

SECURE EMAIL

Goals

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

PGP (Pretty Good Privacy)



- ⦿ Designed by Phil Zimmermann
 - 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

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
- ⦿ Usually found in companies and government

Key Management

- ⦿ Recommended that you use a different public key for signing outgoing and receiving incoming encrypted 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 most secure messaging systems

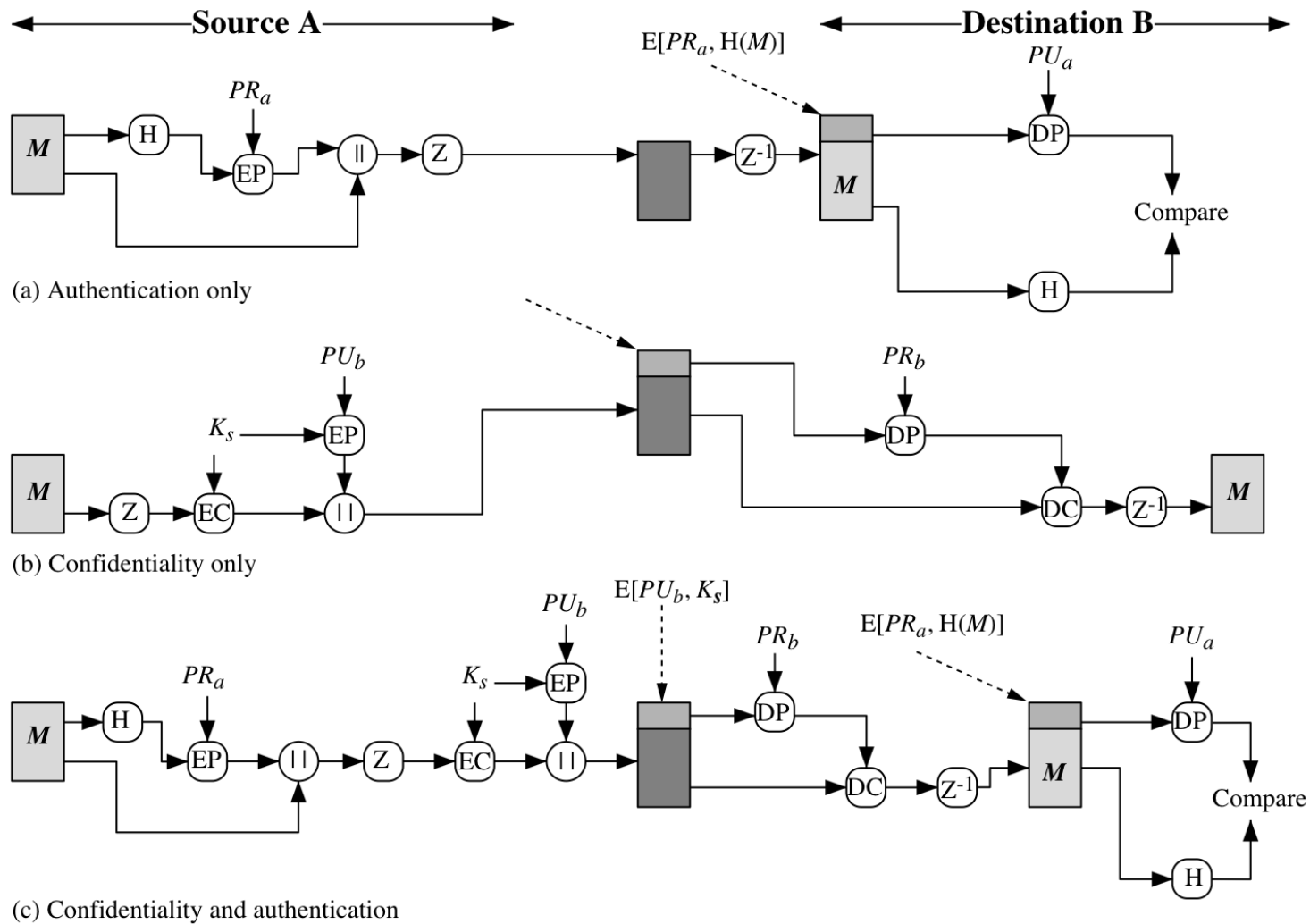
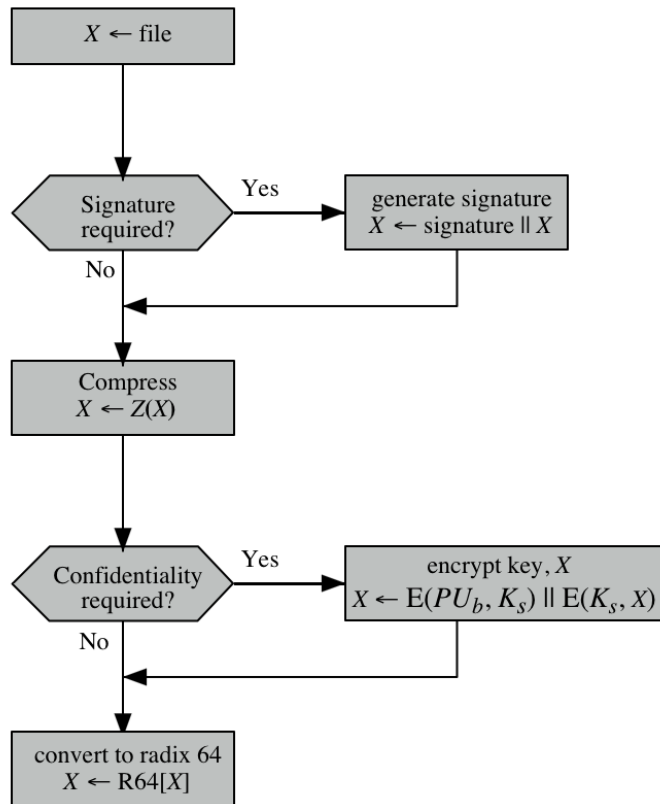
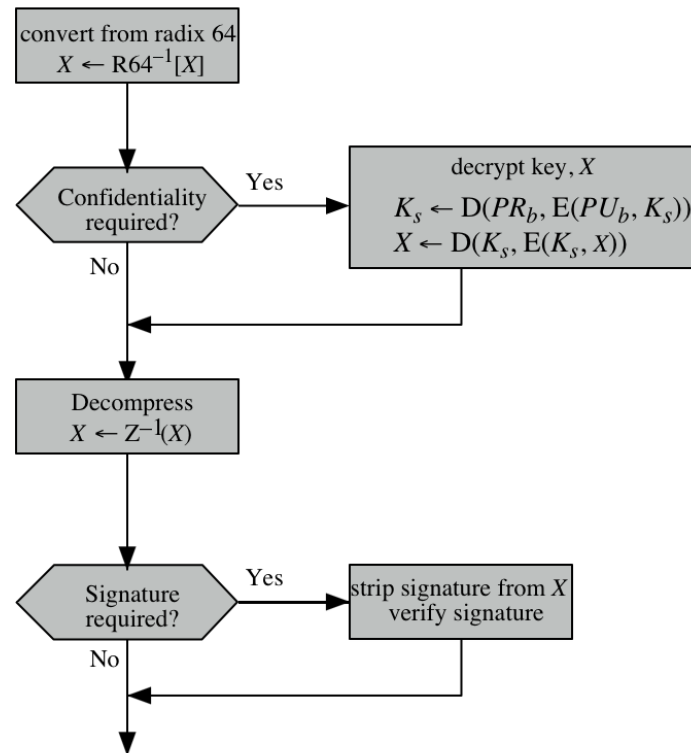


