CS 465

Secure Email

Last Updated: Nov 30, 2017

Goals

- Be able to describe how secure email works to provide confidentiality, integrity, and authentication
- Understand trust models
  - PGP
  - S/MIME
- Gain hands-on experience using secure email

Email

- Is your email secure?
- What is the threat model?

PGP (Pretty Good Privacy)

- Designed by Phil Zimmerman
  - Originally designed as a human rights tool
  - Published for free on the Internet in 1991
  - Phil was the target of a three year criminal investigation
- Where to get PGP?
  - http://www.philzimmermann.com/EN/findpgp/index.html
    - pgp.com
    - GnuPG (GPG)
- In the 1990's, one way to skirt federal export controls was to publish the source code in book form (this was allowed), ship the books to Europe, scan the source code using OCR technology to create the code. Laborious, but legal.
- Trust model
  - Web of trust
  - Grass roots, bottom up
  - Manual key management

S/MIME

- Secure Multipurpose Internet Mail Extension
- Security extension to the MIME Internet email format
- Trust model
  - Hierarchical
  - CAs that issue X.509 certificates
  - Top-down
- Used by companies and government orgs

Key Management

- Recommended that you use a different public key for signing outgoing and decrypting incoming email
- PGP
  - Generate your own keys
  - You are responsible to distribute
    - In person or through key servers
    - Key signing parties!
- S/MIME
  - Generate your own keys
  - Have your public key signed by a CA
  - Start sending signed email
    - Your public key is sent along with a signed message

How Secure Email is Sent

- The following example is taken from a PGP description
- Sign-then-encrypt
- The way that both symmetric encryption and asymmetric encryption are used together is common to many secure messaging systems

