PGP

Pretty Good Privacy

Overview

There are a number of encryption standards and schemas that employ one or more of the encryption approaches previously discussed

Examples include

DES	*AES*	***PGP***
IPsec	*SSH*	*SSL*
HTTPS	*MD5*	*IDEA*
S/key	*Tokens*	*SOCKS*
VPN	*Kerberos*	*GPG*

Pretty Good Privacy

PGP is instructive to look at

- *Practical example*
- *Combines features of* **confidentiality**, **authentication**, *and* **integrity** *and* ***non-repudiation*** *in ways similar to what we discussed in a previous lecture*
- *Typical of approaches that are used today*

Pretty Good Privacy

Created by Philip Zimmermann and small group.

Published in 1995

Zimmermann was more IETF than the IETF

- *IETF developed PEM (Privacy Enhanced Mail) going the official standards route*
  - Progress was slow. Why?
  - The IETF use to be attended by "propeller heads"
    - *No suits or ties!*
  - Now the IETF has as many "suits" as "propeller heads"
- *Zimmermann's group just did it and created working code*

PGP and the Law

During the early to mid 1990s the U.S. Gov't. said that giving foreign persons the ability to obtain PGP was a violation of security laws

- ***Treason!***

U.S. Gov't. has now relaxed its position on exporting encryption technology

- *Versions now produced outside the U.S.*

IETF

- *IETF had sort of a NIH complex over PGP*

Pretty Good Privacy

PGP has

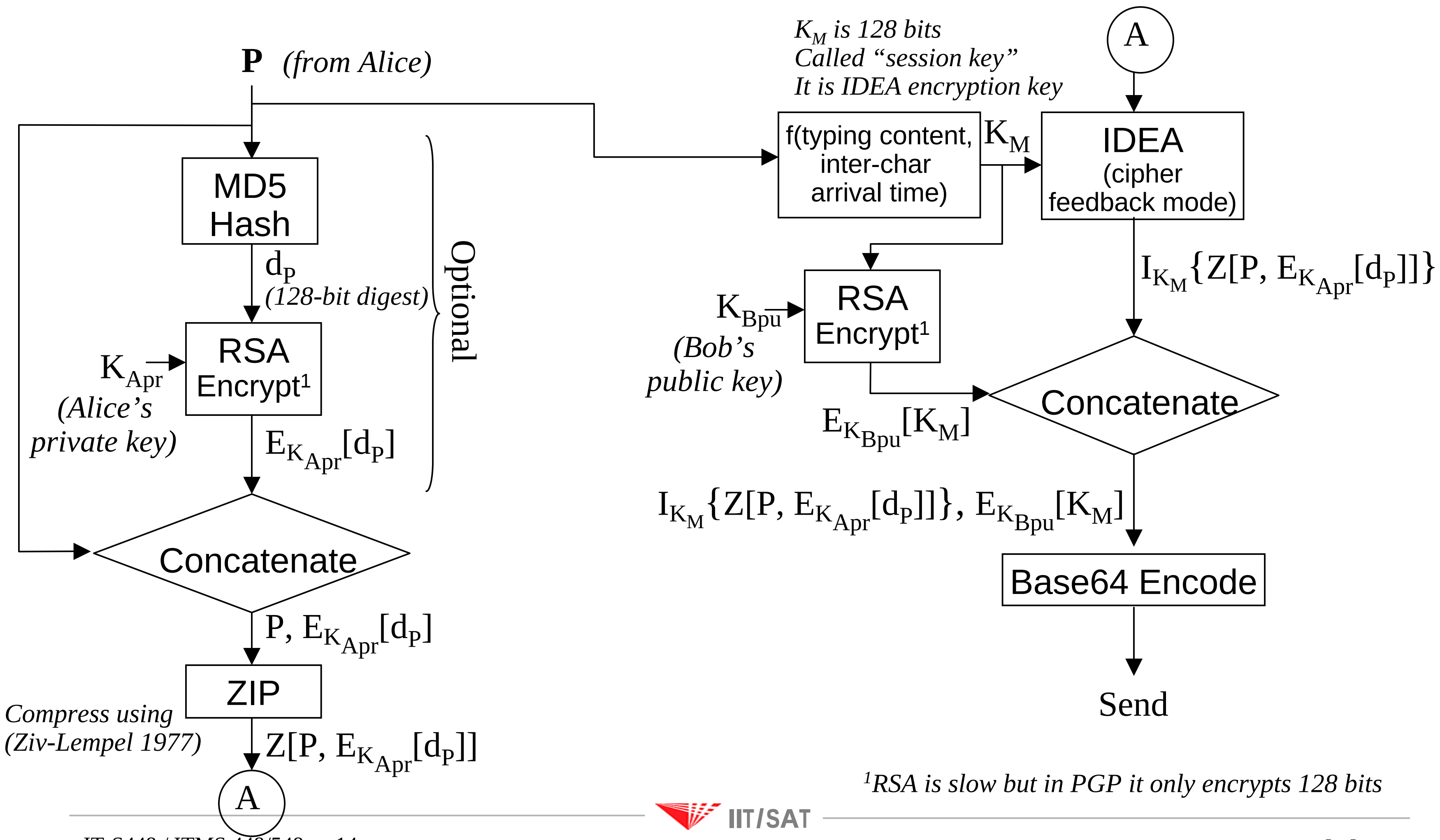
- *privacy,*
- *authentication,*
- *digital signatures, and*
- *compression*

