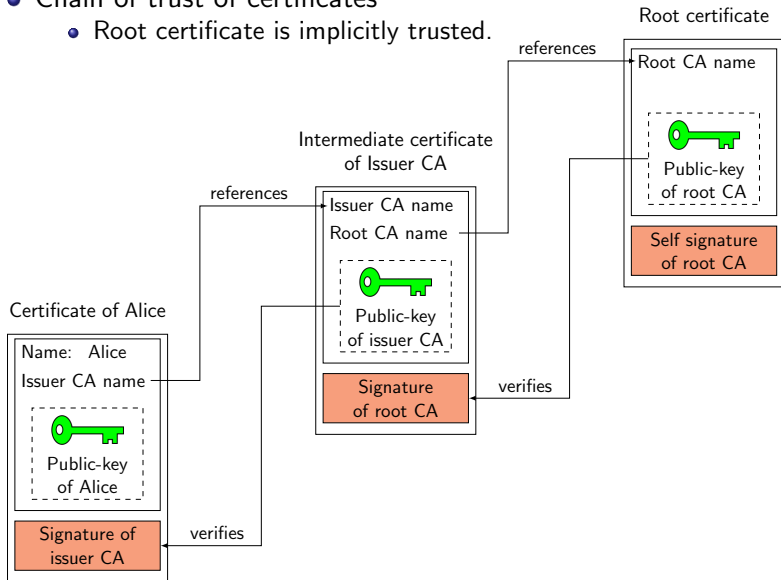


PKI encryption



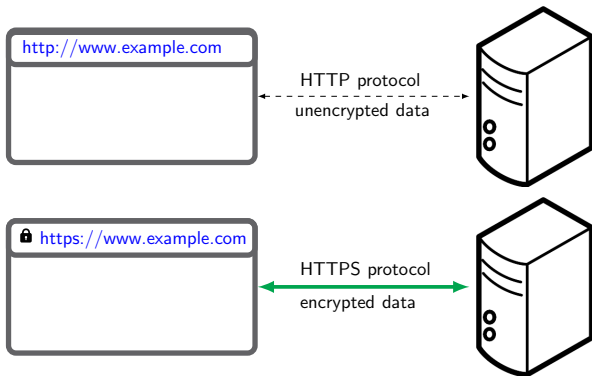
Hierarchy of certificates

- Chain of trust of certificates
 - Root certificate is implicitly trusted.



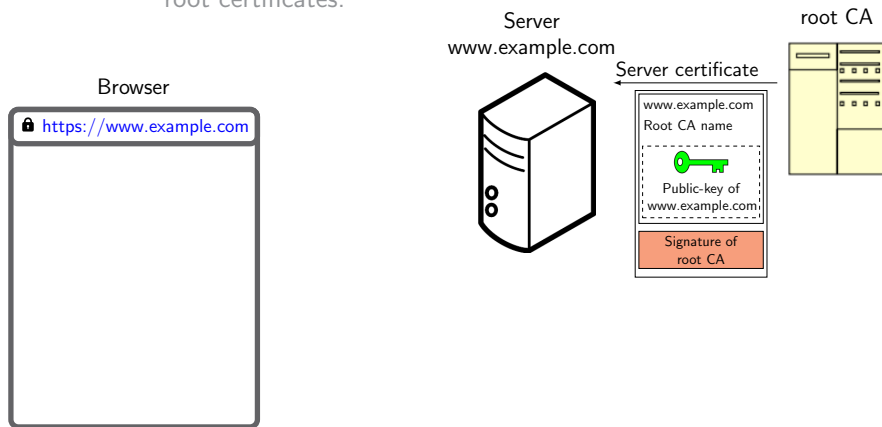
The HTTPS protocol

- Hypertext Transfer Protocol Secure (HTTPS) is a protocol for securely browsing the web.
 - Communication is encrypted using Transport Layer Security (TLS); formerly, Secure Sockets Layer (SSL).
 - Validates the authenticity of an HTTPS web server, and ensures confidentiality and integrity of communications.



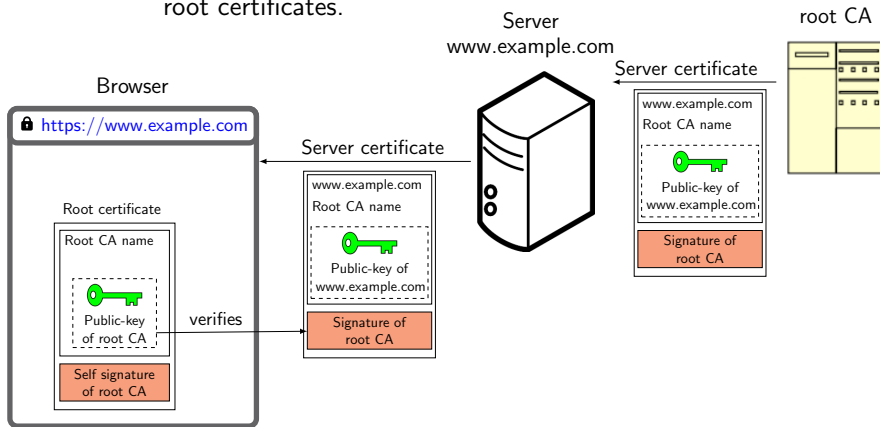
Server authentication in HTTPS

- CA issues a certificate to the server to authenticate the server's public-key
 - The server expects the CA's certificate to be contained in most clients web browser.
 - One needs to trust the browser's publisher to include correct root certificates.



