CSc 466/566

Computer Security

12 : Cryptography — Protocols I

Version: 2019/10/09 11:22:40

Department of Computer Science
University of Arizona

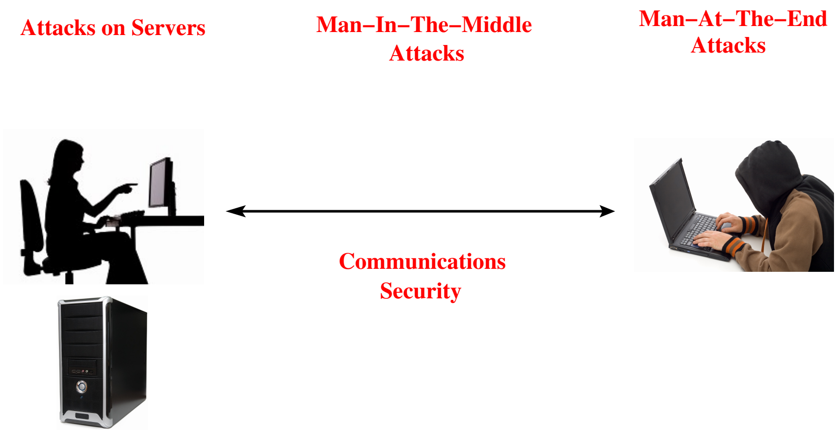
collberg@gmail.com

Christian Collberg

Outline

1. Terminology
2. Security Principles
3. Symmetric Key Protocol
4. Public Key Protocol
5. A Hybrid Protocol
6. GPG
7. Summary

Communications Security



Attacks on Servers

Man–In–The–Middle Attacks

Man–At–The–End Attacks

Communications Security

Cryptographic Concepts

- Cryptography underlies many of the technical means for enforcing security policies.
- Traditionally, encryption is modeled as two parties Alice and Bob who are communicating over an insecure link. An eavesdropper, Eve, is listening in to their communication.

Terminology: Cleartext and Ciphertext

Definition (plaintext or cleartext)

A message we want to transfer securely.

Definition (ciphertext)

The encrypted form of the plaintext message.

Terminology: Encryption and Decryption

Definition (encryption)

Disguising a message to hide its contents.

Definition (encipher or encrypt)

Converting the plaintext to the ciphertext.

Definition (decipher or decrypt)

Converting the ciphertext to the plaintext.

Terminology: Encoding and Decoding

Definition (encode)

Converting the plaintext into a standard alphabet.

Definition (decode)

Converting the encoded message back into the plaintext.

Terminology: Encoding and Decoding...

- Example: uuencode converts a binary file into ASCII text.

```
> echo hello | uuencode -m -o outfile
   remotefile
> cat outfile
begin-base64 644 remotefile
aGVsbG8K
====
> uudecode -p outfile
hello
```

Terminology: Ciphers

Definition (cipher or cypher)

A map from the space of the plaintext to the space of the ciphertext.

Definition (stream cipher)

A cipher that enciphers the plaintext one character at a time.

Definition (block cipher)

A cipher that enciphers the plaintext in chunks of characters.

Terminology: Ciphers. . .

- A cipher is thus an algorithm for encryption/encipherment.
- Ciphers are often named as acronyms:
  - RSA (Rivest-Shamir-Adleman): a public key cipher.
  - DES (Data Encryption Standard): an obsolete block cipher.
  - AES (Advanced Encryption Standard): a current block cipher.
  - RC4 (Rivest Cipher 4): a stream cipher.
  - TEA (Tiny Encryption Algorithm): a block cipher.

Mathematical Notation

P:	plaintext	M:	plaintext (message)
C:	ciphertext.		
E:	encryption function	D:	decryption function.

- The encryption/decryption process:

$$\begin{aligned} E(M) &= C \\ D(C) &= M \\ D(E(M)) &= M \end{aligned}$$

Mathematical Notation. . .

$$\begin{aligned} E(M) &= C \\ D(C) &= M \\ D(E(M)) &= M \end{aligned}$$

- It should be safe to transmit C over an insecure channel.
- The ciphers are chosen such that it is infeasible for anyone but Alice and Bob to find M given C.

Keys

- Ciphers need some sort of secret information known only to Alice and Bob.

K: The key, used by the encryption and decryption functions.

$$\begin{aligned} E_K(M) &= C \\ D_K(C) &= M \\ D_K(E_K(M)) &= M \end{aligned}$$

Definition (keyspace)

The range of possible values of the key.

