CAN304
Computer Systems Security

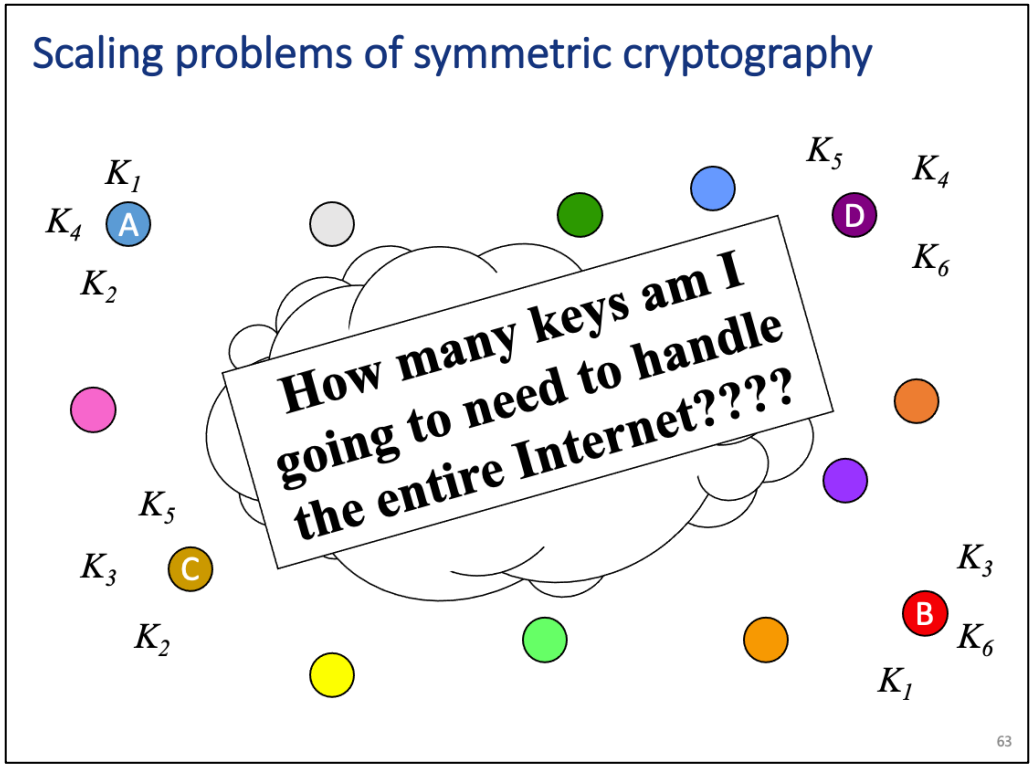
Lecture 4. Fundamentals of cryptography (3)

Week 4: 2022-03-18, 14:00-16:00, Friday

Jie Zhang
Department of Communications and Networking
Email: jie.zhang01@xjtlu.edu.cn
Office: EE522

Review

- Symmetric encryption
- Message authentication code



Outline

- The public-key revolution
- Public-key encryption
- Digital signature
- Diffie-Hellman key agreement

Learning objetives

- Understand and be able to apply the public-key primitives.

- **NB**
  - You NEED NOT be able to explain the design of the algorithms or realize the algorithms
    - crypto libraries

1. The public-key revolution

Private-key cryptography

- Private-key cryptography allows two users who share a secret key to establish a “secure channel”
- The need to share a secret key incurs several drawbacks...

The key-distribution problem

- How do users share a key in the first place?
  - Need to share the key using a secure channel...

- This problem can be solved in some settings...
  - E.g., physical proximity, trusted courier
  - inconvenient, expensive

The key-management problem

- Imagine an organization with N employees, where each pair of employees might need to communicate securely

- Solution using private-key cryptography:
  - Each user shares a key with all other users
  - ⇒ Each user must store/manage N-1 secret keys!

Lack of support for “open systems”

- Say two users who have no prior relationship want to communicate securely
  - When would they ever have shared a key?

- This is not at all far-fetched!
  - Sending an email to a professor in another university
  - Sending application form to a university

Symmetric cryptography offers no solution to these problems!