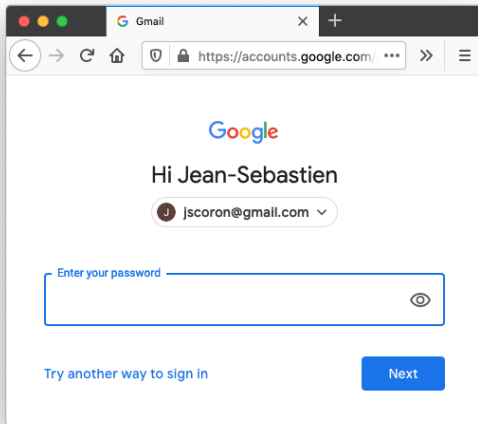
Server authentication in HTTPS

- CA issues a certificate to the server to authenticate the server's public-key
 - The server expects the CA's certificate to be contained in most clients web browser.
 - One needs to trust the browser's publisher to include correct root certificates.



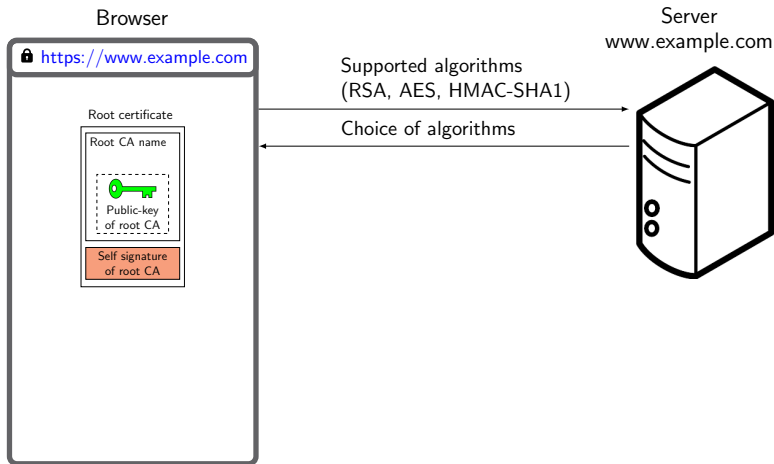
Client authentication in HTTPS

- Generally, only the server is authenticated
 - Mutual authentication requires a client certificate.
 - Most services use passwords to authenticate users, instead of client certificates.



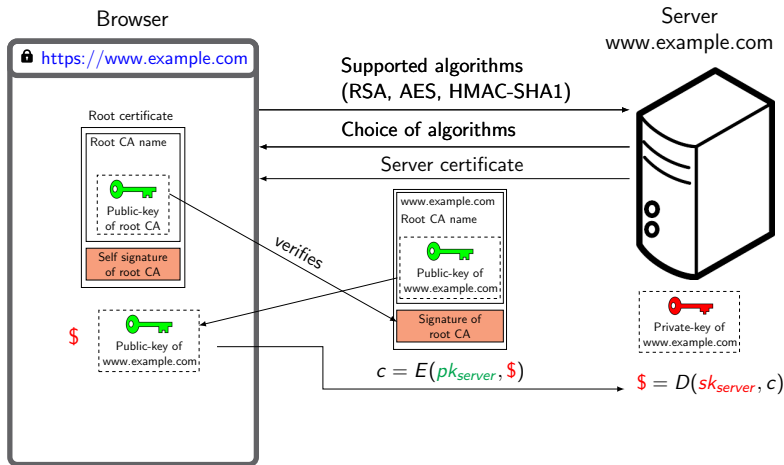
The TLS protocol

- Three steps
 - Negotiation for algorithms used.
 - Certificate verification and PK encryption for session key.
 - Symmetric encryption for traffic encryption.



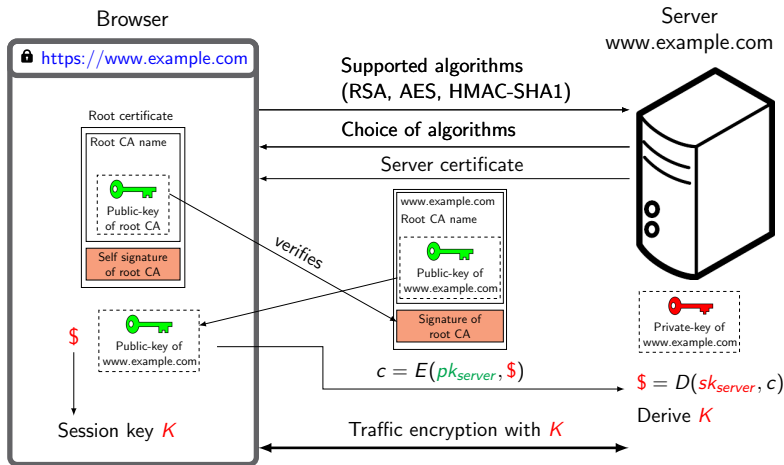
The TLS protocol

- Three steps
 - Negotiation for algorithms used.
 - Certificate verification and PK encryption for session key.
 - Symmetric encryption for traffic encryption.



The TLS protocol

- Three steps
 - Negotiation for algorithms used.
 - Certificate verification and PK encryption for session key.
 - Symmetric encryption for traffic encryption.



Credit card via HTTPS

- HTTPS only protects the credit card number during transit between the user's computer and the server
 - Does not protect against an attack on the server
- Attack on the server usually easier than interception in transit
 - Credit card number often saved in a database in merchant site
 - Attacks generally concentrate on the server and database