Figure 5.1 PGP Cryptographic Functions



(a) Generic Transmission Diagram (from A)

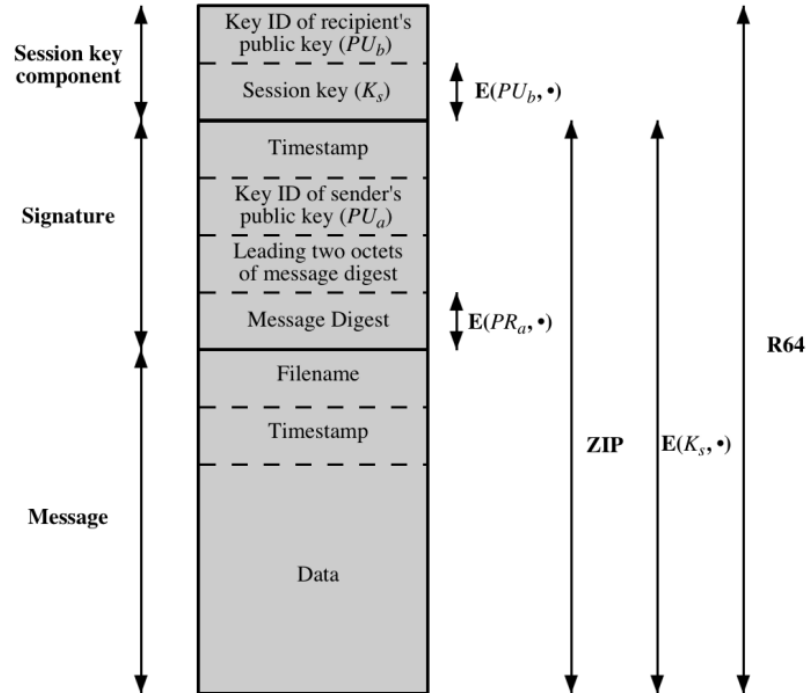


(b) Generic Reception Diagram (to B)

