

Next Generation E-Commerce Security

Dr. Çetin K. Koç
koc@ece.orst.edu

Oregon State University
Information Security Laboratory
<http://security.ece.orst.edu>

December 2, 1999

Electronic Commerce

- Purchasing and ordering of products using credit cards
- On-line purchasing of digital content, subscription
- Banking and credit services, payments, investments
- Internal business activities, stock control, ordering
- Business-to-business activities

These activities are performed on the Internet

Security Issue

For electronic commerce to flourish on the Internet, all parties need a way of verifying each other's identities - and establishing trust.

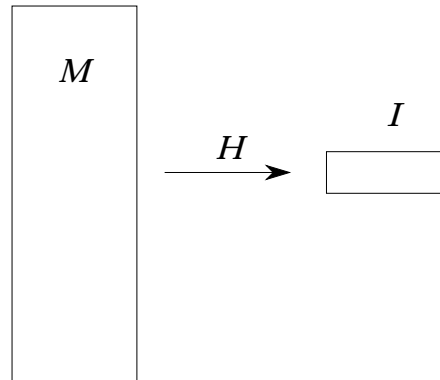
Crypto Basics

Security protocols use a few cryptographic functions:

- Message Digest (integrity)
- Secret Key Encryption (privacy)
- Public Key Encryption (privacy and authentication)
- Digital Envelopes (integrity and privacy)
- Digital Signatures (authentication)
- Digital Certificates (authentication)

Message Digest

$$I = H(M)$$



Short (128 or 160 bits) I is obtained from a long message M

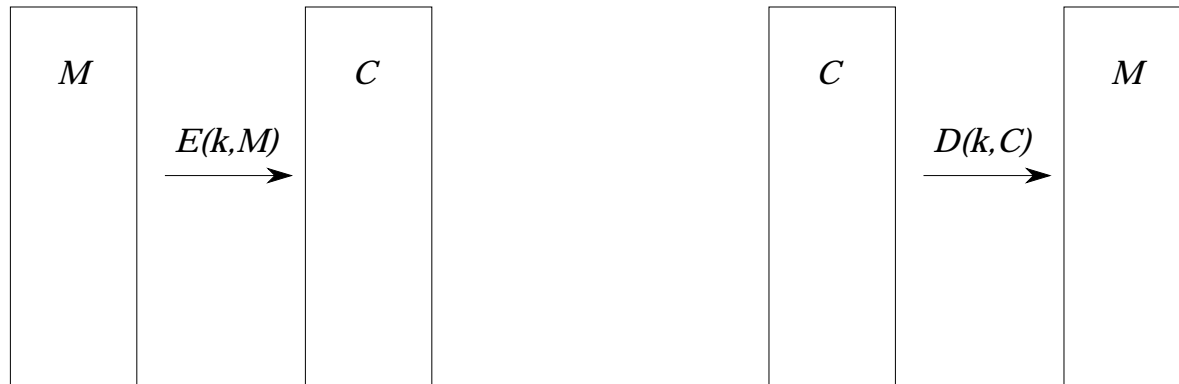
Examples: SHA & MD5

Security requirement: Should be hard to obtain two distinct messages M_1 and M_2 with the same I