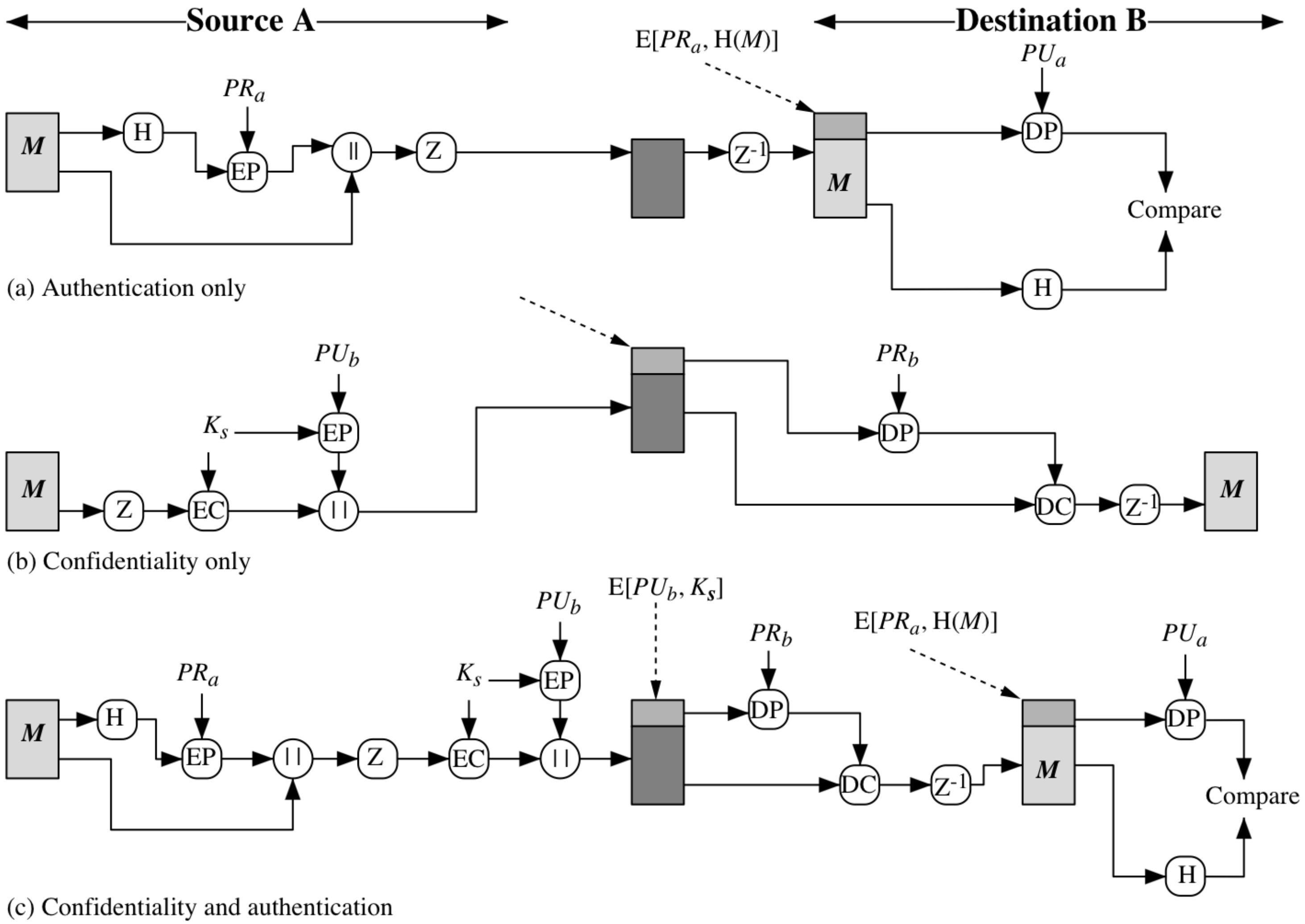
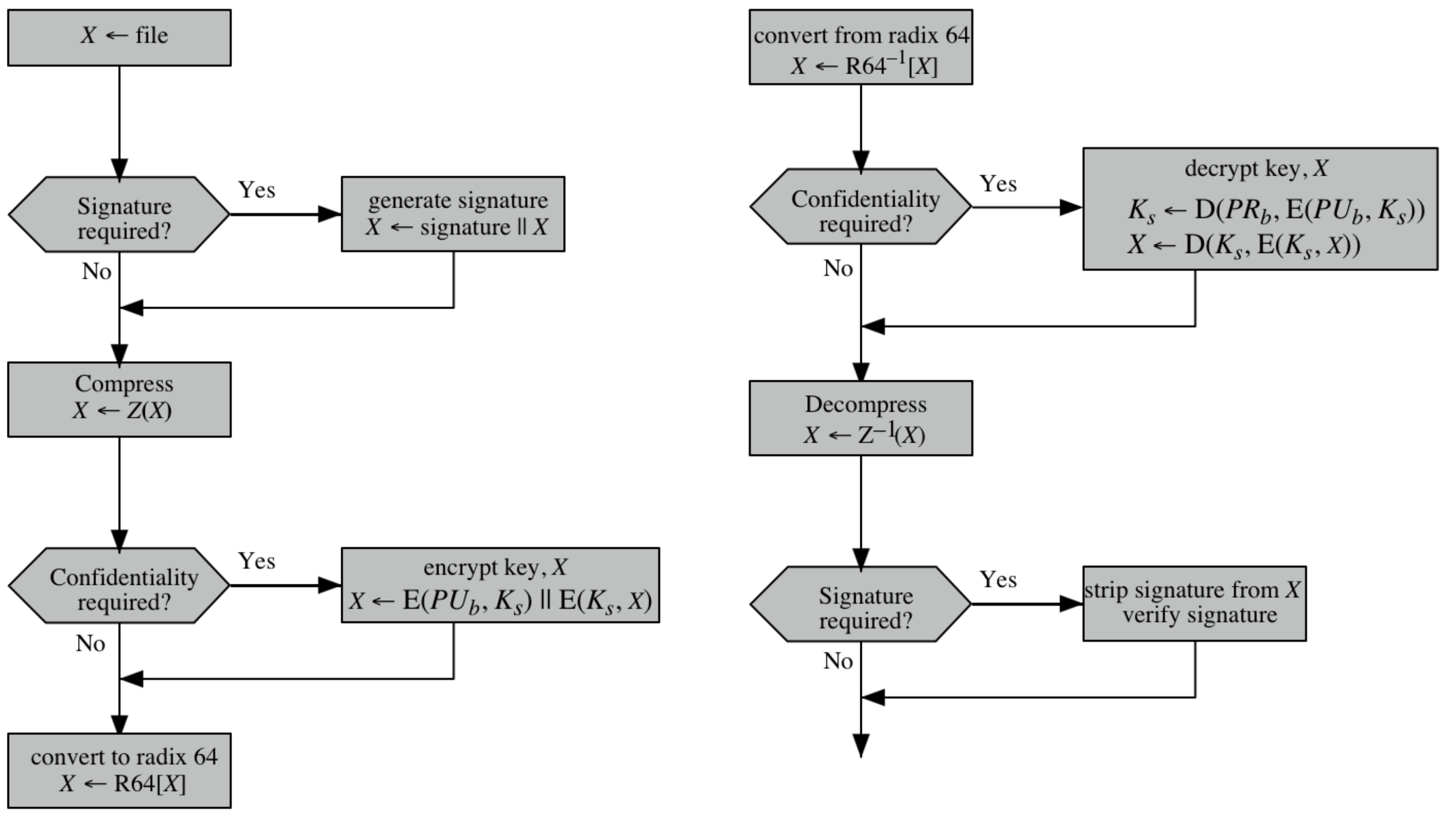