New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

***Abstract*—Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.**

I. Introduction

WE STAND TODAY on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade cryptographic devices down to where they can be used in such commercial applications as remote cash dispensers and computer terminals. In turn, such applications create a need for new types of cryptographic systems which minimize the necessity of secure key distribution channels and supply the equivalent of a written signature. At the same time, theoretical developments in information theory and computer science show promise of providing provably secure cryptosystems, changing this ancient art into a science.

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel. In order to use cryptography to insure privacy, however, it is currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such as private courier or registered mail. A private conversation between two people with no prior acquaintance is a common occurrence in business, however, and it is unrealistic to expect initial business contacts to be postponed long enough for keys to be transmitted by some physical means. The cost and delay imposed by this key distribution problem is a major barrier to the transfer of business communications to large teleprocessing networks.

Section III proposes two approaches to transmitting keying information over public (i.e., insecure) channels without compromising the security of the system. In a *public key cryptosystem* enciphering and deciphering are governed by distinct keys, E and D, such that computing D from E is computationally infeasible (e.g., requiring 10^{100} instructions). The enciphering key E can thus be publicly disclosed without compromising the deciphering key D. Each user of the network can, therefore, place his enciphering key in a public directory. This enables any user of the system to send a message to any other user enci-

New directions...

- Key ideas:
  - Some problems exhibit asymmetry
    - Easy to compute, but hard to invert
  - Use this asymmetry to enable two parties to agree on a shared secret key using public discussion

The public-key setting

- A party generates a pair of keys: a public key pk and a private key sk
  - Public key is widely disseminated
  - Private key is kept secret, and shared with no one

- Private key used by this party; public key used by everyone else

- Also called asymmetric cryptography

Public key distribution

- Assume parties are able to obtain correct copies of each others' public keys
  - i.e., the attacker is passive, at least during key distribution
- We will revisit this assumption later.

Easy and hard

- Some problems exhibit asymmetry
  - Easy to compute, but hard to invert

- Easy:
  - Can be computed efficiently
  - There exist efficient algorithms
  - E.g., integer addition, integer multiplication
- Hard:
  - There doesn’t exist efficient algorithm by now.

Factoring problem

- Multiplying two numbers is easy; factoring a number is hard in some settings.
  - Given x, y, easy to compute $x \cdot y$
  - Given xy, hard to find x and y

- Compare:
  - Multiply 10101023 and 29100257
  - Find the factors of 293942365262911

Factoring problem

- It’s not hard to factor all numbers
  - 50% of the time, random number is even
  - 1/3 of the time, random number is divisible by 3...

- The hardest numbers to factor are those that are the product of two, equal-length primes

Discrete-logarithm problem

- Fix cyclic group G of order m, and generator g

- Dlog problem in G:
  - Given g and an element h in G, find x such that $g^x = h$
- Dlog assumption in G:
  - Solving the discrete log problem in G is hard

Discrete-logarithm problem

- Fix cyclic group G of order m, and generator g

- Group
  - In mathematical context, a *group* is a set of elements (typically, numbers) that are related to each other according to certain well-defined rules.
  - Example: Z_p^* = the set of nonzero integers (between 1 and $p - 1$) modulo some prime number p
  - $Z_5^* = \{1,2,3,4\}$

- When G has a finite number of elements, we say G is finite
- The **order** of a finite group G is the number of elements in G

Discrete-logarithm problem

- Fix cyclic group G of order m, and generator g

- Cyclic group
  - $G = \{g^0, g^1, \ldots, g^{m-1}\}$
  - g is a generator of G

Diffie-Hellman problems

- Fix group G with generator g

- Computational Diffie-Hellman (CDH) problem:
  - Given g, g^x, g^y, compute g^{xy}
- Decisional Diffie-Hellman (DDH) problem:
  - Given g, g^x, g^y, distinguish the correct g^{xy} from a uniform element of G