Secret Key Encryption & Decryption

Secret Key Encryption: $C = E(k, M)$

Secret Key Decryption: $M = D(k, C)$



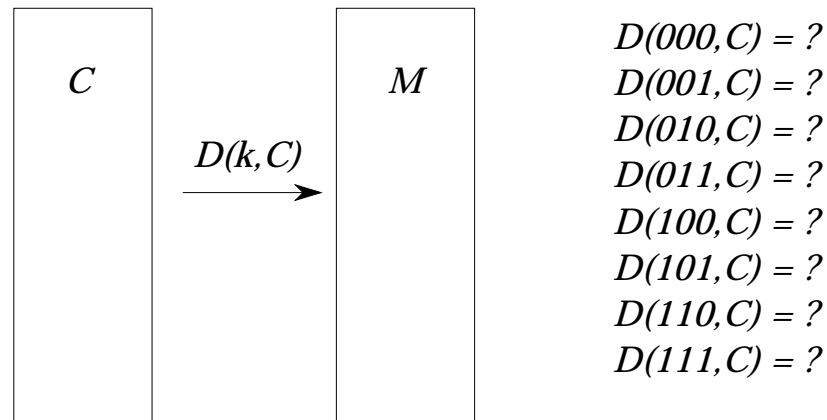
Examples: DES, 3DES, AES, IDEA, RC2, RC4, RC5, CDMF

Security requirement: Should be hard to discover k or M

Breaking Secret Key Encryption Functions

Mathematical attacks: Find shortcuts ... (not so easy)

BRUTE FORCE ATTACK: Try all k values one by one



CDMF: 40 bits (2^{40} tries)
DES: 56 bits (2^{56} tries)
3DES: 112 bits (2^{112} tries)
IDEA: 128 bits (2^{128} tries)

Breaking Secret Key Encryption Functions

What is feasible?

1,000,000 CDMF encryptions take 1 second on a Pentium II

We need about $\frac{2^{40}}{1,000,000} \approx 1.1$ million seconds ≈ 12.7 days!

Use a network of computers – idle CPU time

If we have 13 machines, we need 1 day!!

Breaking Secret Key Encryption Functions

DES (56 bits): 1M encryptions per second:
834,000 days or 1 day with 834,000 machines!

1B encryptions per second: 834 days!

not out of reach ... doable ... **done!** (6/97, 2/98, & 7/98)
(First: **90 days** ... Second: **39 days** ... Third: **56 hours**)

We need at least 80 bits to be secure (for a while)

1M: 10^3 computers \times 38M years ...

1B: 10^3 computers \times 38K years ...

1T: 10^3 computers \times 38 years ... out of reach

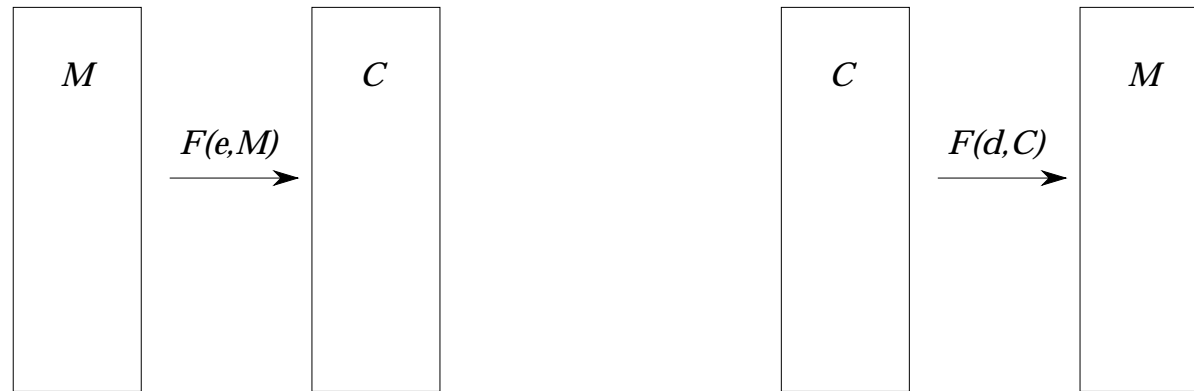
1T: 3DES (112 bits): 10^3 computers \times 10^{15} years

1T: IDEA (128 bits): 10^3 computers \times 10^{20} years

Public Key Encryption & Decryption

Public Key Encryption: $C = F(e, M)$... e is public

Public Key Decryption: $M = F(d, C)$... d is secret



Anyone can encrypt, only the person who knows d can decrypt

Example: RSA, Elliptic Curve, ElGamal

Security requirement: Should be hard to discover d from e

Breaking Public Key Encryption Functions

Mathematical shortcuts are possible: factorization & discrete log (No need to try all keys)