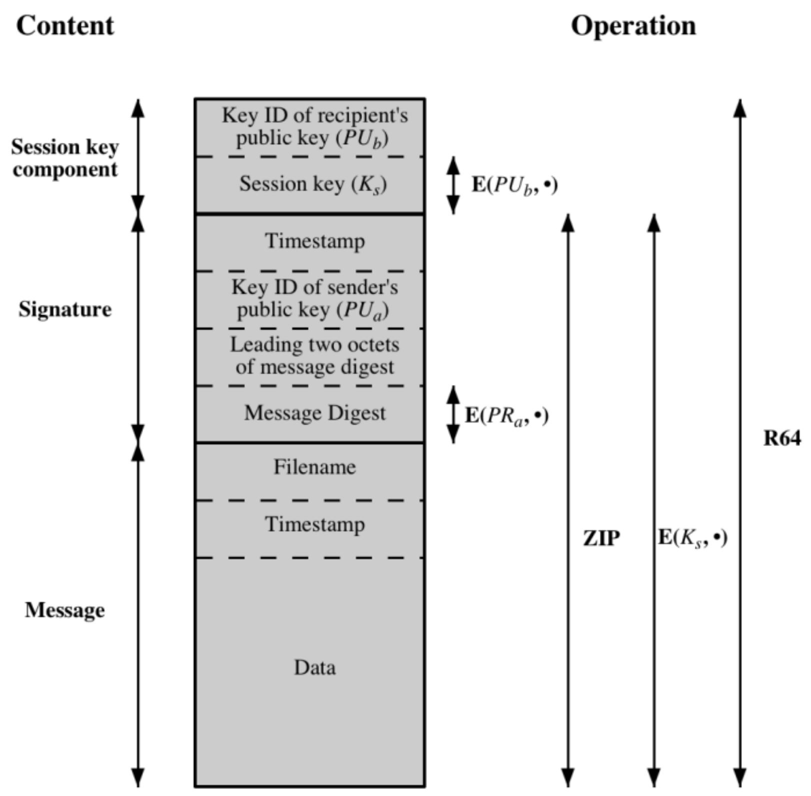
Figure 5.1 PGP Cryptographic Functions



$X \leftarrow$ file
Signature required?
Yes
No
generate signature $X \leftarrow$ signature $\|$ X
Compress $X \leftarrow Z(X)$
Confidentiality required?
Yes
No
encrypt key, X $X \leftarrow E(PU_b, K_s) \| E(K_s, X)$
convert to radix 64 $X \leftarrow R64[X]$
convert from radix 64 $X \leftarrow R64^{-1}[X]$
Confidentiality required?
Yes
No
decrypt key, X $K_s \leftarrow D(PR_b, E(PU_b, K_s))$ $X \leftarrow D(K_s, E(K_s, X))$
Decompress $X \leftarrow Z^{-1}(X)$
Signature required?
Yes
No
strip signature from X verify signature

(a) Generic Transmission Diagram (from A)   (b) Generic Reception Diagram (to B)

Figure 5.2 Transmission and Reception of PGP Messages



Notation:

E(PU_b, •)	=	encryption with user b's public key
E(PR_a, •)	=	encryption with user a's private key
E(K_s, •)	=	encryption with session key
ZIP	=	Zip compression function
R64	=	Radix-64 conversion function

Figure 5.3 General Format of PGP Message (from A to B)