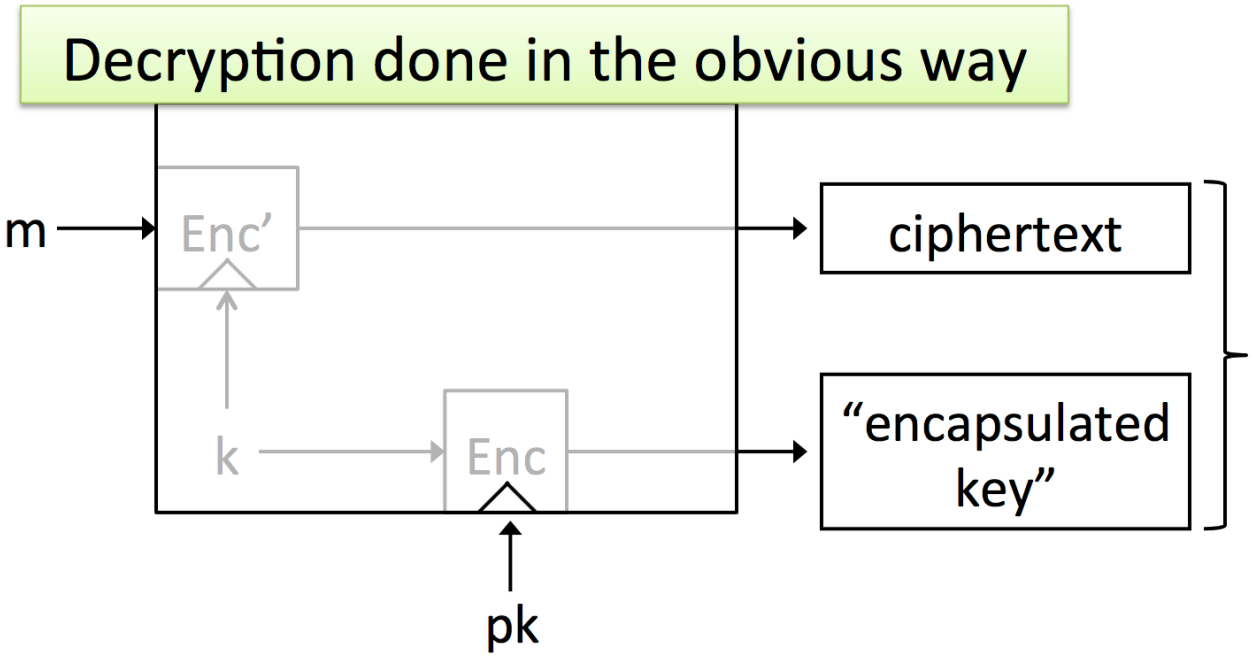
2. Public-key encryption

Public-key encryption

- A public-key encryption scheme is composed of three algorithms:
  - Gen: key-generation algorithm that on input 1^n outputs pk, sk
  - Enc: encryption algorithm that on input pk and a message m outputs a ciphertext c
  - Dec: decryption algorithm that on input sk and a ciphertext c outputs message m or $\perp$

For all m and pk, sk output by Gen,
$Dec_{sk}(Enc_{pk}(m)) = m$

Hybrid encryption



- The functionality of public-key encryption at the (asymptotic) efficiency of private-key encryption!

Dlog-based PKE: ElGamal encryption

- Gen
  - Initialize group prameters G, q, g. Choose uniform $x \in Z_q$, compute $h = g^x$.
  - Public key is h, private key is x

- $Enc_{pk}(m)$, where $m \in G$
  - Choose uniform $y \in Z_q$.
  - The ciphertext is $(c_1, c_2) = (g^y, h^y \cdot m)$

- $Dec_{sk}(c_1, c_2)$
  - Output $\frac{c_2}{c_1^x}$

RSA encryption

- $RSAGen(1^n)$
  - Generate uniform n-bit primes p, q
  - Set $N = pq$
  - Choose arbitrary e with $\gcd(e, \phi(N)) = 1$
  - Compute $d = e^{-1} \bmod \phi(N)$
  - Output the public key (N, e) and the private key d
- $Enc_{pk}(m)$
  - Output $c = m^e \bmod N$
- $Dec_{sk}(c)$
  - Output $m = c^d \bmod N$