Figure 5.2 Transmission and Reception of PGP Messages

Content

Operation



Notation:

- $E(PU_b, \bullet)$ = encryption with user b's public key
- $E(PR_a, \bullet)$ = encryption with user a's private key
- $E(K_s, \bullet)$ = 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)

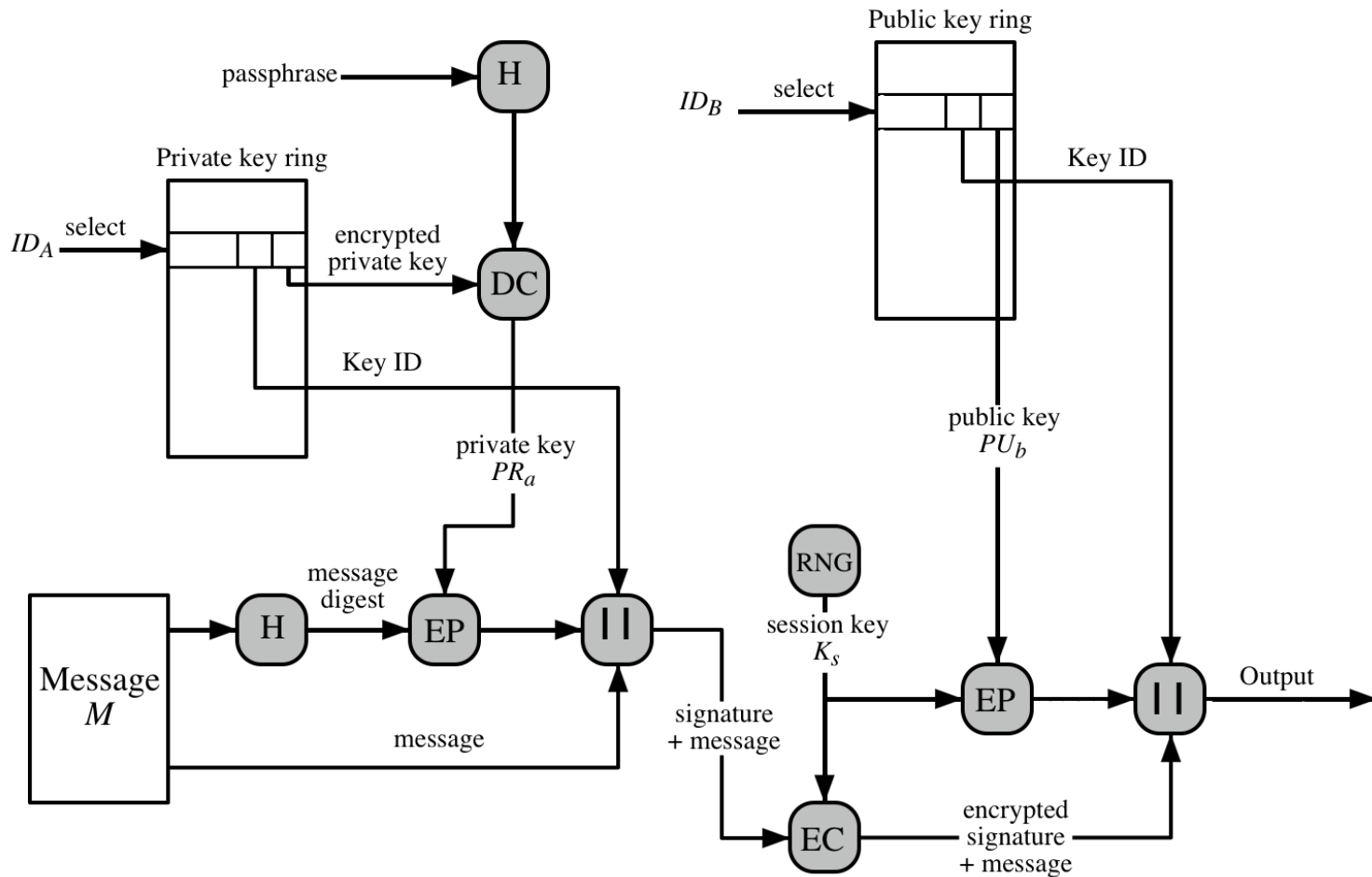


Figure 5.5 PGP Message Generation (from User A to User B; no compression or radix 64 conversion)

