یا ذاالامن والامان

امنیت داده و شبکه

رمزنگاری نامتقارن (کلید عمومی)

مرتضی امینی - نیمسال اول ۹۰-۸۹

مرکز امنیت داده و شبکه شریف
http://dnsl.ce.sharif.edu

فهرست مطالب

- مبانی رمزنگاری کلید عمومی
- مقایسه با رمزنگاری سنتی و متقارن
- کاربردهای رمزنگاری کلید عمومی
- الگوریتم رمز RSA
- الگوریتم رمز دیفی-هلمن

مبانی رمزنگاری کلید عمومی

- رمزنگاری کلید عمومی اساساً با انگیزه رسیدن به دو هدف طراحی شد:
  - حل مساله توزیع کلید در روشهای رمزنگاری متقارن
  - امضای دیجیتال

  - دیفی و هلمن اولین راه حل را در ۱۹۷۶ ارایه دادند.

رمزنگاری کلید عمومی

- کلید های رمزگذاری و رمزگشایی متفاوت اما مرتبط هستند.
- رسیدن به کلید رمزگشایی از کلید رمزگذاری از لحاظ محاسباتی ناممکن است.
- رمزگذاری امری همگانی است و اساساً نیازی به اشتراک گذاشتن اطلاعات محرمانه ندارد.
- رمزگشایی از طرف دیگر امری اختصاصی بوده و محرمانگی پیام‌ها محفوظ می‌ماند.

نمادها و قراردادها

- **کلید عمومی:** کلید رمزگذاری
  - این کلید را برای شخص A با PU_a نشان می‌دهیم.
- **کلید خصوصی:** کلید رمزگشایی
  - این کلید را برای شخص A با PR_a نشان می‌دهیم.

نیازمندیهای رمزنگاری کلید عمومی

- از نظر محاسباتی برای طرف B، تولید یک زوج کلید (کلید عمومی PU_b و کلید خصوصی PR_b) آسان باشد.
- برای فرستنده، تولید متن رمز آسان باشد:

$$C = E_{PU_b}(M)$$

- برای گیرنده، رمزگشایی متن با استفاده از کلید خصوصی آسان باشد:

$$M = D_{PR_b}(C) = D_{PR_b}[E_{PU_b}(M)]$$

نیازمندیهای رمزنگاری کلید عمومی

- از نظر محاسباتی، تولید کلید خصوصی (PR_b) با دانستن کلید عمومی (PU_b) غیر ممکن باشد.
- بازیابی پیام M، با دانستن PU_b و C غیرممکن باشد.
- ویژگی تقارنی: از هر یک از کلیدها می‌توان برای رمزکردن استفاده کرد. در این صورت از کلید دیگر برای رمزگشایی استفاده می‌شود.

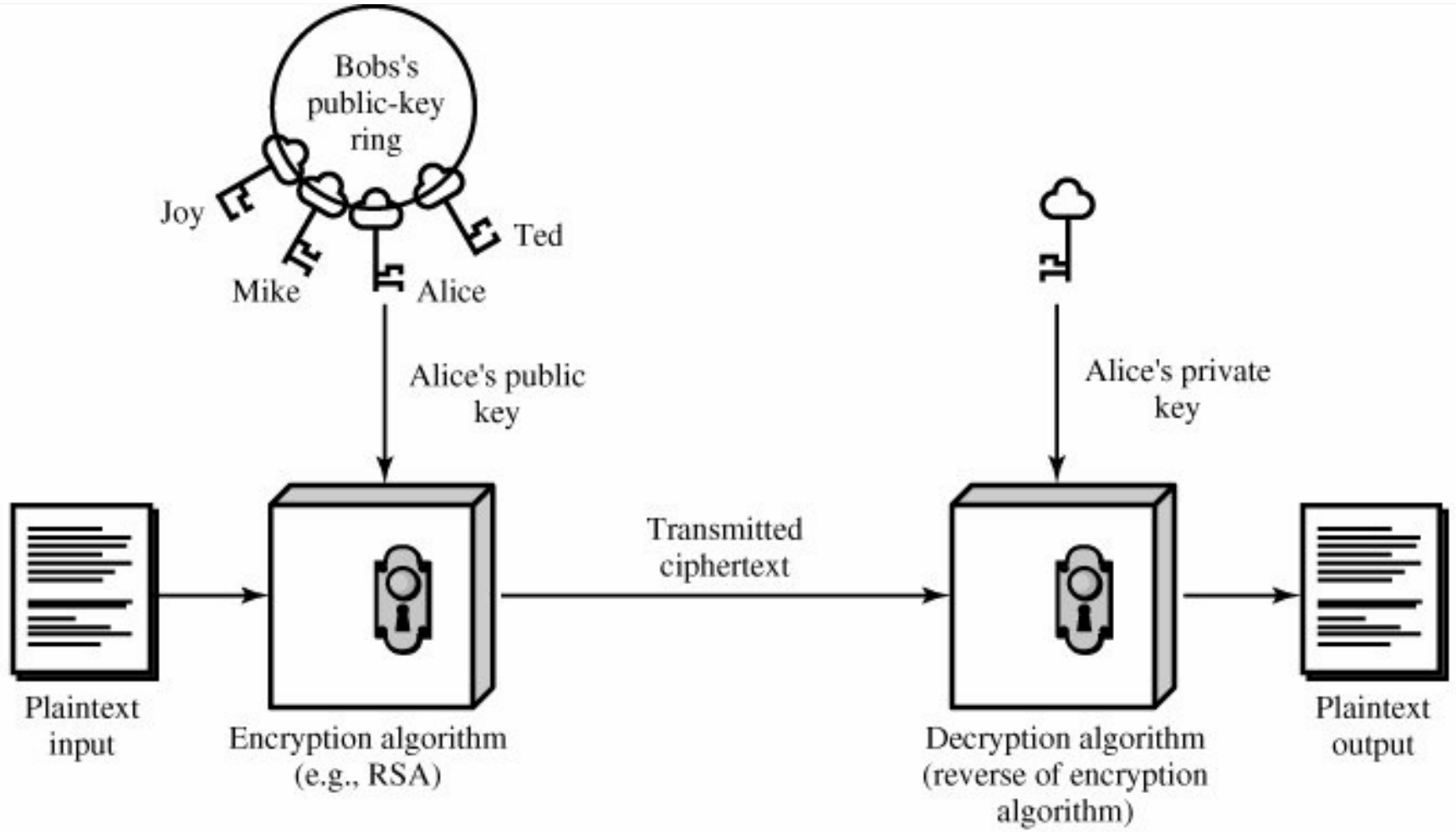
$$M = D_{PR_b}[E_{PU_b}(M)] = D_{PU_b}[E_{PR_b}(M)]$$