Example: DES Cipher

```
> cat > message
Attack at dawn^D

> openssl enc -e -des -in message -a \
     -out message.asc -pass pass:Alice4Evah
> cat message.asc
U2FsdGVkX18yv1HS/4te51YLAPk3/ciUuRXw6qU4T08=

> openssl enc -d -des -in message.asc \
     -a -pass pass:Alice4Evah
Attack at dawn
```

Definition (Symmetric-key Algorithms)

Symmetric-key cryptographic algorithms use identical keys for encryption and decryption.

Definition (Public-key Algorithms)

Public-key cryptographic algorithms use different keys for encryption and decryption.

$$\begin{aligned} E_K(M) &= C \\ D_K(C) &= M \\ D_K(E_K(M)) &= M \end{aligned} \qquad \begin{aligned} E_{K_1}(M) &= C \\ D_{K_2}(C) &= M \\ D_{K_2}(E_{K_1}(M)) &= M \end{aligned}$$

Cryptosystems

- To be able to communicate using ciphers we need
  1. Set of possible plaintexts
  2. Set of possible ciphertexts
  3. Set of encryption keys
  4. Set of decryption keys
  5. Correspondence between encryption and decryption keys
  6. Encryption algorithm
  7. Decryption algorithm
- This is known as a cryptosystem.

Example: Caesar Cipher

- Add 3 to the ASCII value of each character, mod 26:

$$A \to D, B \to E, X \to A, \ldots .$$

- Cryptosystem:
  - Set of possible plaintexts and ciphertexts: Latin alphabet
  - Set of encryption keys = {3}
  - Set of decryption keys = {-3}
  - Decryption key = - Encryption key
  - Encryption algorithm = Decryption algorithm = $(x + \mathrm{key})$ mod 26.

Example: ROT13

- Unix utility used on Usenet. Adds 13 mod 26 to each letter.

$$P = \mathtt{ROT13}(\mathtt{ROT13}(P)).$$

```
> echo "hello" | tr 'A-Za-z' 'N-ZA-Mn-za-m'
uryyb
> echo "uryyb" | tr 'A-Za-z' 'N-ZA-Mn-za-m'
hello
```

Outline

1. Terminology
2. Security Principles
3. Symmetric Key Protocol
4. Public Key Protocol
5. A Hybrid Protocol
6. GPG
7. Summary

Open design

Definition (Open design)

The security architecture and design of a system should be made publically available.

- Only cryptographic keys should be kept secret!
- Open design: Allows multiple parties to examine a system for vulnerabilities.
- Open implementation (open source): Anyone can find and fix bugs.
- Opposite: security-through-obscurity.

Open design: Examples

1. Cryptographic algorithms which are safe only if kept secret — once broken, hard to update! Keys are easier to replace if compromised.

Open design: Kerckhoffs Principles I

Auguste Kerckhoffs, *La Cryptographie Militaire*, 1883:

1. The system must be practically, if not mathematically, indecipherable;
2. It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience;
3. Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents;
4. It must be applicable to telegraphic correspondence;
5. It must be portable, and its usage and function must not require the concourse of several people;
6. Finally, it is necessary, given the circumstances that command its application, that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe.

Psychological acceptability

Definition (Psychological acceptability)

User interfaces should be intuitive and security settings should be set to what a user might reasonably expect.

- Examples:
  1. Why don't we always encrypt all email? Apparently it is difficult to design intuitive interfaces.
  2. If a security mechanism makes a system harder to use, then users may turn it off.

Work factor

Definition (Work factor)

The cost of circumventing a security mechanism should be compared to the resources available to the attacker.

- Hard to determine work factor if the attacker can get help from automating the attack.

Work factor: Examples

1. Protecting student grades: most students probably aren't very accomplished hackers.
2. Protecting military secrets: the adversary is a nation state with unlimited resources.
3. Brute force password cracking: now feasible with more powerful computing systems.

Exercise — Goodrich & Tamassia R-1.16

- Give an example how someone might use security-by-obscurity in the design of a system and what the consequences could be.