$\phi(N) = (p-1)(q-1)$

gcd: greatest common divisor

RSA-based PKE: RSA encryption

- "Plain" RSA encryption

c

$(N, e, d) \leftarrow RSAGen(1^n)$

$pk = (N, e)$

$sk = d$

N, e

$c = m^e \bmod N$

$m = c^d \bmod N$

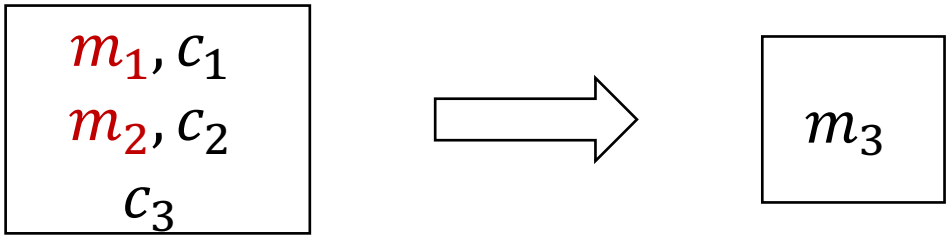
Security?

- Chosen-plaintext attack

Chosen-ciphertext attack
Chosen-plaintext attack
Known-plaintext attack
Ciphertext-only attack

- Adversary power
  - can obtain plaintext-ciphertext pairs for plaintexts of its choice
- Adversary aim
  - deduce the information about the underlying plaintext of some other ciphertext produced using the same key

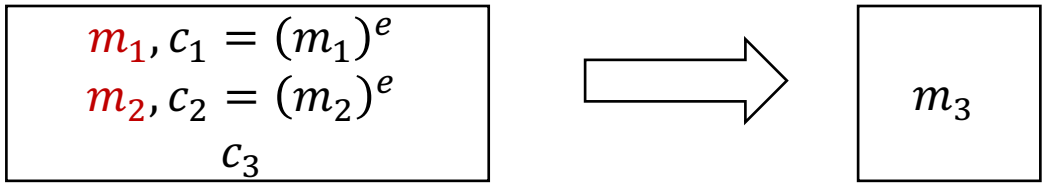
Attacker has pk!



Security?

- In plain RSA

mod N is omitted here



- What if $c_1 \cdot c_2 = c_3$?
- $m_3 = m_1 \cdot m_2$
- Note: we are not showing the real CPA attacks to plain RSA; we are just illustrating that plain RSA doesn't have CPA security!

- Plain RSA is not CPA-secure!

Plain RSA should never be used!

PKCS #1 v1.5

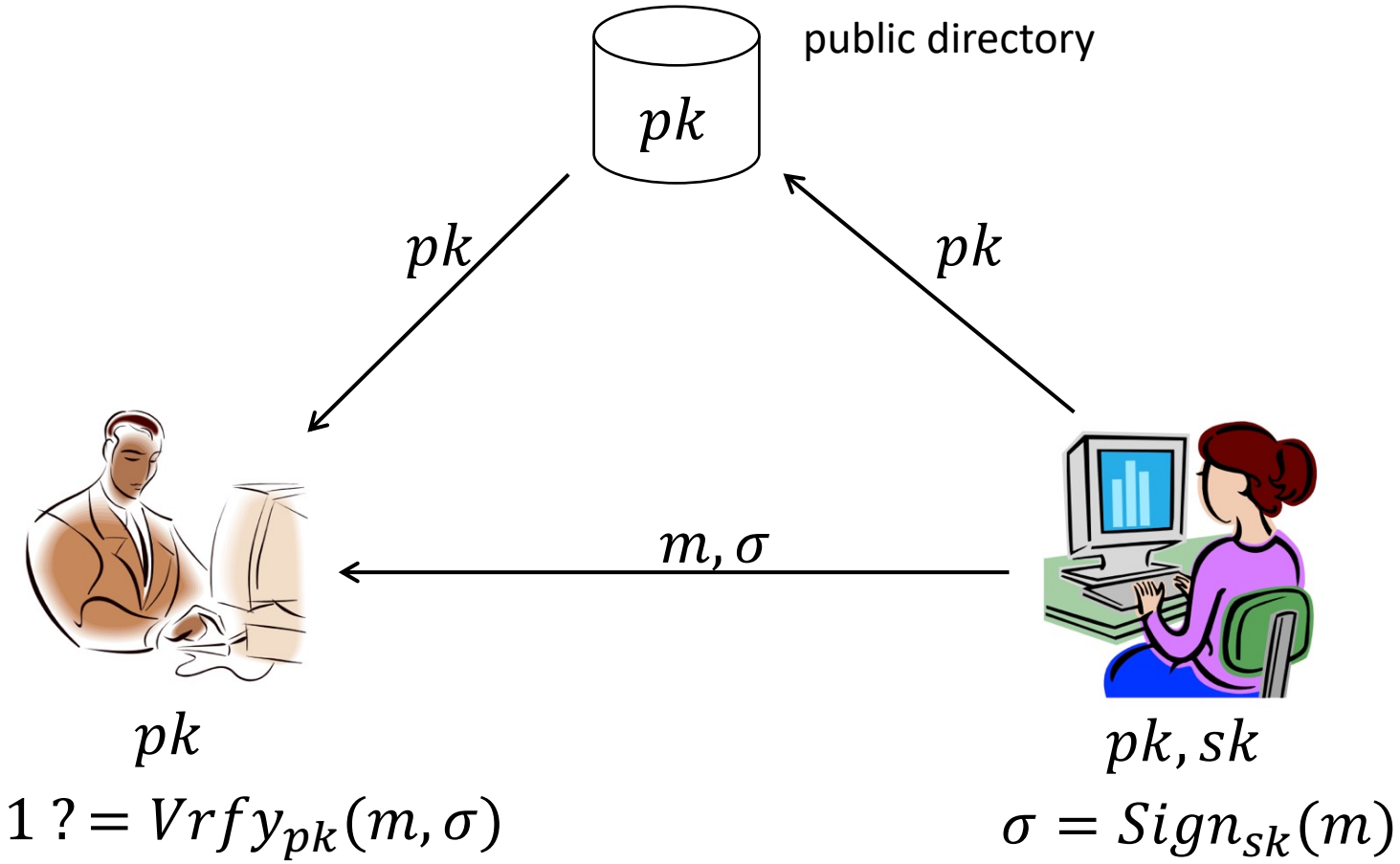
- Idea: add random padding
  - To encrypt m, choose random r
  - $c = [(r|m)^e \bmod N]$