رمزگذاری کلید عمومی

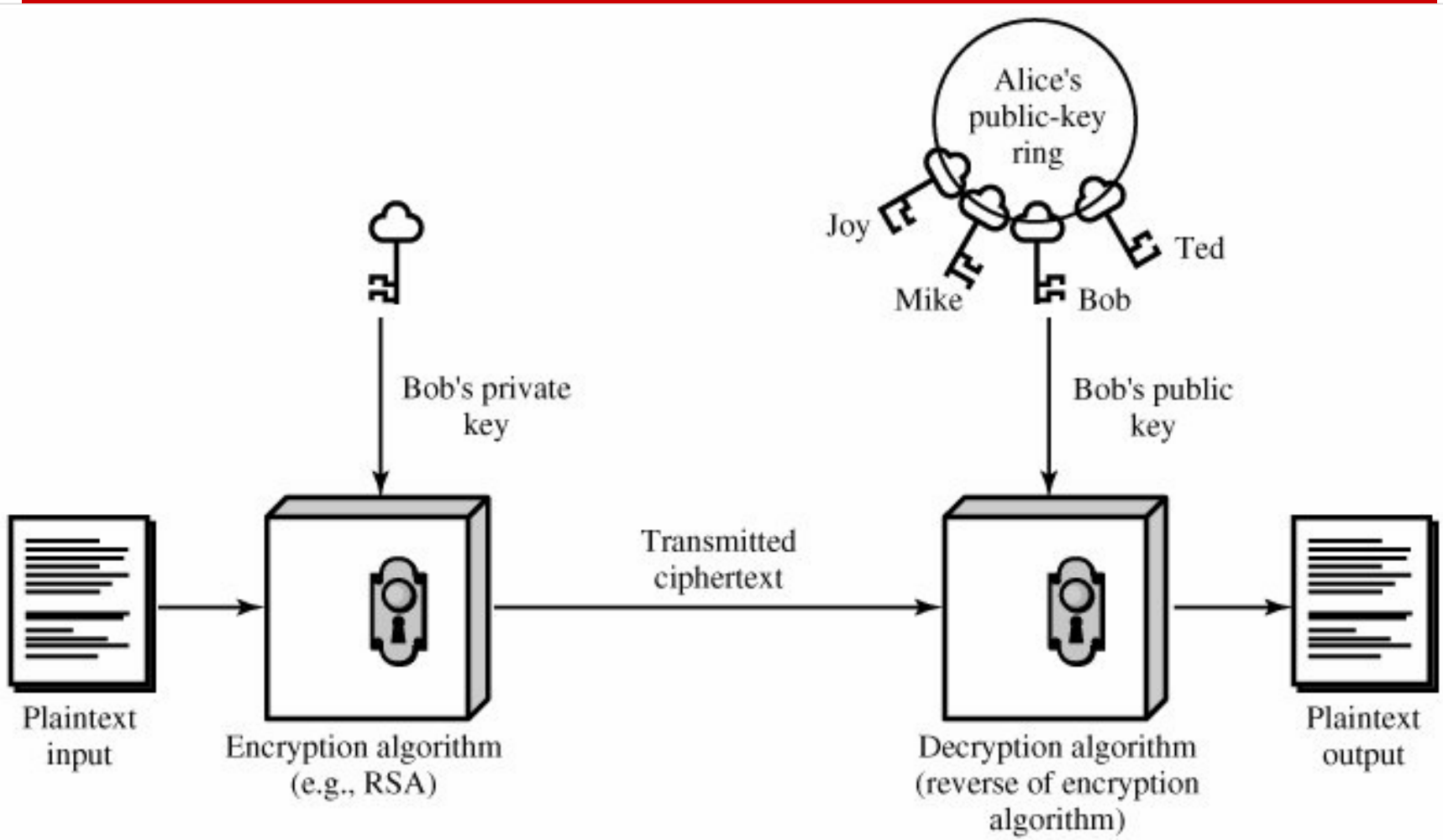
- برای رمز نگاری کلید عمومی گام‌های زیر را برمی‌داریم:

1. هر کاربر یک زوج کلید رمزگذاری و رمزگشایی تولید می‌کند.
2. کاربران کلید رمزگذاری خود را به صورت عمومی اعلان می‌کنند درحالی که کلید رمزگشایی مخفی می‌باشد.
3. همگان قادر به ارسال پیام رمز شده برای هر کاربر دلخواه با استفاده از کلید رمزگذاری (عمومی) او هستند.
4. هر کاربر می‌تواند با کمک کلید رمزگشایی (خصوصی) پیام‌هایی که با کلید رمزگذاری (عمومی) او رمز شده رمزگشایی کند.