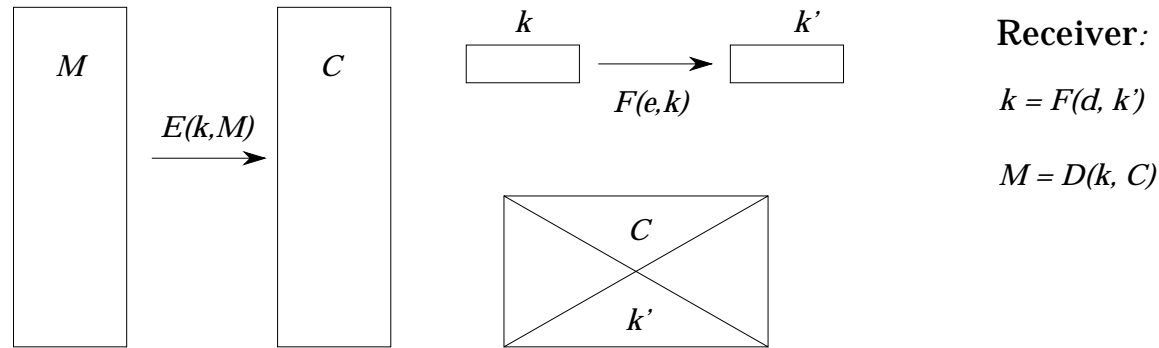
Maximum length RSA which can be broken is about 512 bits
Current standards propose 1024 bits ... out of reach

Maximum length EC which can be broken is about 70 bits
Current standards propose 160 bits or more ... out of reach

Tomorrow: Better algorithms for factorization and discrete logarithm?

Digital Envelopes

Use secret key encryption to encrypt the long message M
Use public key encryption to encrypt the short key k



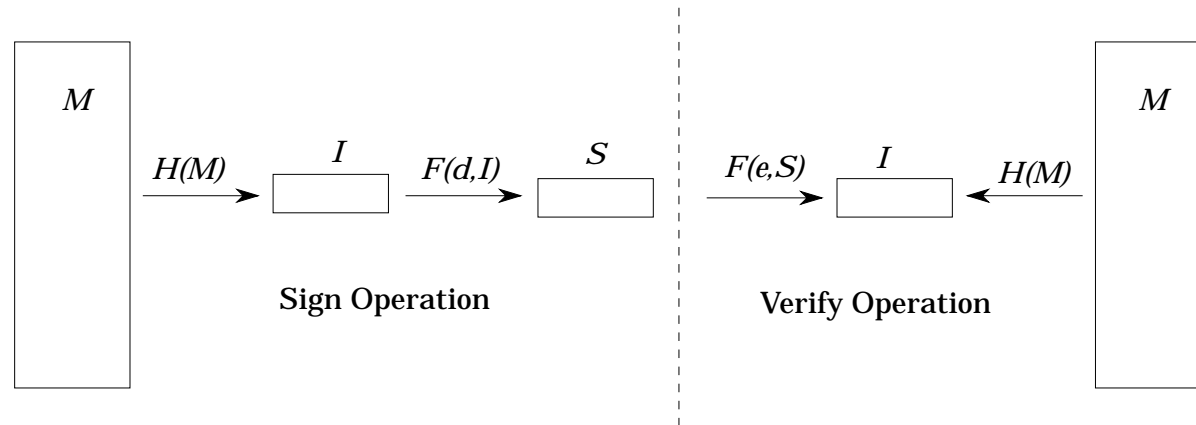
Receiver uses public key decryption with d and obtains k
Receiver uses secret key decryption with k and obtains M

Faster than encrypting M using public key encryption

Security requirement: Should be hard to obtain d from e or to discover k or M

Digital Signatures

Only the owner of d can sign M producing S



Anyone can verify using publicly known e and M

Examples: RSA, DSA, Elliptic curve, ElGamal

Security requirement: Should be hard to obtain d from e or to generate S without knowing d