Outline

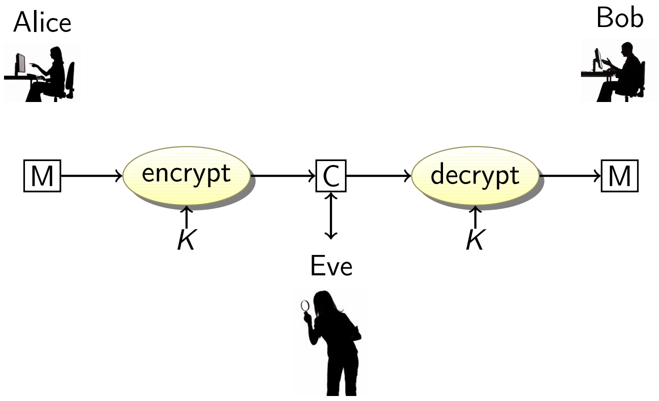
1. Terminology
2. Security Principles
3. Symmetric Key Protocol
4. Public Key Protocol
5. A Hybrid Protocol
6. GPG
7. Summary

Cryptographic Protocol

- Assume that we have a symmetric-key cryptosystem (such as DES).
- How do we use it?
- We need a cryptographic protocol!
- The protocol describes how each party uses the cryptosystem to solve a communication/security problem.

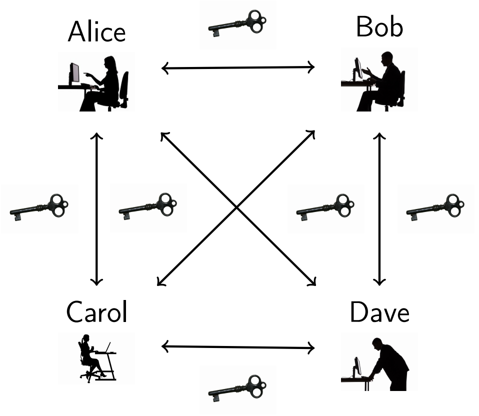
Symmetric-key Encryption Protocol

1. Alice and Bob agree on a cryptosystem.
2. Alice and Bob agree on a key.
3. Alice encrypts her plaintext, getting a ciphertext.
4. Alice sends the ciphertext to Bob.
5. Bob decrypts the message using the same cryptosystem and key.



- Advantages: Ciphers (DES,AES,...) are fast; keys are small.

Symmetric Encryption: Key Distribution



- Disadvantages: $n\frac{n-1}{2}$ keys to communicate between n parties.

Symmetric Encryption Protocol – Attacks

- ciphertext-only attack: Eve listens in on the communication between Alice and Bob. She will get a sequence of ciphertext messages and uses these to launch a ciphertext-only attack.
- evesdrop in key-exchange: Eve listens in to the first two parts of the protocol, where Alice and Bob decide on a key and cryptosystem to use.
- Man-In-The-Middle attack: Eve sits in the middle, intercepts Alice's messages, substitutes her own messages encrypted with the key she has discovered.

Exercise — Goodrich & Tamassia C-1.17

- Alice and Bob want to verify they have the same secret n-bit key K. They engage in the following protocol:
  1. Alice generates a random n-bit value R.
  2. Alice sends $X \leftarrow K_A \oplus R$ to Bob ($\oplus$ = exclusive-or).
  3. Bob sends $Y \leftarrow K_B \oplus X$ to Alice.
  4. Alice compares R and Y. If $R = Y$, she concludes that $K_A = K_B$.
- How can Eve recover the keys?

Exercise — Goodrich & Tamassia C-1.17...

✎

Outline

1. Terminology
2. Security Principles
3. Symmetric Key Protocol
4. Public Key Protocol
5. A Hybrid Protocol
6. GPG
7. Summary

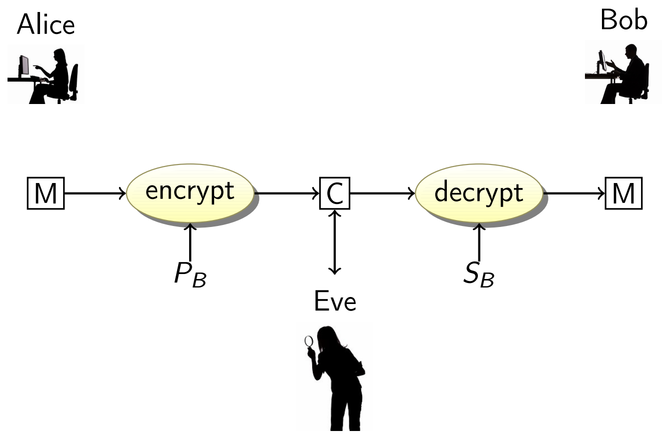
Public Key Protocol

- Key-management is the main problem with symmetric algorithms – Bob and Alice have to somehow agree on a key to use.
- In public key cryptosystems there are two keys, a public one used for encryption and and private one for decryption.
- Bob's public key: P_B
- Bob's secret/private key: S_B

Public Key Protocol...

$$
\begin{aligned}
E_{P_B}(M) &= C\\
D_{S_B}(C) &= M\\
D_{S_B}(E_{P_B}(M)) &= M
\end{aligned}
$$

1. Alice and Bob agree on a public key cryptosystem.
2. Bob sends Alice P_B, or Alice gets it from a public database.
3. Alice encrypts her plaintext using P_B and sends it to Bob.
4. Bob decrypts the message using S_B.



