Chapter 13

by

David G. Messerschmitt

Trustworthiness

by

David G. Messerschmitt
Some objectives

- High availability
  - Expanding expectations, approaching 24x7
  - Redundancy/replication, security, human factors
- Protect confidential information
- Limit services to legitimate users or customers
- Conduct secure commercial transactions

Availability

- Application up and running correctly
- Some types of downtime:
  - Off-line upgrade and maintenance
  - Software crashes
  - Equipment failure
  - Successful denial-of-service attack
Availability costs!

- On-line upgrade and maintenance
- More application testing, more rapid bug reports and fixes
- Equipment or application redundancy
- Data replication
- Operational vigilance

Question

- What availability would you like to see in:
  - Consumer stock trading system?
  - Currency trading system?
  - Train control system?
  - Bank ATM?
  - Social application like email?
  - Telephone system?
Different security environments

- **Intranet and extranet**
  - All users may be trusted

- **Organization-to-organization**
  - Users in other organizations are less trusted, have less access

- **Citizenry**
  - Determined adversaries must be assumed

Access control

- First line of defense is to limit information and services to authorized users

- Requires:
  - Authorization policies
  - Databases with authorizations
  - Confidentiality of information and communication
  - Authentication of users who do gain access
Non-repudiation

• The second line of defense is to maintain a provable audit of commitments
  – Requires non-repudiation: neither sender nor recipient can deny message
  – Non-repudiation requires message integrity

Core technology

• Encryption
  – Depends on the existence of hard (not impossible) problems that are thought to be uncomputable by the fastest computers in reasonable time
  – “Size” of problem can be adjusted to future and anticipated computing technology
  – Symmetric and asymmetric versions
Virus

Normal executable

Infected executable

Sequence of program instructions

Original program

Replication and payload

Understanding Networked Applications 11

A First Course

Alice writes message on paper in permanent ink

Alice adds her signature

Alice seals message in envelope

Only Bob breaks seal and opens envelope

Bob verifies Alice’s signature

Post office physical security

Alice requests return acknowledgement from Bob

Understanding Networked Applications 12

A First Course
Encryption

- Transform “plaintext” data to “ciphertext” data in a way that
  - plaintext cannot be recovered without knowledge of a key
  - at least not without extraordinary computing resources
Symmetric case

Asymmetric case

Put plaintext in lockbox
Close and lock using sender's locking key
Recipient opens using unlocking key
Transport to recipient
Locking and unlocking keys are different

Symmetric case

Asymmetric case

Alice (sender) Bob (recipient)

Symmetric
Secret key Secret key
SK SK
Plaintext Ciphertext Original plaintext

Public key Secret key
Encryption Decryption

Confidentiality protocol

Understanding Networked Applications 15 A First Course
Confidentiality

Alice (sender) ➔ Bob (recipient)

Bob must possess a secret not available to anyone else

Confidentiality (con’t)

Alice (sender) ➔ Bob (recipient)

Alice must be able to transform the message so that only the person possessing that secret can read it

Bob must possess a secret not available to anyone else
Confidentiality (con’t)

Alice (sender) must be able to transform the message so that only the person possessing that secret can read it. Bob (recipient) must possess a secret not available to anyone else.

 Encrypt
  Decrypt

Authentication

Goal:
• Before Bob can trust a message received from Alice, he needs to verify that Alice is who she claims she is.
• Alice may want to verify Bob's identity before sending him a message.
Authentication

Alice (sender)

Bob (recipient)

Alice must possess a secret not available to anyone else (alternatively, physical characteristics like a fingerprint might be used)

Authentication (con’t)

Alice (sender)

Bob (recipient)

Alice must possess a secret not available to anyone else (alternatively, physical characteristics might be used)

Bob must be able to verify that Alice possesses that secret without Alice revealing it on the network or to Bob
Authentication (con’t)

Alice (sender)

Bob (recipient)

Alice must possess a secret not available to anyone else (alternatively, physical characteristics might be used)

Bob must be able to verify that Alice possesses that secret without Alice revealing it on the network (and possibly not to Bob)

Challenge-response protocol

Bob | Alice (being authenticated)
--- | ---
Public key APK | Secret key ASK

Encryption

Decryption

k

Challenge

l

Compare

Public key APK

Decryption

Encryption

k+l

Response

Understanding Networked Applications 23 A First Course

Understanding Networked Applications 24 A First Course
Question

- How does Bob obtain Alice’s public key?
- How does Bob authenticate that public key?
- Answer: Key must come from a trusted authority

Authentication (con’t)

Alice (sender)
Bob (recipient)

Alice must possess a secret not available to anyone else
(alternatively, physical characteristics might be used)

