E-mail Security

Dr. Natarajan Meghanathan
Associate Professor of Computer Science
Jackson State University
E-mail: natarajan.meghanathan@jsums.edu

Motivation for E-mail Security

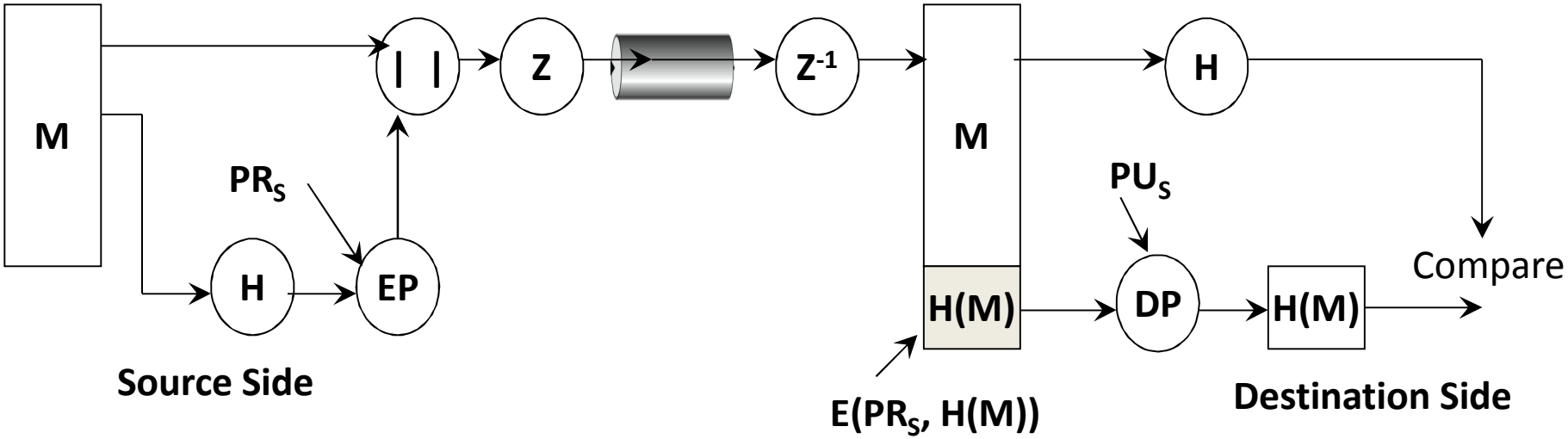
- Electronic mail (E-mail) is one of the widely used and regarded network –based application in virtually all distributed environments.
- Currently, message contents are not secure.
  - May be inspected either in transit or by suitably privileged users on destination systems
- Requirements for E-mail Security
  - Confidentiality: protection from disclosure
  - Authentication of sender of message
  - Message integrity: protection from modification
  - Non-repudiation of origin: protection from denial by sender
- PGP (Pretty Good Privacy) and S/MIME (Secure Multi-purpose Internet Mail Extensions) are the two commonly used E-mail Security Standards. DKIM is a third e-mail security standard.

E-mail Security Standards

- Pretty Good Privacy (PGP), Secure Multi-purpose Internet Mail Extensions (S/MIME) and Domain Keys Identified Mail (DKIM)
- Both PGP and S/MIME provide confidentiality and authentication services that can be used for E-mail and file storage applications.
- PGP was originally designed for plaintext messages; S/MIME handles all sorts of data files (for e.g., spreadsheets, graphics, presentations, movies, and sound) and is integrated into many commercial e-mail packages – hence S/MIME is likely to dominate the secure e-mail market.
- With PGP, each user is free to decide on the level of trust for a public key of a remote user (usually done based on a "web of trust" among the users) and not done through a centralized Certificate Authority (CA) to issue certificates for the public keys.
- S/MIME uses the standard public-key certificates issued by a CA.
- S/MIME also provides several options for the algorithms to be used for confidentiality and authentication services; whereas PGP is fixed with respect to the algorithms that can be used.
- Support for S/MIME is built-in to most of the standard browsers; while PGP typically requires a user to download and install one or more plug-ins.