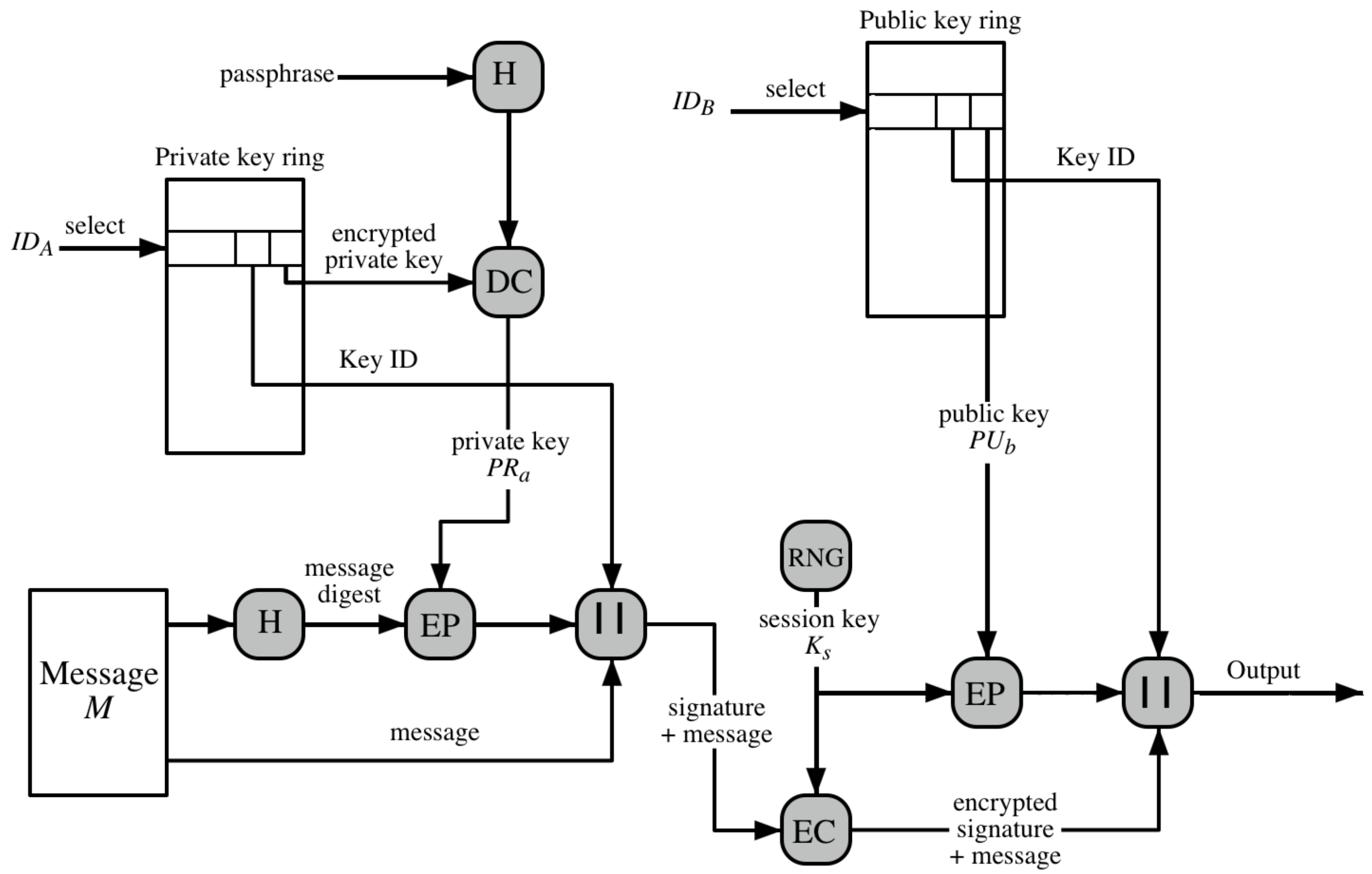


Figure 5.5 PGP Message Generation (from User A to User B; no compression or radix 64 conversion)