رمزگذاری با کلید عمومی



رمزگشایی با کلید عمومی



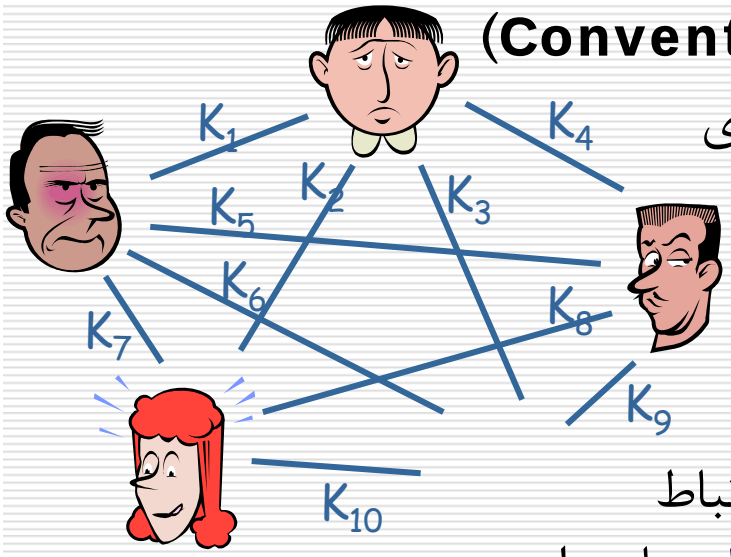
فهرست مطالب

- مبانی رمزنگاری کلید عمومی
- **مقایسه با رمزنگاری سنتی و متقارن**
- کاربردهای رمزنگاری کلید عمومی
- الگوریتم رمز RSA
- الگوریتم رمز دیفی-هلمن

مقایسه رمزنگاری مرسوم و رمزنگاری کلید عمومی

رمزنگاری مرسوم (Conventional Cryptography)

- استفاده از یک کلید یکسان و مخفی برای رمزنگاری



معایب

- مشکل مدیریت کلیدها
  - نیاز به توافق بر روی کلید پیش از برقراری ارتباط
  - برای ارتباط n نفر باهم به n(n-1)/2 کلید احتیاج داریم.
- عدم پشتیبانی از امضاء الکترونیکی

مزایا

- با این وجود از الگوریتم‌های رمزنگاری با کلید عمومی سریع‌تر است.

مقایسه رمزنگاری مرسوم و رمزنگاری کلید عمومی

- در **رمزگذاری مرسوم** برای امن بودن باید:
  - کلید سری، مخفی نگه داشته شود.
  - رسیدن به پیام آشکار از روی متن رمزشده از نظر محاسباتی ناممکن باشد.
  - اطلاع از الگوریتم و داشتن نمونه‌هایی از پیغام رمزشده برای تعیین کلید کافی نباشد.

مقایسه رمزگذاری مرسوم و رمزگذاری کلید عمومی

- ملزومات امنیتی **رمزگذاری با کلید عمومی**
  - تنها یکی از دو کلید باید مخفی بماند.
  - رسیدن به پیام آشکار از روی متن رمز شده حتی با داشتن کلید عمومی از نظر محاسباتی ناممکن باشد.
  - اطلاع از الگوریتم، داشتن یکی از کلیدها و نیز دراختیار داشتن نمونه پیغام‌های رمزشده برای تعیین کلید دوم کافی نباشد.

جایگزینی یا تکمیل؟

از نظر کاربردی، رمزگذاری با کلید عمومی بیش از آنکه **جایگزینی** برای رمزگذاری مرسوم باشد, نقش **مکمل** آن را برای حل مشکلات توزیع کلید بازی می‌کند.

سوء برداشت!

- دو تصور اشتباه دیگر درباره الگوریتم‌های کلید عمومی
  - رمزنگاری با کلید عمومی امن‌تر است!
    - در هر دو روش رمزنگاری امنیت به طول کلید وابسته است.
  - مسئله توزیع کلید در رمزنگاری با کلید عمومی برطرف شده است!
    - چگونه مطمئن شویم کلید عمومی لزوما متعلق به شخص ادعاکننده است؟!
    - پس توزیع کلید عمومی آسانتر است، ولی بدیهی نیست.