Alice

Bob

M

encrypt

C

decrypt

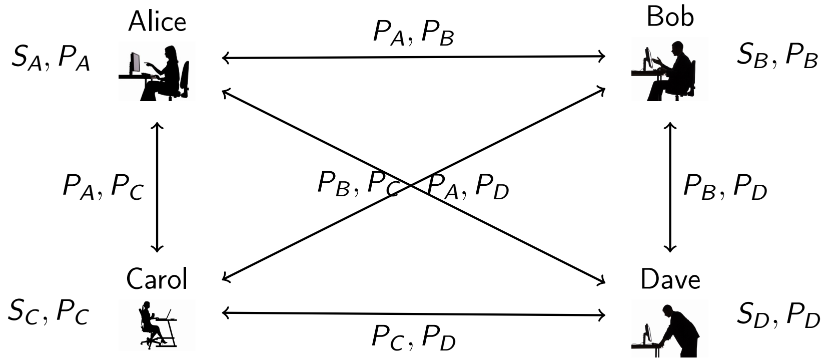
M

P_B

Eve

S_B

Public Key Encryption: Key Distribution



S_A, P_A Alice

Bob S_B, P_B

P_A, P_B

P_A, P_C

P_B, P_C

P_A, P_D

P_B, P_D

S_C, P_C Carol

Dave S_D, P_D

P_C, P_D

- Advantages: n key pairs for n parties.
- Disadvantages: Ciphers are slow; keys are large

Exercise — Goodrich & Tamassia R-1.12

What are the strengths and weaknesses of symmetric-key encryption and public-key encryption?

- Symmetric key:
- Public key:

Exercise — Goodrich & Tamassia C-1.10

- Bob is Alice's bookie. They are communicating using public key cryptography.
- Bob send messages of the form E_{P_A}(3rd race @ saratoga?).
- Alice responds with a message of the form E_{P_B}(\$100 on Golden Mane).
- Eve knows P_A and P_B, the form of the messages, that Alice only bets in multiples of \$100 and never more than \$1000, and all the races and all the horses at all the race tracks (easy to get via a web search).
- How can Eve learn what Alice is betting?

Exercise — Goodrich & Tamassia C-1.10...

✎

Exercise — Goodrich & Tamassia C-1.11

- Can you think of a way to prevent Eve in the previous exercise from learning the contents of the communication?

Exercise — Goodrich & Tamassia C-1.11...

✎

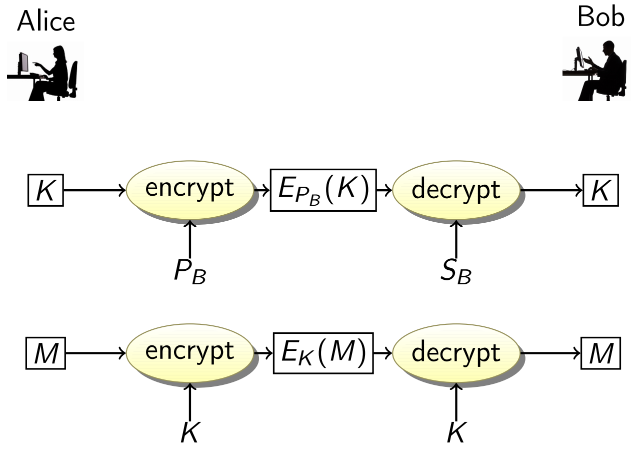
Outline

1. Terminology
2. Security Principles
3. Symmetric Key Protocol
4. Public Key Protocol
5. A Hybrid Protocol
6. GPG
7. Summary

A Hybrid Protocol

- In practice, public key cryptosystems are not used to encrypt messages – they are simply too slow.
- Instead, public key cryptosystems are used to encrypt keys for symmetric cryptosystems — session keys.
- Session keys are discarded once the communication session is over.