Digital Certificates

A digital ID signed by a certification authority

“John Doe’s public key is 0100...1”

“Signature of the CA”

Your identification card for Internet transactions

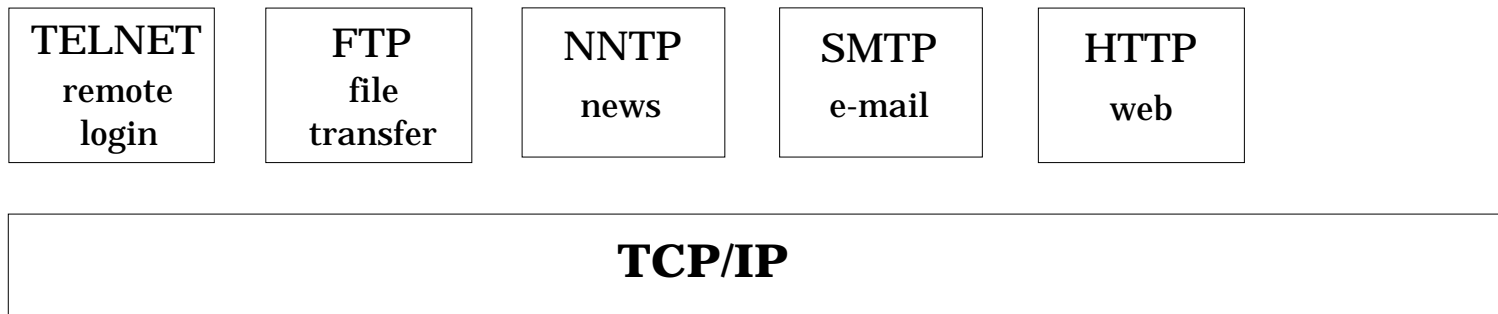
Certification Authorities: Many may coexist

Current digital signature algorithms are RSA and DSA

Elliptic curve cryptography signatures: ECDSA

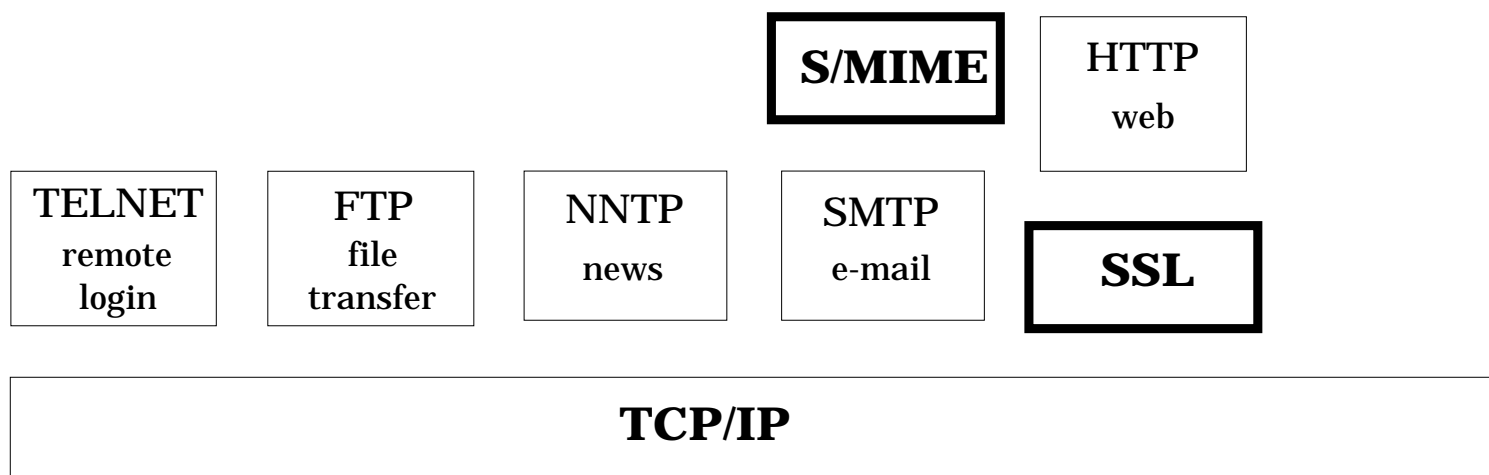
Security requirement: Obtain and verify the certificate
(Need to the signature of the CA)

Historical Internet Protocols



Not secure ...