محرمانگی و احراز اصالت به صورت همزمان

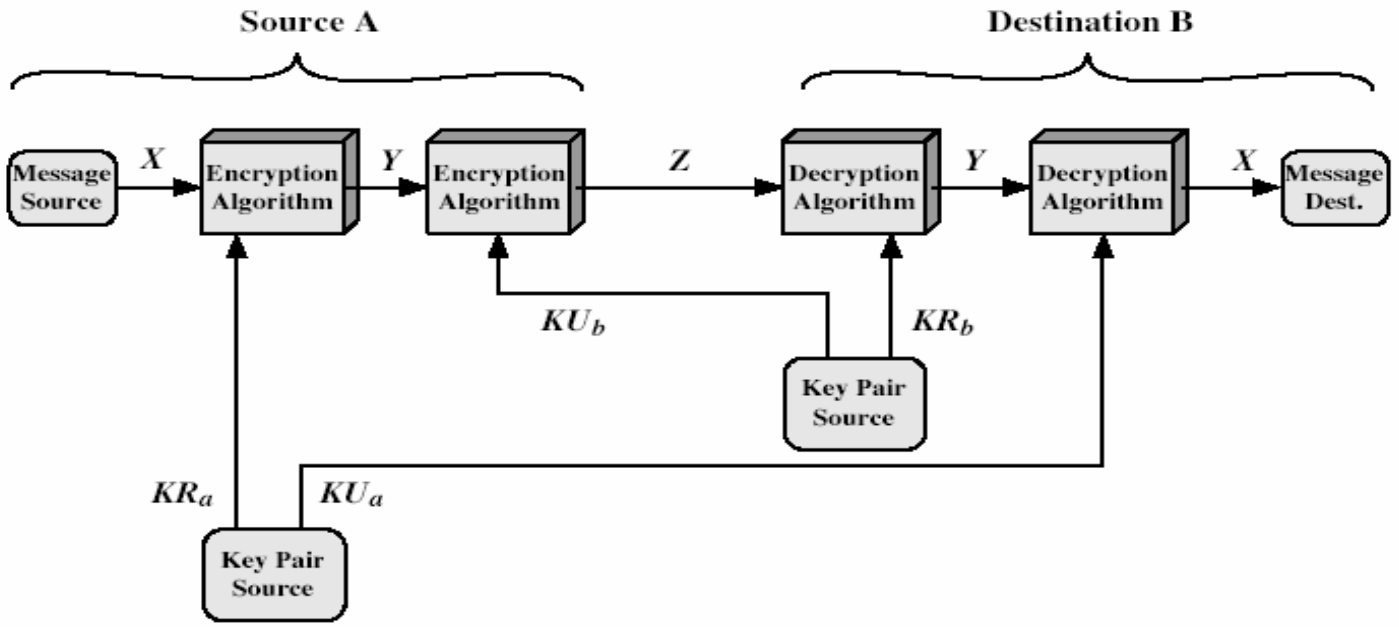


Figure 9.4 Public-Key Cryptosystem: Secrecy and Authentication

رمزگذاری کلید عمومی: محرمانگی و احراز اصالت به صورت همزمان

فهرست مطالب

- مبانی رمزنگاری کلید عمومی
- مقایسه با رمزنگاری سنتی و متقارن
- **کاربردهای رمزنگاری کلید عمومی**
- الگوریتم رمز RSA
- الگوریتم رمز دیفی-هلمن

کاربردهای رمزنگاری کلید عمومی

- **رمزگذاری / رمزگشایی:** برای حفظ محرمانگی
- **امضاء رقمی:** برای حفظ اصالت پیام و معین نمودن فرستنده پیام (پیوند دادن پیام با امضاء کننده)
- **توزیع کلید:** برای توافق طرفین روی کلید مخفی جلسه، قبل از برقراری ارتباط

جایگاه عملی رمزنگاری کلید عمومی

- کلیدهای این نوع از الگوریتم‌ها بسیار طولانی تر از الگوریتم‌های مرسوم (کلید خصوصی) هستند.
  - الگوریتم RSA با پیمانه ۱۰۲۴ بیتی امنیتی در حد الگوریتم‌های متقارن با کلیدهای ۸۰ بیتی دارد.

- سرعت الگوریتم‌های کلید عمومی از الگوریتم‌های رمزگذاری مرسوم پایین‌تر است.
  - RSA تقریباً ۱۰۰۰ بار کندتر از رمزهای کلید سری (با امنیت یکسان) است.

جایگاه عملی رمزنگاری کلید عمومی

- امروزه کاربرد این الگوریتم‌ها به حل مساله توزیع کلید و امضای دیجیتال محدود می‌شود.
(مطابق اهداف و انگیزه های اولیه طراحی)

حملات به رمزنگاری کلید عمومی

- جستجوی فراگیر (Brute force)
- محاسبه کلید خصوصی از کلید عمومی
  - اثبات نشده که غیر ممکن است!
- حمله پیام احتمالی (Probable-message attack)
  - مخصوص رمزنگاری کلید عمومی
  - در صورت کوچک بودن پیام (مثلا پیام، یک کلید ۵۶ بیتی DES باشد) می‌توان همه کلیدهای ممکن DES را با کلید عمومی رمز کرد و کلید رمز شده را پیدا کرد.

توابع یک طرفه دریچه

- **تابع یک طرفه (One-way func.):**

تابع $f(.)$ را یک طرفه گوییم اگر یافتن مقدار ورودی تابع از روی مقدار خروجی از لحاظ محاسباتی ناممکن باشد.

- **تابع یک طرفه دریچه (Trap-door one-way func.)**

تابع معکوس پذیر $f_k(.)$ با مشخصات زیر:

- محاسبه $y= f_k(x)$ با دانستن k و x آسان باشد،
- محاسبه $x= f_k^{-1}(y)$ با دانستن k و y آسان باشد،
- محاسبه $x= f_k^{-1}(y)$ با دانستن y و مخفی بودن k امکانپذیر نباشد.

توابع یک طرفه دریچه برای رمزنگاری کلید عمومی

- توابع یک طرفه دریچه ابزارهای مناسبی برای طراحی الگوریتم‌های رمزنگاری و امضای دیجیتال هستند.
- در حقیقت ثابت می‌شود وجود توابع یک طرفه دریچه شرط لازم و کافی برای وجود الگوریتم‌های رمزگذاری و امضای دیجیتال امن است.