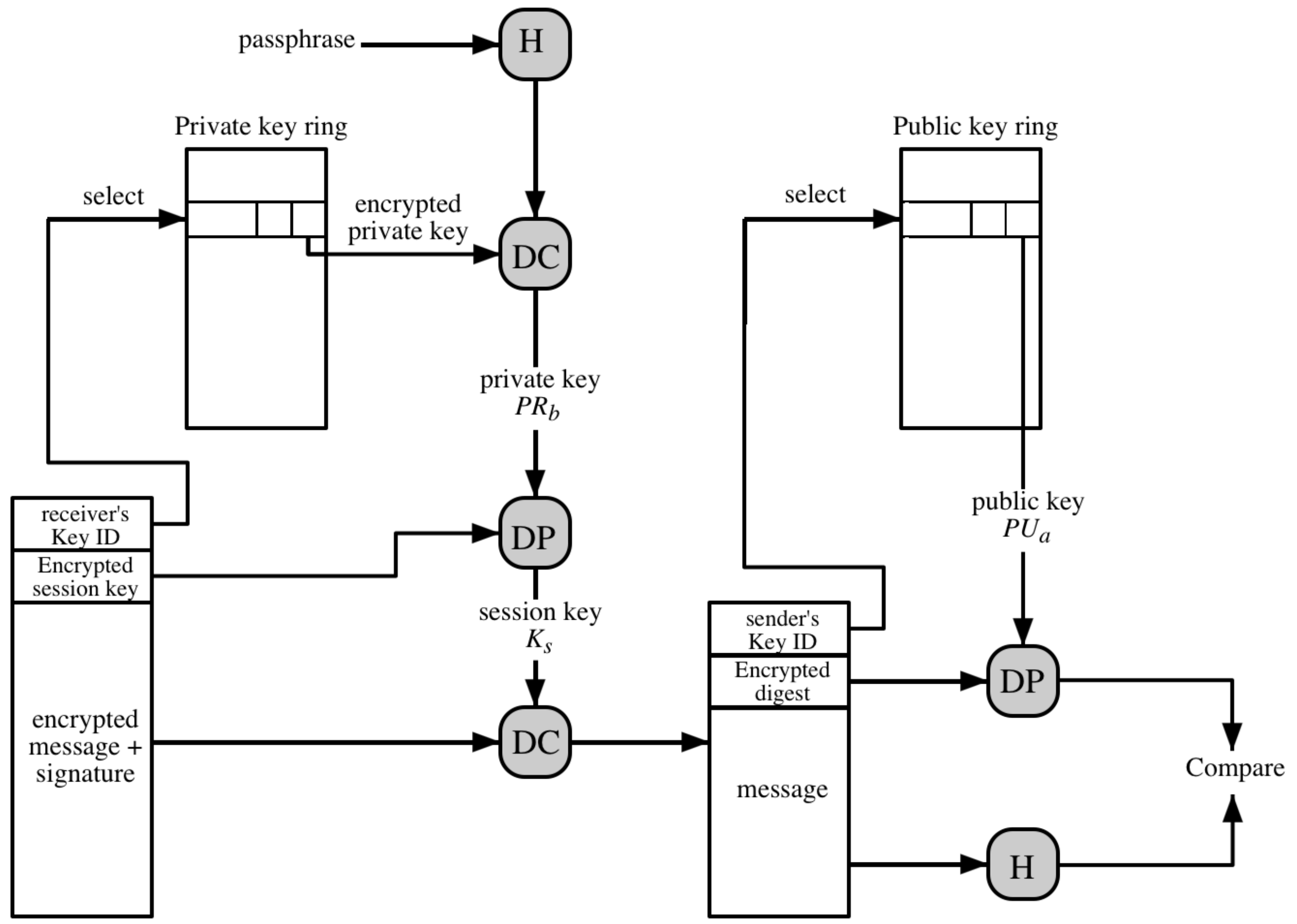


Figure 5.6 PGP Message Reception (from User A to User B; no compression or radix 64 conversion)

How Secure is our Email?

- Until the last decade, most email was sent and received in the clear over the network
- FireSheep prompted webmail providers to use HTTPS to protect browser connections (2010)
- Email is usually stored in the clear
  - Free email supported by ad revenue
  - Search, spam/malware protection
- Connections between providers is sometimes encrypted (STARTTLS)

Email Encrypted in Transit

- HTTPS by default (2010)
  - https://gmail.googleblog.com/2010/01/default-https-access-for-gmail.html
- Mandated HTTPS from the browser (2014)
  - https://googleblog.blogspot.com/2014/03/staying-at-forefront-of-email-security.html

Email Authenticity

- DKIM (Domain Keys Identified Email)
  - Providers sign email sent from their domain
  - Public key published in DNS
  - Clients validate the signature (problematic)
- SPF (Sender Policy Framework)
  - Authorized email hosts published in DNS
- Google visual indicators for email sent in the clear and not authenticated
  https://blog.google/products/gmail/making-email-safer-for-you-posted-by/

Why is Secure Email Not Used?

- Users accept the risk
- Usability challenges of key management
  - Chicken and egg problem
- PGP history of usability studies that failed
  - Why Johnny Can’t Encrypt (1999)
  - Why Johnny Still Can’t Encrypt (2006)
  - Why Johnny Still, Still Can’t Encrypt (2015) – BYU
    - Schneier reacts - https://www.schneier.com/blog/archives/2015/11/testing_the_usa.html
- Recent success with secure messaging apps supporting end-to-end encryption
  - Signal, WhatsApp, Facebook Messenger, Google Allo
  - Prevents passive attacks
  - Key validation is not done, so active attack may be possible