Bob also must have confirmation from a trusted authority of Alice’s public key

Digital certificate
Non-repudiation

Alice (sender)  Bob (recipient)

Alice can transform the message using a secret only she possesses

Bob must be able to verify the signature using public information confirmed by a trusted authority

Digital certificate

Understanding Networked Applications 27 A First Course

Understanding Networked Applications 28 A First Course
Digital signature

(Alice) sender

Secret key

ASK

Encryption

S

Signature

Decryption

Bob (recipient)

Public key

APK

P

 Plaintext

Compare

Summary

• A message can be sent from Alice to Bob, such that:
  – It is confidential
  – Alice’s identity is authenticated
  – Provably the message was not modified after Alice generated it, and she cannot repudiate it

• All this requires a system for distribution and certification of secrets
Distribution of secrets

- Users choose their own secrets and inform sites (password)
- In a closed administrative environment, secrets can be distributed by administrative fiat
  - Authentication servers avoid the “n^2 secret problem”
- For the citizenry, infrastructure required

Digital certificate protocol

1. Alice convinces CA of her identity
2. CA gives digital certificate and secret key to Alice
3. Alice provides Bob with a replica of her digital certificate, which provides and certifies Alice’s public key
4. Bob verifies CA signature using CA public key
### Chain of trust

<table>
<thead>
<tr>
<th>Certificate authority</th>
<th>Bank’s certificate issued by CA</th>
<th>Merchant’s certificate issued by bank</th>
</tr>
</thead>
</table>

- Authority’s known public key
- Bank’s public key
- Merchant’s public key

Verify signature

---

### Consumer electronic commerce

- **CA**
- **Customer (client)**
- **Seller (server)**

**Trust**ing CA public key, client can obtain authenticated public key of a seller
Client can generate a random, secret “session key” and send confidentially to server.

Client can authenticate server using challenge response protocol.

Client and server can communicate confidentially.
This is what “secure socket layer (SSL)” provides today.

What is missing?

Certificate infrastructure

• Certificate authorities
• Individual and corporate certificates
• Benefits:
  – Authentication of sellers and buyers
    • Avoid sales to minors etc.
  – Non-repudiation of transactions
Privacy concerns

- On-line transactions can be tracked
- Traditional opposition to “identity card” for this reason
- Safeguards are possible
  - Example: Secure Electronic Transactions (SET)
Encryption obscures data representation

Information

Data

Encryption

Ciphertext

Interpretation assumed by encryption algorithm

Assembly

Fragmentation

Block: plaintext

Representation by data (defined by application)

Information

Data

Decrypt

Decryption

Ciphertext

Block: plaintext

Understanding Networked Applications 41

A First Course
Block substitution table

<table>
<thead>
<tr>
<th>Plaintext (n bits)</th>
<th>Ciphertext (n bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>For each of the  $2^n$ possible plaintext blocks</td>
<td>The substitute ciphertext block of n bits</td>
</tr>
<tr>
<td>The table has n•$2^n$ bits total</td>
<td></td>
</tr>
</tbody>
</table>

Confidentiality based on the block substitution cipher

- This is a symmetric encryption/decryption algorithm
- The key is the table, which has n•$2^n$ bits
Practicality

- For small block size $n$, statistical techniques can easily infer the table.
- For large block size $n$, the table is too large to be practical.
  - e.g. $n=64$, $n \cdot 2^n = 10^{21}$, far greater than the total storage in a computer.

Practicality (con’t)

- Keys need not be as large for an exhaustive key trial attack.
  - e.g. $10^9$ trials/sec, 10 years = $3 \times 10^8$ sec
  - $3 \times 10^{17}$ trials in 10 years
  - $2^{59} = 6 \times 10^{17}$
  - 59 bit key will do it!

- Conclusion: need an encryption algorithm!
  - Key with 64 or 128 bits may be enough.
Bit-by-bit addition (base-two) for DES symmetric algorithm:

- 64-bit plaintext
- 56-bit key

Confusion and Diffusion:

Plaintext block:

\[ P = \text{plaintext} \]

Ciphertext block:

\[ C = \text{ciphertext} \]

Encrypt:

\[ C = P^s \mod n \]

Decrypt:

\[ P = C^t \mod n \]

C cannot be computed from (n,s) in reasonable time.

RSA asymmetric algorithm:

Understanding Networked Applications 47

A First Course
Notice the asymmetry

The two keys can be applied in either order, and we still return to where we started.

Kerberos

Authentication server

E_{BSK}[k,ABSK,ID_A]

E_{BSK}[k,ABSK,ID_A]

E_{ASK}[k,ABSK,ID_B]

E_{ASK}[k,ABSK,ID_B]