PGP for Authentication

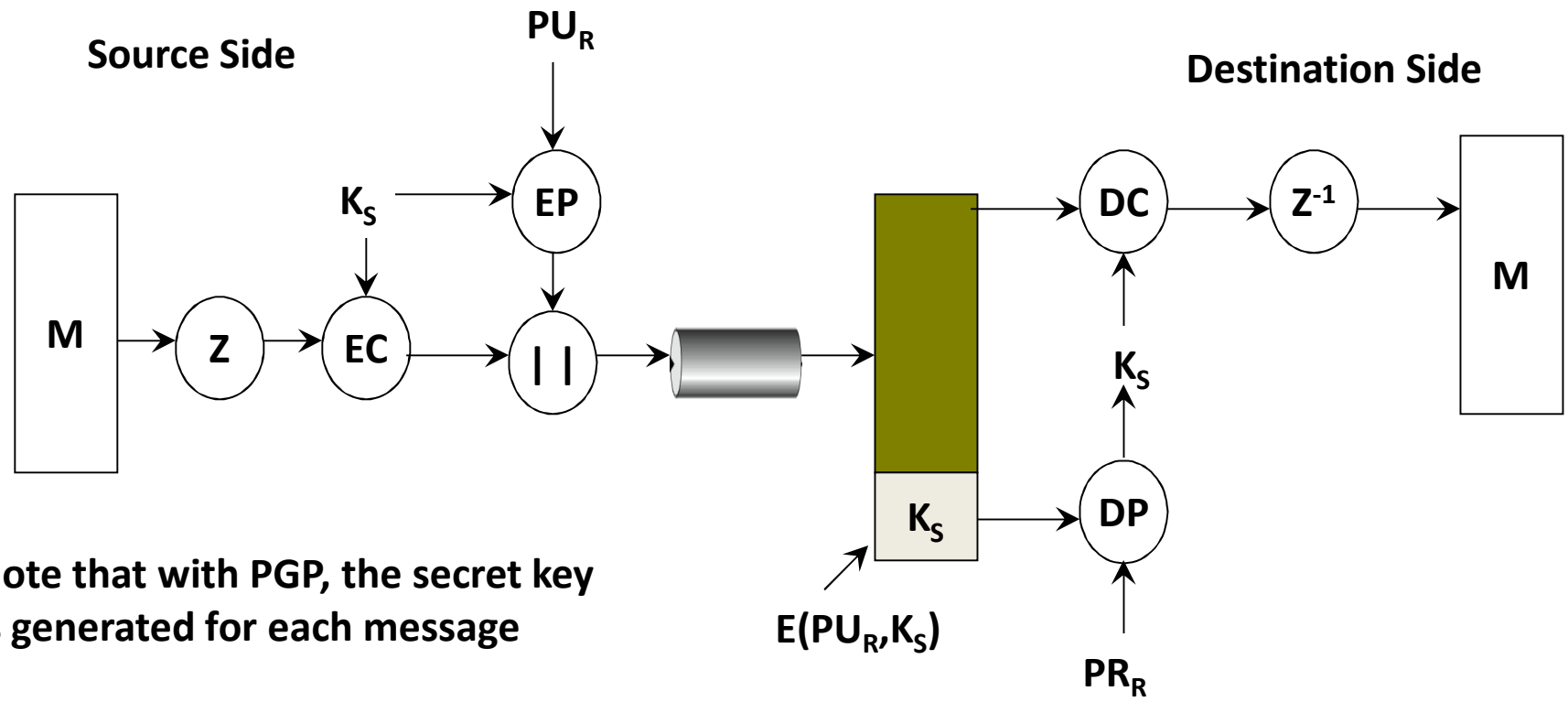
ZIP [M || $E_{Pri\text{-}Sender}$(H(M))]



Hashing Algorithm, H - SHA-1160 bit hash
Encryption for digital signature (public-key encryption), EP – RSA
Compression, Z – ZIP algorithm

Source (adapted from): Figure 18.1 (a), from William Stallings – Cryptography and Network Security, 5th Edition