فهرست مطالب

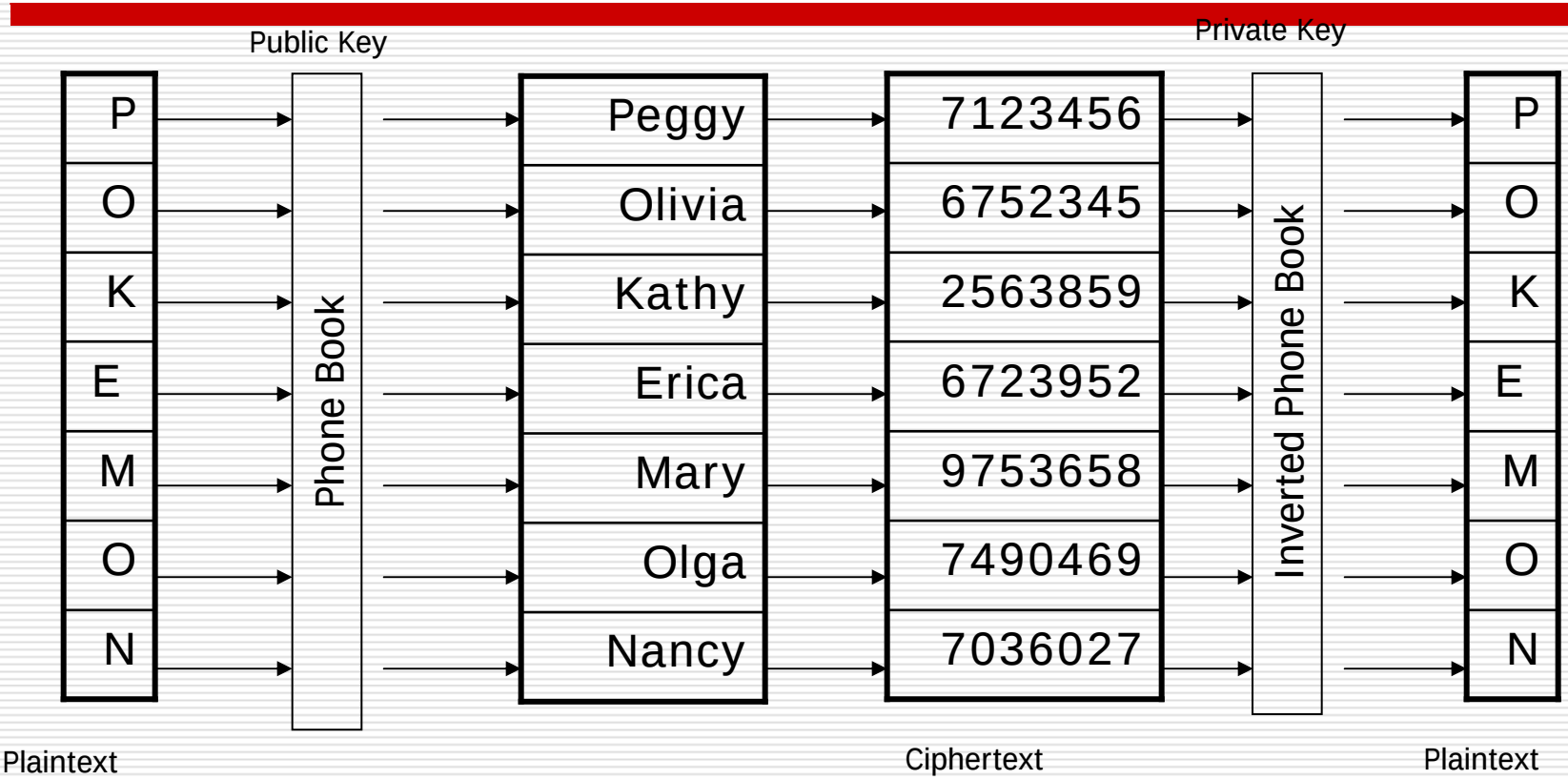
- مبانی رمزنگاری کلید عمومی
- مقایسه با رمزنگاری سنتی و متقارن
- کاربردهای رمزنگاری کلید عمومی
- **الگوریتم رمز RSA**
- الگوریتم رمز دیفی-هلمن

کلیات الگوریتم رمزنگاری RSA

- کلیات
  - توسط Rivest-Shamir -Adleman در سال ۱۹۷۷ در MIT ارائه شد.
  - مشهورترین و پرکاربردترین الگوریتم رمزگذاری کلیدعمومی
  - مبتنی بر توان رسانی پیمانه‌ای
  - استفاده از اعداد طبیعی خیلی بزرگ
  - امنیت آن ناشی از دشوار بودن تجزیه اعداد بزرگ، که حاصلضرب دو عامل اول بزرگ هستند، می‌باشد.
  - مستندات مربوط به آن تحت عنوان PKCS استاندارد شده است.

Public Key Cryptography Standards

A Simple Example



- Anyone can map from plaintext to ciphertext.
- Decryption easy only with inverted phone book.

One-way functions and trapdoors.

- A function f() is said to be *one-way* if given x it is "easy" to compute y = f (x), but given y it is "hard" to compute x = $f^{-1}(y)$.
- A trap-door one-way function $f_K()$ is such that to compute
  - $y = f_K(x)$ is easy if K and x are known.
  - $x = f^{-1}_K(y)$ is easy if K and y are known.
  - $x = f^{-1}_K(y)$ is hard if y is known but K is unknown.
- Given a trap-door one-way function one can design a public key cryptosystem.

Encryption and 1-way trap doors

- Two keys:
  - public encryption key *e*
  - private decryption key *d*
- Encryption easy when *e* is known
- Decryption hard when *d* is not known
- *d* provides “trap door”: decryption easy when *d* is known
- We’ll study the RSA public key encryption scheme. First we need some number theory.