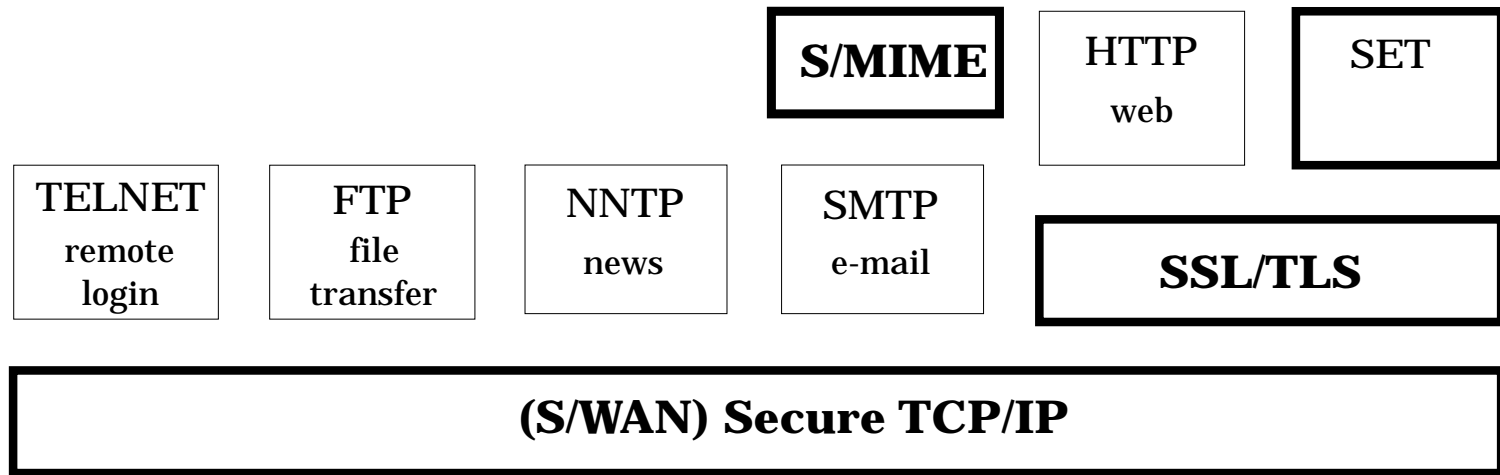
Current Internet Protocols



SSL: Secure Sockets Layer (Web Security)

S/MIME: Secure Multipart Mail

Emerging Internet Protocols



SET: Secure Electronic Transactions

TLS: Transport Layer Security (based on SSL 3.0)

S/WAN: Secure TCP/IP

Secure Sockets Layer and Transport Layer Security

SSL protects the communication between a client and a server and provides authentication to both parties

SSL (TLS) resides below application protocols & above transport protocols

Most common use: to secure web protocol HTTP
(Works with other protocols as well)

Most common use: to secure TCP/IP
(Works with other protocols as well)

Primitives: Message digest, public key encryption, secret key encryption, digital signatures, & certificates

Secure Sockets Layer and Transport Layer Security

Broadly supported: Netscape, Microsoft, Oracle, Lotus

SSL 3.0 (and TLS 1.0) supports **weak crypto**:

- 40-bit encryption (RC4 and CDMF)
- no encryption

Exportable and US versions of crypto toolkits

Exportable and US versions of security applications

SET: Secure Electronic Transactions

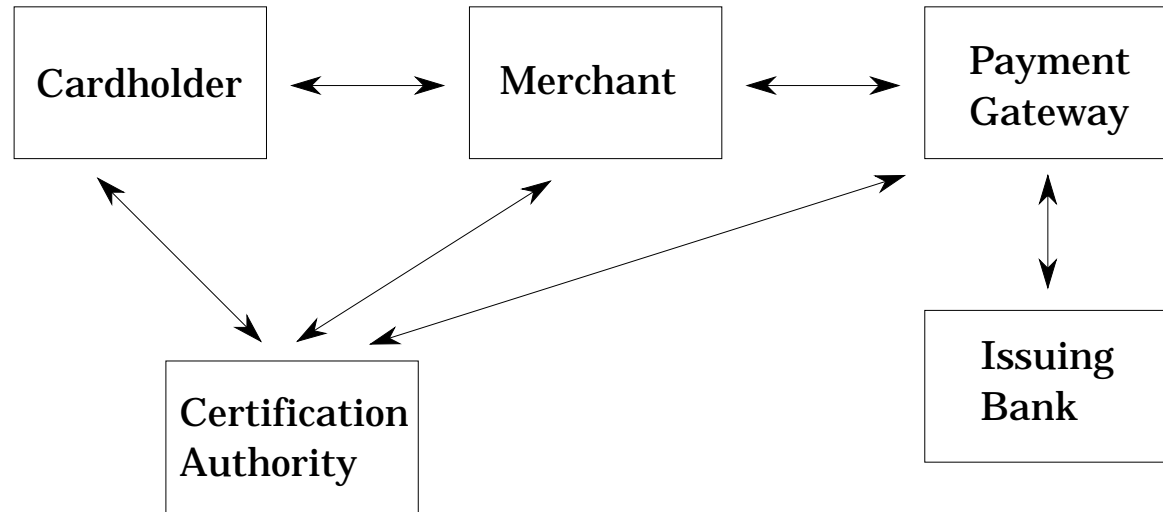
Mastercard & Visa collaboration: a technical **standard** for credit card authorization on the Internet

Reduce fraud by consumers, merchants, and hackers

- Confidentiality of payment information
- Confidentiality of order information
- Data integrity
- Authentication of the cardholder
- Authentication of the merchant

Primitives: Message digest, public key encryption, secret key encryption, digital signatures, & certificates

SET: Secure Electronic Transactions



CA provides certificates for C, M, and PG

C purchases and provides payment information to M

M deliver goods and gives credit information to IB by PG

IB pays M

SET: Secure Electronic Transactions

Message digest: SHA & HMAC

Secret key encryption: DES (56 or 112 bits)

CDMF (40 bits)

Digital envelopes & signatures: RSA (1024 bits)

Elliptic curves are added

Certification method: X509 v2 & v3

Exportable and US versions of SET toolkits

Exportable and US versions of SET applications

Problems with SET

Significant changes in the existing payment infrastructure

Changes in the business model

Slow

Banks	need	750	transactions	per	second
SET	offers	1	transaction	per	second

Banks are unwilling to build

Not a single bank deployed SET as of now

SET needs to be **revived**

Existing Payment Infrastructure

Customer - Merchant - Transaction Company - Banks

Account based methodology

Non-face-to-face authentication:

- mother's maiden name

- social security number

- full address

- PINs

“Add” strong authentication to the payment structure

An emerging, alternative technology:

AADS: Account Authority Digital Signatures

AADS: Account Authority Digital Signatures

A simple extension of the existing account-based method

Registering the public-key: registering a PIN

Maintains the current business environment

Eliminates the need to append a certificate to a digitally signed transaction: **document + digital signature**

Authentication at the center: retrieve the public-key from the account

AADS: Account Authority Digital Signatures

Standardization: X9.59

No need for “certificate authorities”

No changes in the business model

No modifications (or small modifications) to the existing payment infrastructure (hardware & software)