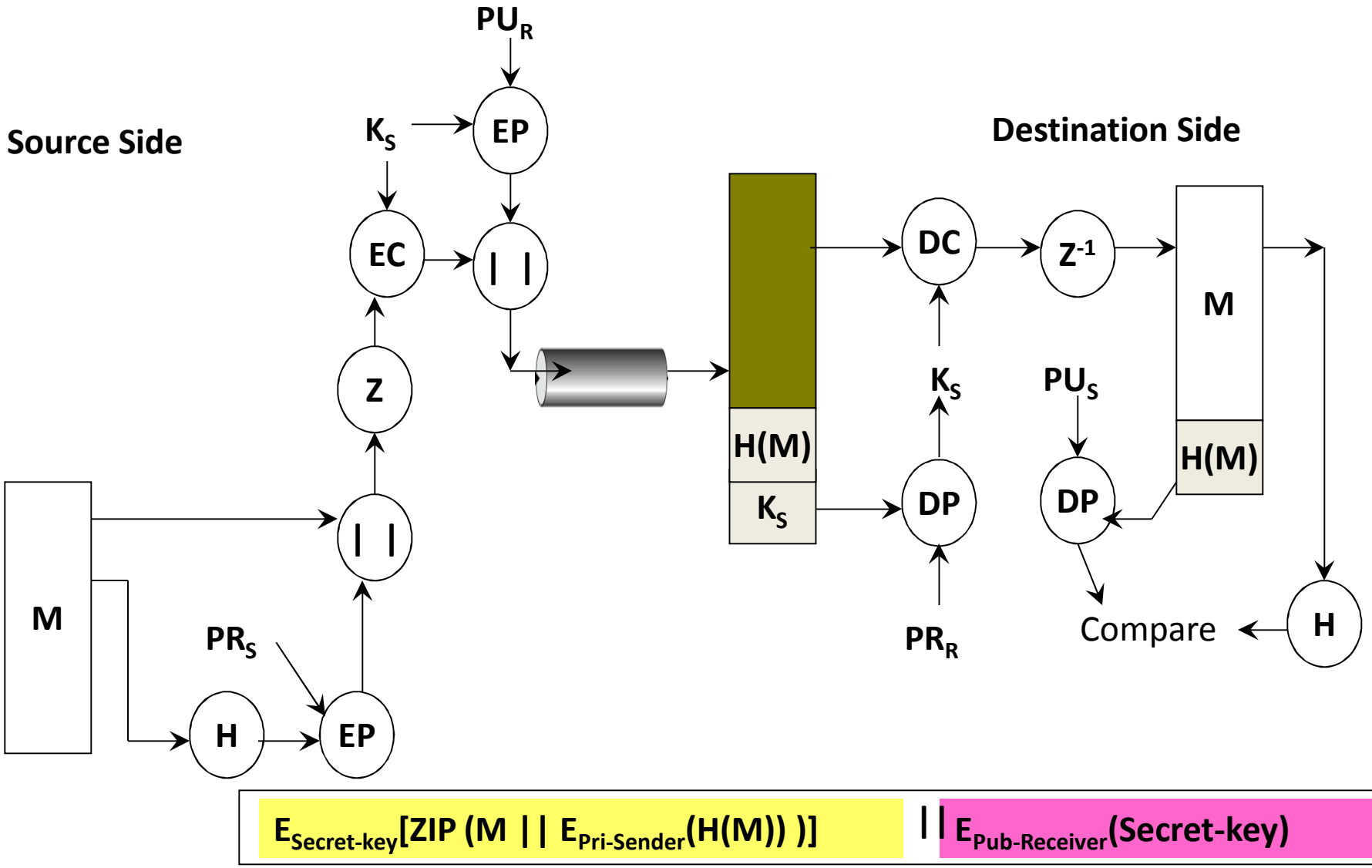
PGP for Confidentiality



Encryption for confidentiality, EC – IDEA (International Data Encryption Algorithm)

Source (adapted from): Figure 18.1 (b), from William Stallings – Cryptography and Network Security, 5th Edition