- PKCS: Public-Key Cryptography Standard (PKCS)
- Issues:
  - No proof of CPA-security (unless m is very short)
  - Chosen-plaintext attacks known if r is too short
  - Chosen-ciphertext attacks known
- RSA-OAEP
  - More complex padding
  - Lab2

3. Digital signature

Digital signatures

- Provide integrity in the public-key setting



Comparison to MACs?

- Public verifiability
  - “Anyone” can verify a signature
  - (Only a holder of the key can verify a MAC tag)

- ⇒ Transferability
  - Can forward a signature to someone else...
- ⇒ Non-repudiation

Non-repudiation

- Signer cannot (easily) deny issuing a signature
  - Crucial for legal applications
  - Judge can verify signature using public copy of pk
- MACs cannot provide this functionality!
  - Without access to the key, no way to verify a tag
  - Even if receiver leaks key to judge, how can the judge verify that the key is correct?
    - Even if key is correct, receiver could have generated the tag also!

Signature schemes

PPT: Probabilistic Polynomial Time

- A signature scheme is defined by three PPT algorithms (Gen, Sign, Vrfy):
  - Gen: takes as input 1^n; outputs pk, sk
  - Sign: takes as input a private key sk and a message $m \in \{0,1\}^*$; outputs signature σ

$$\sigma \leftarrow \mathrm{Sign}_{sk}(m)$$

  - Vrfy: takes public key pk, message m, and signature σ as input; outputs 1 or 0

For all m and all pk, sk output by Gen,
$\mathrm{Vrfy}_{pk}(m, \mathrm{Sign}_{sk}(m)) = 1$

Hash-and-sign paradigm

- Given
  - A signature scheme Π=(Gen, Sign, Vrfy) for "short" messages of length n
  - Hash function H: $\{0,1\}^* \rightarrow \{0,1\}^n$

- Construct a signature scheme Π'=(Gen, Sign', Vrfy') for arbitrary-length messages:
  - $\mathrm{Sign}'_{\mathrm{sk}}(m) = \mathrm{Sign}_{\mathrm{sk}}(H(m))$
  - $\mathrm{Vrfy}'_{\mathrm{pk}}(m, \sigma) = \mathrm{Vrfy}_{\mathrm{pk}}(H(m), \sigma)$