1. Alice:
   1. Gets Bob's public key P_B.
   2. Generates a session key K.
   3. Encrypts K with Bob's public key.
   4. Sends $E_{P_B}(K)$ to Bob.
2. Bob:
   1. Decrypts the message using his private key S_B to get the session key K.
3. Alice and Bob:
   1. communicate by encrypting their messages using a symmetric cipher and K.



Outline

1. Terminology
2. Security Principles
3. Symmetric Key Protocol
4. Public Key Protocol
5. A Hybrid Protocol
6. GPG
7. Summary

Software – GPG

- gpg is a public domain implementation of pgp.
- Supported algorithms:

Pubkey:	RSA, RSA-E, RSA-S, ELG-E, DSA
Cipher:	3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH, CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash:	MD5, SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression:	Uncompressed, ZIP, ZLIB, BZIP2

- http://www.gnupg.org.

```
> sudo port install gnupg2
> gpg --gen-key
> gpg --armor --export Bobby
> cat message
Attack at dawn
> gpg --recipient bobby --armor --encrypt message
> cat message.asc
> gpg --decrypt message.asc
> gpg --list-keys
> gpg --list-secret-keys
> gpg --cipher-algo=AES --armor --symmetric message
> gpg --armour --gen-random 0 100
```

Goal: Read a message encrypted with gpg

- Draw the attack tree for reading a message encrypted with gpg.

1. Decrypt the message itself **OR**
2. Determine symmetric key used to encrypt the message by other means **OR**
3. Get recipient to help decrypt message **OR**
4. Obtain private key of recipient.

http://www.schneier.com/paper-attacktrees-fig7.html

A:

Decrypt the message itself:

1. Break asymmetric encryption **OR**
    1. Brute force break asymmetric encryption **OR**
    2. Mathematically break asymmetric encryption **OR**
        1. Break RSA **OR**
        2. Factor RSA modulus/calculate Elgamal discrete log
    3. Cryptanalyze asymmetric encryption **OR**
        1. General cryptanalysis of RSA/Elgamal **OR**
        2. Exploit weakness in RSA/Elgamal **OR**
        3. Timing attack on RSA/Elgamal
2. Break symmetric-key encryption
    1. Brute force break symmetric-key encryption
    2. Cryptanalysis of symmetric-key encryption

B:

Determine symmetric key by other means:

1. Fool sender into encrypting message using public key whose private key is known **OR**
    1. Convince sender that fake key (with known private key) is the key of the intended recipient
    2. Convince sender to encrypt with more than one key—the real key of the recipient and a key whose private key is known.
    3. Have the message encrypted with a different public key in the background, unbeknownst to the sender.
2. Have the recipient sign the encrypted publc key **OR**
3. Monitor the sender's computer memory **OR**
4. . . .

B:

Determine symmetric key by other means:

1. . . .
2. . . .
3. . . .
4. Monitor the receiver's computer memory **OR**
5. Determine key from pseudo-random number generator **OR**
    1. Determine state of randseed during encryption **OR**
    2. Implant virus that alters the state of randseed. **OR**
    3. Implant software that affects the choice of symmetric key.
6. Implant virus that that exposes public key.

C:

Get recipient to help decrypt message:

1. . . .

D:

Obtain private key of recipient:

1. . . .

What immediately becomes apparent from the attack tree is that breaking the RSA or IDEA encryption algorithms are not the most profitable attacks against PGP. There are many ways to read someone's PGP-encrypted messages without breaking the cryptography. You can capture their screen when they decrypt and read the messages (using a Trojan horse like Back Orifice, a TEMPEST receiver, or a secret camera), grab their private key after they enter a passphrase (Back Orifice again, or a dedicated computer virus), recover their passphrase (a keyboard sniffer, TEMPEST receiver, or Back Orifice), or simply try to brute force their passphrase (I can assure you that it will have much less entropy than the 128-bit IDEA keys that it generates).

In the scheme of things, the choice of algorithm and the key length is probably the least important thing that affects PGP's overall security. PGP not only has to be secure, but it has to be used in an environment that leverages that security without creating any new insecurities.

http://www.schneier.com/paper-attacktrees-fig7.html

Outline

1. Terminology
2. Security Principles
3. Symmetric Key Protocol
4. Public Key Protocol
5. A Hybrid Protocol
6. GPG
7. **Summary**

Readings

- Chapter 1 in *Introduction to Computer Security*, by Goodrich and Tamassia.

Acknowledgments

Material and exercises have also been collected from these sources:

1. Bishop, *Introduction to Computer Security*.