PGP for Authentication and Confidentiality

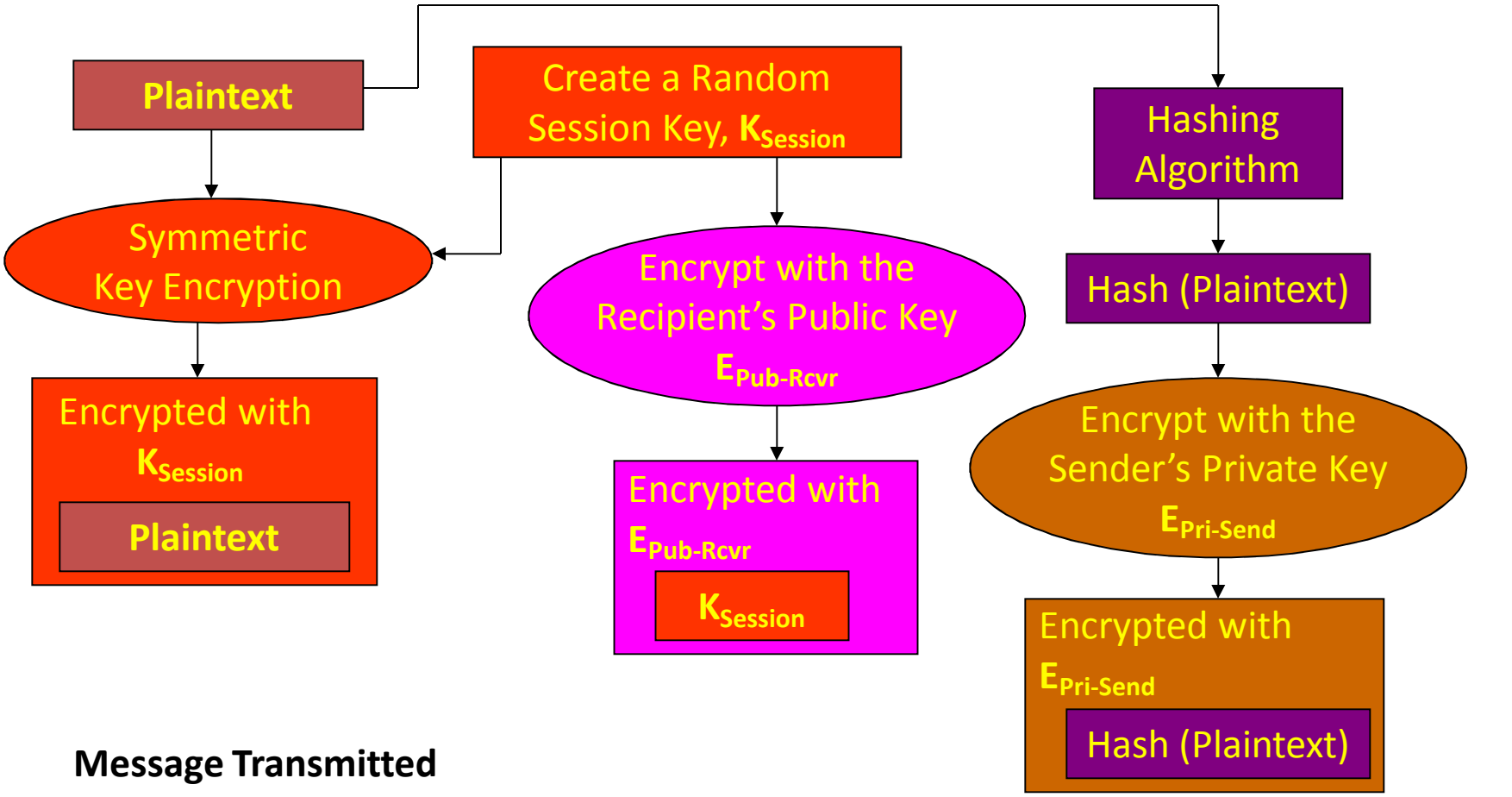


Source (adapted from): Figure 18.1 (c), from William Stallings – Cryptography and Network Security, 5th Edition

PGP Keys

- PGP Session Key
- The session key is associated with a single message and is used only for the purpose of encrypting and decrypting that message.
- IDEA uses a 128-bit symmetric key.
- Session keys are generated using the ANSI X12.17 generator, based on keystroke input from the user, where both the keystroke timing and the actual keys struck are used to generate a randomized stream of numbers constituting the key.
- PGP Public and Private Keys
- Since many public/private keys may be in use with PGP, there is a need to identify which key is actually used to encrypt the session key for any specific message.
- Since, it would be inefficient to send the full public-key with every message, PGP uses a key identifier based on the least significant 64-bits of the key, which will very likely be unique. Then only the much shorter key ID would need to be transmitted with any message.
- A key ID is also required for the PGP digital signature.