- Used extensively in practice

RSA-based signatures

- "Plain" RSA signatures

m, σ

N, e

$m? = \sigma^e \bmod N$

$(N, e, d) \leftarrow RSAGen(1^n)$

$pk = (N, e)$

$sk = d$

$\sigma = m^d \bmod N$

Attacks

- Can sign specific messages
  - E.g., $m = 1, \sigma = [m^d \; mod \; N]$

- Can sign "random" messages
  - Choose arbitrary σ; set $m = [\sigma^e \bmod N]$

- Can combine two signatures to obtain a third
  - Say σ_1, σ_2 are valid signatures on m_1, m_2 with respect to public key N, e
  - Then $\sigma' = \sigma_1\sigma_2 \bmod N$ is a valid signature on the message $m' = m_1 m_2 \bmod N$

$$(\sigma_1\sigma_2)^e = \sigma_1^e\sigma_2^e = m_1 m_2 \bmod N$$

RSA-FDH

- RSA-FDH: RSA full-domain hash

- Public key: (N, e); private key: d
- $\text{Sign}_{\text{sk}}(m) = H(m)^d \bmod N$
- $\text{Vrfy}_{\text{pk}}(m, \sigma)$: output 1 iff

$$\sigma^e = H(m) \bmod N$$

- (This also handles long messages)

Signatures from the discrete-logarithm problem

- Schnorr signature scheme

- DSA and ECDSA
  - included in the current Digital Signature Standard (DSS) issued by NIST

P451-461, *Introduction to Modern Cryptography*

4. Diffie-Hellman key agreement

New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

***Abstract*—Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.**

I. Introduction

WE STAND TODAY on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade cryptographic devices down to where they can be used in such commercial applications as remote cash dispensers and computer terminals. In turn, such applications create a need for new types of cryptographic systems which minimize the necessity of secure key distribution channels and supply the equivalent of a written signature. At the same time, theoretical developments in information theory and computer science show promise of providing provably secure cryptosystems, changing this ancient art into a science.

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel. In order to use cryptography to insure privacy, however, it is currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such as private courier or registered mail. A private conversation between two people with no prior acquaintance is a common occurrence in business, however, and it is unrealistic to expect initial business contacts to be postponed long enough for keys to be transmitted by some physical means. The cost and delay imposed by this key distribution problem is a major barrier to the transfer of business communications to large teleprocessing networks.

Section III proposes two approaches to transmitting keying information over public (i.e., insecure) channels without compromising the security of the system. In a *public key cryptosystem* enciphering and deciphering are governed by distinct keys, E and D, such that computing D from E is computationally infeasible (e.g., requiring 10^{100} instructions). The enciphering key E can thus be publicly disclosed without compromising the deciphering key D. Each user of the network can, therefore, place his enciphering key in a public directory. This enables any user of the system to send a message to any other user enci-

Diffie-Hellman key agreement

G: cyclic group

q: prime, order of G

g: generator of G

Decisional Diffie-Hellman (DDH) problem:
Given g^x, g^y, distinguish g^{xy} from a uniform group element

Hardness

G, q, g

h_1

h_2

$k_1 = (h_2)^x$

$x \leftarrow Z_q$

$h_1 = g^x$

$k_2 = (h_1)^y$

$y \leftarrow Z_q$

$h_2 = g^y$

Elliptic curve Diffie-Hellman key agreement

- E: elliptic curve group
- q: prime, order of E
- P: generator of E

ECDDH problem: Given P, yP, distinguish xyP from uniform group element

Hardness

E, P, q

h_1

h_2

$k_1 = xh_2$

$x \leftarrow Z_q$

$h_1 = xP$

$k_2 = yh_1$

$y \leftarrow Z_q$

$h_2 = yP$

Man-in-the-middle to EC(DH)

$$E, P, q$$

$k_1 = xh$

$x \leftarrow Z_q$

$h_1 = xP$

$k_3 = ch_1, k_4 = ch_2$

$c \leftarrow Z_q$

$h = cP$

$k_2 = yh$

$y \leftarrow Z_q$

$h_2 = yP$

- Solution:
  - Introduce authentication in the protocol.

Summary

- The public-key revolution
- Public-key encryption
- Digital signature
- Diffie-Hellman key agreement