Some Number Theory

- We'll need some number theory to define a one-way trap-door function:
  - Elementary (Review?):
    - Divisors
    - Prime numbers
    - relative primes
    - Modular arithmetic
  - Advanced (Hand-waving overview)
    - Euler's totient function
    - Lagrange's theorem

Divisors

- *x* divides *y* (written *x* | *y*) if the remainder is 0 when *y* is divided by *x*
  - 1|8, 2|8, 4|8, 8|8
- The divisors of *y* are the numbers that divide *y*
  - divisors of 8: {1,2,4,8}
- For every number *y*
  - 1|*y*
  - *y*|*y*

Prime numbers

- A number is prime if its only divisors are 1 and itself:
  - 2,3,5,7,11,13,17,19, ...
- Fundamental theorem of arithmetic:
  - For every number *x*, there is a unique set of primes $\{p_1, \ldots, p_n\}$ and a unique set of positive exponents $\{e_1, \ldots, e_n\}$ such that

$$x = p_1^{e1} * \ldots * p_n^{en}$$

Common divisors

- The common divisors of two numbers x,y are the numbers *z* such that *z*|*x* and *z*|*y*
  - common divisors of 8 and 12:
    - intersection of {1,2,4,8} and {1,2,3,4,6,12}

    = {1,2,4}
- greatest common divisor: *gcd(x,y)* is the number *z* such that
  - *z* is a common divisor of *x* and *y*
  - no common divisor of *x* and *y* is larger than *z*
    - gcd(8,12) = 4

Relative primes

- x and y are relatively prime if they have no common divisors, other than 1
- Equivalently, x and y are relatively prime if gcd(x,y) = 1
  - 9 and 14 are relatively prime
  - 9 and 15 are not relatively prime

Modular Arithmetic

- Definition: *x* is congruent to *y* mod *m*, if *m* divides *(x-y).* Equivalently, *x* and *y* have the same remainder when divided by *m*.

  Notation: $x \equiv y (\mathrm{mod}\, m)$

  Example: $14 \equiv 5 (\mathrm{mod}\, 9)$

- We work in Z_m = {0, 1, 2, …, m-1}, the ring of integers modulo m with binary operators + and * defined modulo m.
- *Example:* $Z_9 = \{0,1,2,3,4,5,6,7,8\}$
- We abuse notation and often write = instead of $\equiv$

Addition and Multiplication

- Many of the same properties as addition and multiplication of integers:
  - Commutative
  - Associative
  - Distributive
  - Additive inverses
- Some differences:
  - Some elements have multiplicative inverses

Addition in Z_m :

- Addition is well-defined:

$$if$$

$$x \equiv x' (\mathrm{mod}\, m)$$

$$y \equiv y' (\mathrm{mod}\, m)$$

$$then$$

$$x + y \equiv x' + y' (\mathrm{mod}\, m)$$

  - 3 + 4 = 7 mod 9.
  - 3 + 8 = 2 mod 9.

Additive inverses in Z_m

- 0 is the additive identity in Z_m

$$x + 0 \equiv x (\mathrm{mod}\, m) \equiv 0 + x (\mathrm{mod}\, m)$$

- Additive inverse
  - Every element has unique additive inverse.
  - 4 + 5= 0 mod 9.
  - 4 is additive inverse of 5.

Multiplication in Z_m :

- Multiplication is well-defined:

$$if$$

$$x \equiv x' (\mathrm{mod}\, m)$$

$$y \equiv y' (\mathrm{mod}\, m)$$

$$then$$

$$x \times y \equiv x' \times y' (\mathrm{mod}\, m)$$

  - 3 * 4 = 3 mod 9.
  - 3 * 8 = 6 mod 9.
  - 3 * 3 = 0 mod 9.

Multiplicative inverses in Z_m

- 1 is the multiplicative identity in Z_m

$$x * 1 \equiv x (\mathrm{mod}\, m) \equiv 1 * x (\mathrm{mod}\, m)$$

- Multiplicative inverse –
  - SOME, but not ALL elements have unique multiplicative inverse.
  - In Z_9 : 3*0=0, 3*1=3, 3*2=6, 3*3=0, 3*4=3, 3*5=6, …, so 3 does not have a multiplicative inverse.
  - On the other hand, 4*2=8, 4*3=3, 4*4=7, 4*5=2, 4*6=6, 4*7=1, so $4^{-1}=7$

Which numbers have inverses?

- In Z_m, x has a multiplicative inverse if and only if x and m are relatively prime
  - E.g., 3 and 4 in Z_9

Euler's totient function

- Given positive integer *n*, Euler's totient function $\Phi(n)$ is the number of positive numbers less than n that are relatively prime to n.
- Fact: If *p* is prime then
  - {1,2,3,…,*p*-1} are relatively prime to *p*.

$$\Phi(p) = p - 1$$

Euler's totient function

- Fact: If p and q are prime and $n=pq$ then

$$\Phi(n) = (p-1)(q-1)$$

- Each number that is not divisible by p or by q is relatively prime to pq.
  - E.g. p=5, q=7: {1,2,3,4,-,6,-,8,9,-,11,12,13,-,-,16,17,18,19,-,-,22,23,24,-,26,27,-,29,-,31,32,33,34,-}
  - (p-1)(q-1) = pq-p-q+1

Important Fact

- If *a* is relatively prime to n then

$$a^{\Phi(n)} \equiv 1 \bmod n$$

- (This is a corollary to a theorem due to Lagrange that states that the order of an element of a multiplicative group divides the order of the group. It's applied to the group Z_n^* of residues mod n that are relatively prime to n.)