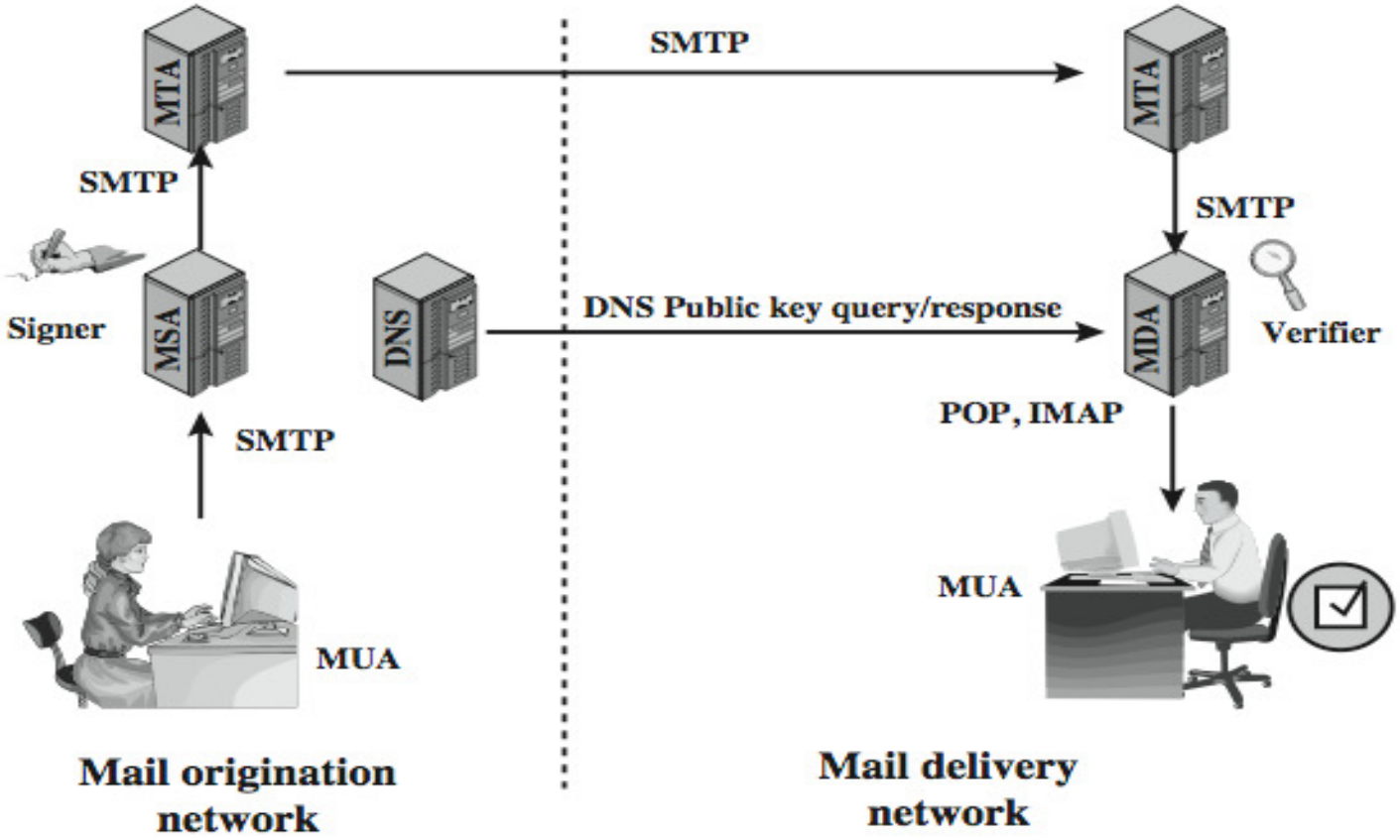
S/MIME


Plaintext
Create a Random Session Key, $K_{Session}$
Hashing Algorithm
Symmetric Key Encryption
Encrypt with the Recipient's Public Key $E_{Pub-Rcvr}$
Hash (Plaintext)
Encrypted with $K_{Session}$
Plaintext
Encrypted with $E_{Pub-Rcvr}$
$K_{Session}$
Encrypt with the Sender's Private Key $E_{Pri-Send}$
Encrypted with $E_{Pri-Send}$
Hash (Plaintext)
Message Transmitted
Encrypted with $K_{Session}$
Plaintext
Encrypted with $E_{Pub-Rcvr}$
$K_{Session}$
Encrypted with $E_{Pri-Send}$
Hash (Plaintext)
Encrypted with $E_{Pub-Rcvr}$
Sender's Public-Key Certificate

Domain Keys Identified Mail (DKIM)

- With DKIM, an e-mail message is signed using the private key of the administrative domain from which the e-mail originates (not signed with the private key of the individual user, as in PGP and S/MIME).
- DKIM is transparent to the end-users as the e-mail is signed by the Mail Submission Agent (MSA) of the sender's domain and validated by the Mail Delivery Agent (MDA) of the recipient's domain.
- At the receiving end, the MDA can access the public key of the sender's administrative domain (through DNS query) and verify the authenticity of the message.
- E-mails that claim to originate from a particular domain – but, not signed with the private key of the administrative domain – are rejected.
- The default signing algorithm is RSA with SHA-256.

DKIM



DNS = domain name system
MDA = mail delivery agent
MSA = mail submission agent
MTA = message transfer agent
MUA = message user agent

(to send and propagate e-mail messages)
SMTP – Simple Mail Transfer Protocol
(to retrieve e-mail message from the MDA to the MUA)
POP – Post Office Protocol
IMAP – Internet Message Access Protocol
http://www.youtube.com/watch?v=GcvdhjLVYfY

Source: Figure 18.10 from William Stallings – Cryptography and Network Security, 5th Edition