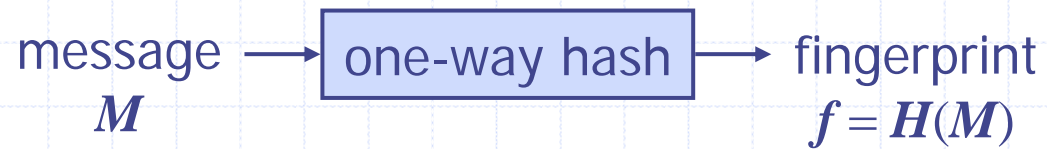


Information Security



Outline and Reading

- ◆ Digital signatures
 - Definition
 - RSA signature and verification
- ◆ One-way hash functions
 - Definition
 - Applications
- ◆ Key distribution
 - Certificates
 - Revocation

Digital Signature

- ◆ A digital signature is a string S associated with a message M and the author A of M that has the following properties
 - Integrity:** S establishes that M has not been altered
 - Nonrepudiation:** S unequivocally identifies the author A of M and proves that A did indeed sign M
- ◆ A digital signature scheme provides algorithms for
 - Signing a message by the author
 - Verifying the signature of a message by the reader
- ◆ A recently passed law in the US gives digital signatures the same validity of handwritten signatures
- ◆ A public-key cryptosystem yields a digital signature scheme provided $encrypt(K_E, decrypt(K_D, M)) = M$
 - Signature:** Alice (author) computes $S = decrypt(K_D, M)$ using her private key K_D and sends the pair (M, S) to Bob
 - Verification:** Bob (reader) computes $M' = encrypt(K_E, S)$ using Alice's public key K_E and checks that $M' = M$

RSA Digital Signature

◆ Setup:

- $n = pq$, with p and q primes
- e relatively prime to $\phi(n) = (p - 1)(q - 1)$
- d inverse of e in $\mathbf{Z}_{\phi(n)}$

◆ Keys:

- Public key: $K_E = (n, e)$
- Private key: $K_D = d$

◆ Signature:

- Message M in \mathbf{Z}_n
- Signature $S = M^d \bmod n$

◆ Verification:

- Check that $M = S^e \bmod n$

◆ Setup:

- $p = 5, q = 11$
 $n = 5 \cdot 11 = 55$
- $\phi(n) = 4 \cdot 10 = 40$
 $e = 3$
- $d = 27$ ($3 \cdot 27 = 81 = 2 \cdot 40 + 1$)

◆ Keys:

- Public key: $K_E = (55, 3)$
- Private key: $K_D = 27$

◆ Signature:

- $M = 51$
- $S = 51^{27} \bmod 55 = 6$

◆ Verification:

- $S = 6^3 \bmod 55 = 216 \bmod 55 = 51$

Cryptographic Hash Function

- ◆ A cryptographic hash function is a function H with the following properties
 - M maps a string M of arbitrary length into an integer $f = H(M)$ with a fixed number of bits, called the **fingerprint** or **digest** of M
 - H can be computed efficiently
 - Given an integer f , it is computationally infeasible to find a string M such that that $H(M) = f$ (**one-way**)
 - Given a string M , it is computationally infeasible to find another string M' such that $H(M) = H(M')$ (**collision resistance**)
 - It is computationally infeasible to find two strings M and M' such that $H(M) = H(M')$ (**strong collision resistance**)

Ideal Cryptographic Hash Function

- ◆ A **random oracle** models a cryptographic hash function
- ◆ An oracle answers all possible questions and does so consistently
 - No question goes unanswered
 - Same question, same answer
- ◆ Computing $H(q)$ consists of asking the oracle question q
- ◆ If the oracle has not previously heard q , it chooses a random value r , answers r and remembers the pair (q, r)
- ◆ Else, it remembers and replies with the previous answer

Practical Cryptographic Hash Functions

- ◆ MD5 (Message Digest 5, 1992)
 - uses a 128-bit (16 bytes) fingerprint
 - Try it: `%> echo "ipsumdolor" | md5sum`
- ◆ SHA-1 (Secure Hash Algorithm 1, 1995)
 - uses a 160-bit (20 bytes) fingerprint
 - Try it: `%> echo "ipsumdolor" | sha1sum`
- ◆ Various attacks published on MD5 and (more recently) SHA-1
- ◆ SHA-1 is considered more secure than MD5

Applications of Cryptographic Hash Functions

◆ Data integrity:

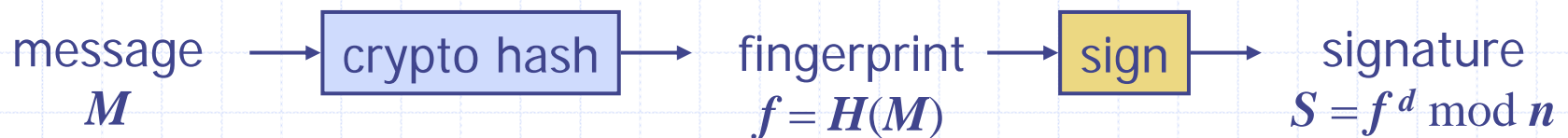
- Alice sends a file to Bob and when Bob receives it, he knows that the file was not modified in transit

◆ Commitment:

- Alice commits to a string x without revealing x , and when she reveals x , Bob knows she didn't change her choice of x

Digitally Signed Fingerprints

- ◆ In the RSA digital signature scheme with modulus n , the message to be signed must be an integer in \mathbf{Z}_n , i.e., the message should have at most $b = \log n$ bits
- ◆ To overcome the above restriction on the message length, we can use the fingerprint $f = H(M)$ of the message instead of the message itself, where H is a cryptographic hash function
 - Alice computes first $f = H(M)$ and then the signature S of f
 - Bob first computes $f = H(M)$ and then verifies S
- ◆ Since the cryptographic hash function H has the collision-resistance property, it is computationally infeasible to modify the message M while preserving the signature of the fingerprint $f = H(M)$