RSA overview

- Alice wants people to be able to send her encrypted messages.
- She chooses two (large) prime numbers, *p* and *q* and computes *n*=*pq* and $\Phi(n)$. ["large" = 100 digits +]
- She chooses a number *e* such that *e* is relatively prime to $Z_{\Phi(n)}$ and computes *d*, the inverse of *e* in $\Phi(n)$
- She publicizes the pair *(e,n)* as her public key. She keeps *d* secret and destroys *p*, *q*, and $\Phi(n)$
- Plaintext and ciphertext messages are elements of Z_n and *e* is the encryption key.

RSA overview

- Bob wants to send a message *x* (an element of Z_n) to Alice.
- He looks up her encryption key, *(e,n)*, in a directory.
- The encrypted message is

$$y = E(x) = x^e \bmod n$$

- Bob sends *y* to Alice.

RSA overview

- To decrypt the message

$$y = E(x) = x^e \bmod n$$

she's received from Bob, Alice computes

$$D(y) = y^d \bmod n$$

Claim: $D(y) = x$

RSA encryption function is 1-way trap door

- Need to show
  - D[E[x]] = x
  - E[x] and D[y] can be computed efficiently if keys are known
  - $E^{-1}[y]$ cannot be computed efficiently without knowledge of the (private) decryption key d.
- Also, it should be possible to select keys reasonably efficiently
  - This does not have to be done too often, so efficiency requirements are

E and D are inverses:

Case 1: gcd(x,n)=1

$$D(y) = y^d \bmod n$$

$$\equiv (x^e \bmod n)^d$$

$$\equiv (x^e)^d \bmod n$$

$$\equiv x^{ed} \bmod n$$

$$\equiv x^{t\Phi(n)+1} \bmod n \qquad \text{Because } ed \equiv 1 \bmod \Phi(n)$$

$$\equiv (x^{\Phi(n)})^t \, x \bmod n$$

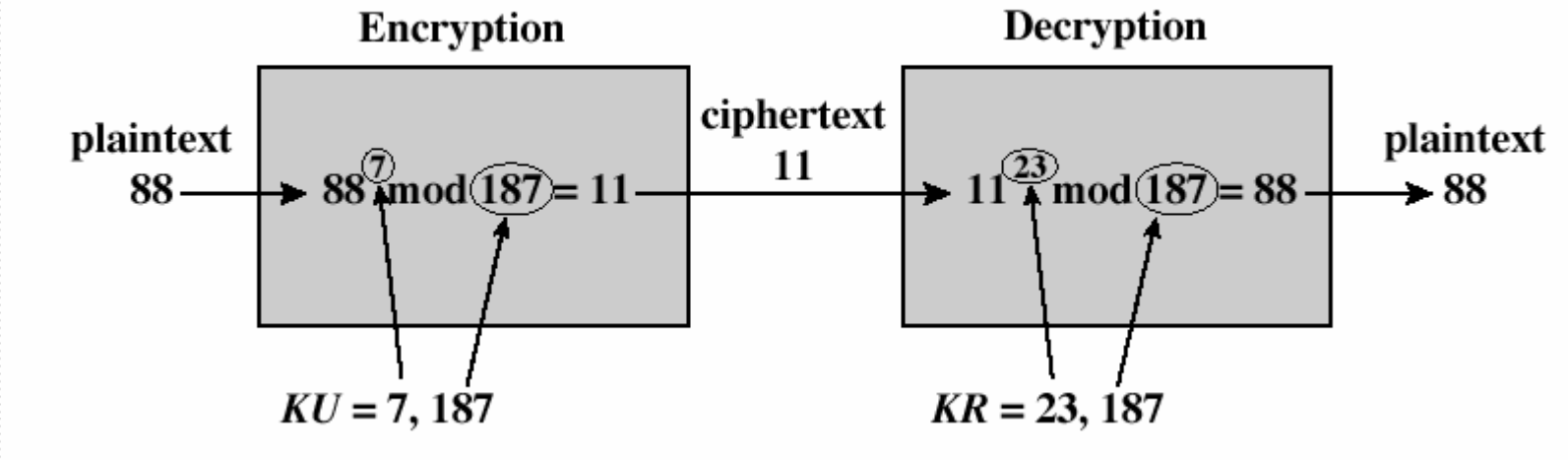
$$\equiv 1^t \, x \bmod n \equiv x \bmod n \qquad \text{From “important fact”}$$

Slide from an earlier lecture by Nasir Memon @ Poly-NYU

Tiny RSA example.

- Let p = 7, q = 11. Then n = 77 and
  $$\Phi(n) = 60$$
- Choose e = 13. Then d = 13^{-1} mod 60 = 37.
- Let message = 2.
- E(2) = 2^{13} mod 77 = 30.
- D(30) = 30^{37} mod 77=2

مثال–RSA



p = 17, q = 11, n = p.q= 187

φ(n) = 16.10 =160, pick e=7, d.e≡1 (mod φ(n))

➔ d = 23

Slightly Larger RSA example.

- Let p = 47, q = 71. Then n = 3337 and

$$\Phi(pq) = 46 * 70 = 3220$$

- Choose e = 79. Then d = 79^{-1} mod 3220 = 1019.
- Let message = 688232… Break it into 3 digit blocks to encrypt.
- E(688) = 688^{79} mod 3337 = 1570.

  E(232) = 232^{79} mod 3337 = 2756
- D(1570) = 1570^{1019} mod 3337 = 688.

  D(2756) = 2756^{1019} mod 3337 = 232.