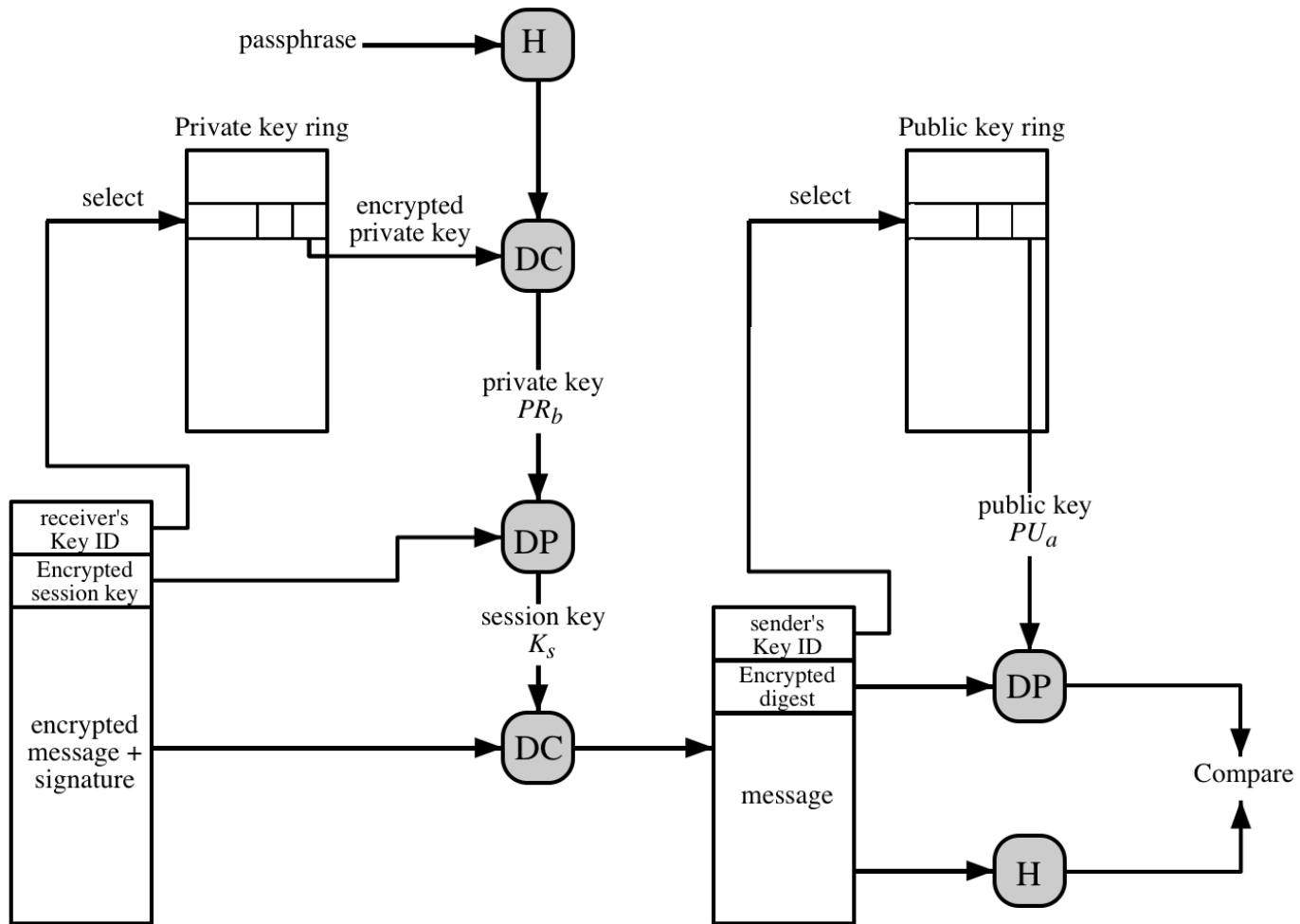


Figure 5.6 PGP Message Reception (from User A to User B; no compression or radix 64 conversion)

How Secure is our Email?

- ⦿ Until recently, most email was sent and received in the clear over the network
- ⦿ FireSheep prompted webmail providers to use SSL to protect connections.
- ⦿ Email is usually stored in the clear
 - Free email supported by ad revenue
- ⦿ Connections between providers is sometimes encrypted (STARTTLS)

Email Authenticity

- ⦿ DKIM (Domain Keys Identified Email)
 - Providers sign email sent from their domain
 - Public key published in DNS
 - Clients validate the signature (problematic)
- ⦿ 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?

- ⦿ 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.