Coin Flipping Over the Net

- ◆ Alice and Bob want to flip a random coin by communicating over the internet
- ◆ The following protocol, based on a cryptographic hash function H , ensures the fairness of the outcome
 - Alice picks a random integer x
 - Alice commits to x
 - ◆ She computes the fingerprint $f = H(x)$ and sends f to Bob
 - Bob sends to Alice his guess of whether x is odd or even
 - Alice announces the result of the coin flip: heads if Bob has guessed correctly and tails otherwise
 - Alice sends to Bob integer x as a proof of the outcome of the flip
 - Bob verifies that $f = H(x)$
- ◆ Because of the strong-collision resistance property, it is computationally infeasible for Alice to cheat

Birthday Attack

◆ Birthday paradox

- Suppose there are 23 people in a room.
- Q: what is the probability that two of them have the same birthday?
- A: 50%

◆ Hash values as birthdays

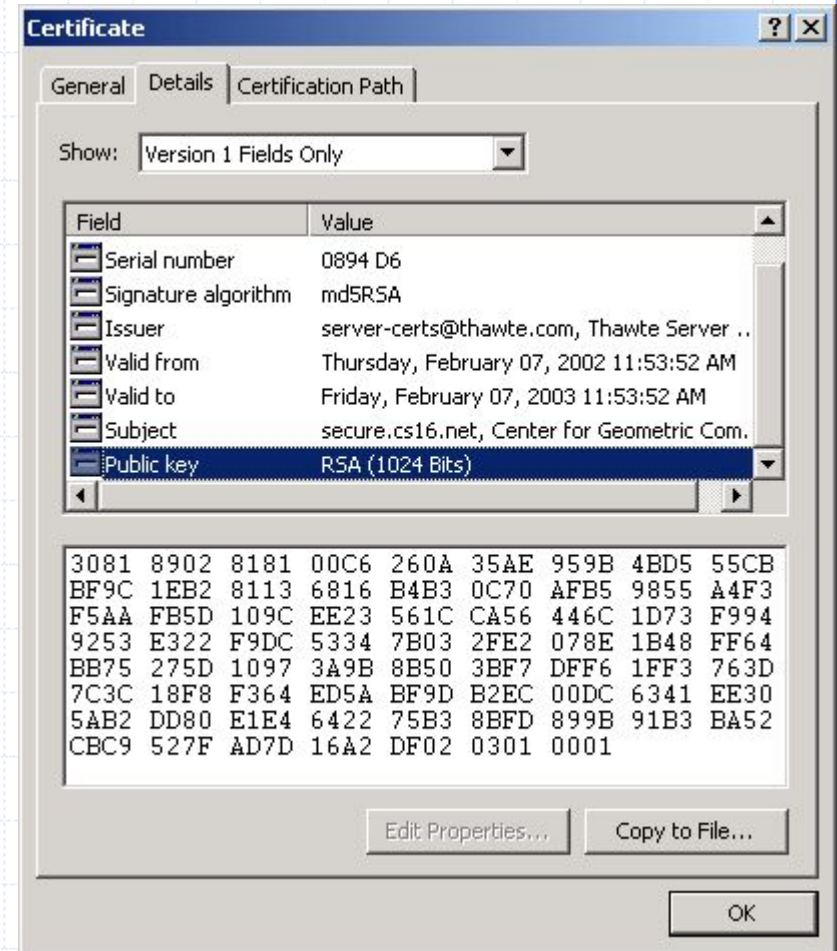
- Suppose that a hash function has m^2 possible values
- We can find a collision with 50% probability by trying $1.2m$ random pairs of strings (about the square root of the size of the domain of the hash function)

Certificates

- ◆ Public-key cryptography is based on the knowledge by each participant of the public key of the other participants
- ◆ It is complicated to securely distribute the public keys of all the participants
- ◆ A certificate is a message of the type (name, public key) signed by a third-party
- ◆ Public-key infrastructure (PKI)
 - An entity trusted by all the participants, called certification authority (CA), issues to each participant a certificate (*Name*, K_E) that authoritatively binds the participants to their public keys
 - Only the CA's public key needs to be distributed securely
 - Before sending an encrypted message to Bob or verifying a message digitally signed by Bob, Alice determines Bob's public key K_E by using Bob's certificate (Bob, K_E)

Web Server Certificates

- ◆ A Web server certificate is used to authenticate the public key of a Web server
- ◆ Fields of a Web server certificate
 - Serial number
 - Hash and signature schemes (e.g., MD5 and RSA)
 - Issuer (certification authority)
 - Period of validity (from, to)
 - Subject (URL and organization)
 - Public key
- ◆ The SSL (secure socket layer) protocol uses Web server certificates to provide encryption and authentication in a secure Web connection (https)



Certificate Revocation

- ◆ In certain circumstances, a certificate may have to be revoked before its expiration date
 - The private key of the subject has been compromised
 - The certificate was incorrectly issued by the CA
- ◆ Certificate Revocation List (CRL)
 - Time-stamped list of all the unexpired certificates that have been revoked by the CA
 - Periodically published and signed by the CA
- ◆ When presented with a certificate, one should
 - Verify the CA's signature on the certificate
 - Check that the certificate has not been revoked by searching in the latest available CRL
- ◆ By default, Web browsers do not check the revocation status of a Web server certificate, which poses a security risk