Use to be free, but now Zimmerman sells it

- *Free versions still obtainable over the Internet*
- *e.g., open PGP*
    - *GPG: Gnu Privacy Guard*

Runs on many flavors of UNIX, MacOS, Windows

PGP Encoding
What Alice Does



[1]*RSA is slow but in PGP it only encrypts 128 bits*

PGP Decoding
What Bob Does

(5) IDEA decrypts $I_{K_M}\{Z[P, E_{K_{Apr}}[d_P]]\}$ using K_M'

$D_{K_M'}\{I_{K_M}\{Z[P, E_{K_{Apr}}[d_P]]\}\}$

$= Z[P, E_{K_{Apr}}[d_P]]$

(4) Develops the symmetric decrypting key K_M' from K_M

K_M' is the IDEA decryption key

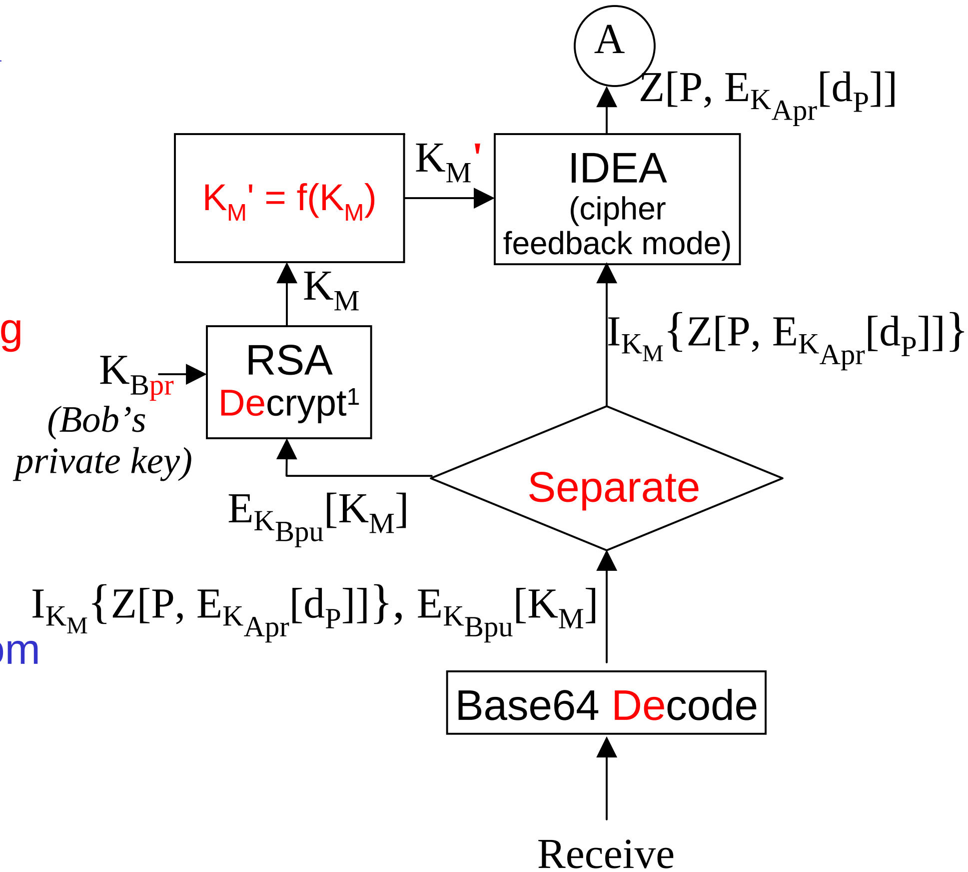
(3) RSA decrypts K_M

$D_{K_{Bpr}}[E_{K_{Bpu}}[K_M]] = K_M$

(2) Separates encrypted IDEA key from the IDEA-encrypted message

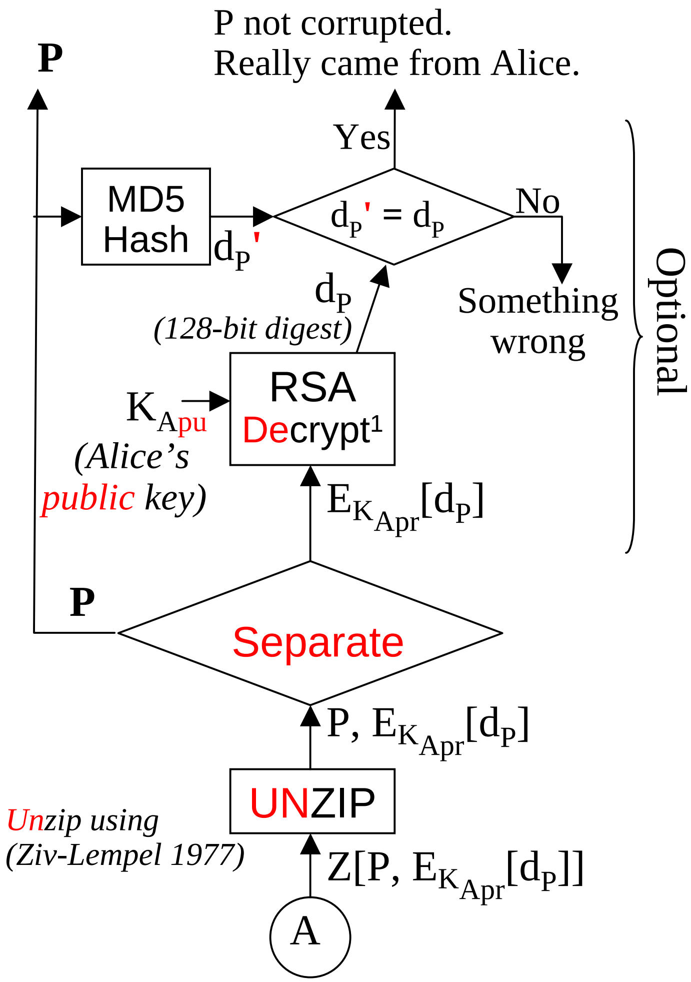
$E_{K_{Bpu}}[K_M]$ $\quad I_{K_M}\{Z[P, E_{K_{Apr}}[d_P]]\}$

(1) Base64 decodes



[1]*RSA is slow but in PGP it only encrypts 128 bits*

PGP Decoding
What Bob Does



(10) If $d_P' = d_P$ then

P is correct

It has **integrity**

P really came from Alice

It's **authentic**

(9) Take MD5 hash of P to get d_P'

(8) RSA decrypt the digest d_P (optional)

$D_{K_{Apu}}[E_{K_{Apr}}[d_P]] = d_P$

(7) Separate

P $E_{K_{Apr}}[d_P]$

(6) Unzip $Z[P, E_{K_{Apr}}[d_P]]$

Result: $P, E_{K_{Apr}}[d_P]$

Comments on PGP

RSA is slow and computationally demanding, but it is used only to encode two 128-bit values, d_P and K_M

IDEA encrypts the much longer string, P

- *IDEA is fast*

There are 3 (or maybe 4) RSA key lengths

- *384 bits*
  - Can be broken today with very fast supercomputers
- *512 bits*
  - Today can be broken by organizations such as NSA in a couple of months
- *1024 bits*
  - Probably breakable in a few months
  - Should be depreciated for long term confidentiality needs
- *2048 bits*
  - Not sure whether it's supported or not

Assign08a

Study for midterm exam.