RSA encryption function is 1-way trap door

- Need to show
  - ✓ D[E[x]] = x
  - E[x] and D[y] can be computed efficiently if keys are known
  - $E^{-1}[y]$ cannot be computed efficiently without knowledge of the (private) decryption key d.
- Also, it should be possible to select keys reasonably efficiently
  - This does not have to be done too often, so efficiency requirements are less stringent.

Decryption without trapdoor

- Suppose Oscar intercepts the encrypted message *y* that Bob has sent to Alice.
- Oscar can look up *(e,n)* in the public directory (just as Bob did when he encrypted the message)
- If Oscar can compute $d = e^{-1} \bmod \Phi(n)$ then he can use the formula $D(y) = y^d \bmod n = x$ to recover the plaintext *x*.
- If Oscar can compute $\Phi(n)$, he can compute *d* (the same way Alice did).

Decryption without trapdoor

- Oscar knows that *n* is the product of two primes
- If he can factor *n*, he can compute $\Phi(n)$
- But factoring large numbers is *very difficult*:
  - Grade school method takes $O(\sqrt{n})$ divisions.
  - Prohibitive for large n, such as 200 digits (roughly 512 bits)
  - Better factorization algorithms exist, but they are still too slow for large n
  - Lower bound for factorization is an open problem

How big should n be?

- Today we need n to be at least 768 bits. Better 1024 or even 2048 bits.
- No other (implementation independent) attack on RSA known.

نمادگذاری RSA

- n : پیمانه محاسبات
- e: نمای رمزگذاری
- d: نمای رمزگشایی
- M: پیام، عدد صحیح متعلق به Z_n^*
- تابع RSA: تابع یکطرفه دریچه $C = M^e \bmod n$
- تابع معکوس: $M = C^d \bmod n$

مبانی ریاضی RSA

- p و q دو عدد اول می‌باشند.
- φ(n): تعداد اعداد (کوچکتر از n) که نسبت به n اول است.
- کلید عمومی:{e,n}
- کلید خصوصی: {d,n}

$$n = p.q$$

$$f(n) = (p-1)(q-1)$$

$$\gcd(f(n), e) = 1, \quad 1 < e < f(n)$$

$$d.e \equiv 1 \bmod f(n), \; d \equiv e^{-1} (\bmod f(n))$$

$$C = M^e \bmod n, \quad M < n$$

$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

روند تولید کلید در RSA

1. ابتدا دو عدد اول بزرگ p و q را به طور تصادفی انتخاب کن به گونه‌ای که $p \neq q$
2. عدد n و $\varphi(n)$ را محاسبه کن $n=p.q$ و $\varphi(n)=(p-1).(q-1)$
3. عدد صحیح فرد e کوچکتر از $\varphi(n)$ را به گونه ای انتخاب کن که $\gcd(e, \varphi(n))=1$ باشد.
4. d را محاسبه کن $d \equiv e^{-1} \pmod{\varphi(n)}$
5. زوج $PU=(e,n)$ را به عنوان کلید عمومی اعلام کن.
6. زوج $PR=(d,n)$ را به عنوان کلید خصوصی ذخیره کن.

قراردادها و پروتکل RSA

- هم فرستنده و هم گیرنده مقدار n را می‌دانند.
- فرستنده مقدار e را می‌داند.
  - کلید عمومی : (n , e)
- تنها گیرنده مقدار d را می‌داند.
  - کلید خصوصی : (n, d)
- نیازمندی‌ها:
  - محاسبه M^e و C^d آسان باشد.
  - محاسبه d با دانستن کلید عمومی غیرممکن باشد.

روشهای کارا برای محاسبه نما

- برای محاسبه $a^b \pmod n$ الگوریتم‌های متفاوتی ابداع شده است...
  - فرض کنید $b_k b_{k-1} \ldots b_0$ نمایش مبنای ۲ عدد b باشد.
  - بنابراین خواهیم داشت:

$$a^b = a^{\sum_{b_i \neq 0} 2^i} = \prod_{b_i \neq 0} a^{2^i}$$

$$a^b \bmod n = \left[\prod_{b_i \neq 0} a^{2^i}\right] \bmod n = \left[\prod_{b_i \neq 0} \left(a^{2^i} \bmod n\right)\right] \bmod n$$

الگوریتم توان و ضرب

- بر این مبنا می‌توان الگوریتم زیر را طراحی نمود:

$c \neg 0; d \neg 1$

for $i \neg k$ *downto* 0

do $c \neg 2.c$ ⟹ c is prefix of b

$d \neg d^2 \bmod n$

if $b_i = 1$

then $c \neg c + 1$

$d \neg (d.a) \bmod n$ ⟹ $d = a^c \bmod n$

return d

مثال عددی الگوریتم توان و ضرب

اگر a، b و n با β بیت قابل نمایش باشند،

- نیاز به $O(\beta)$ عمل ریاضی

```
c ← 0; d ← 1
for i ← k downto 0
do c ← 2.c
   d ← d^2 mod n
   if b_i = 1
      then c ← c + 1
           d ← (d.a) mod n
return d
```

i	9	8	7	6	5	4	3	2	1	0
b_i	1	0	0	0	1	1	0	0	0	0
c	1	2	4	8	17	35	70	140	280	560
d	7	49	157	526	160	241	298	166	67	1

Figure 9.8 Result of the Fast Modular Exponentiation Algorithm for $a^b \bmod n$, where $a = 7$, $b = 560 = 1000110000$, $n = 561$

حملات ممکن بر RSA

حمله آزمون جامع(Brute Force)

- طول کلید با پیدایش هر نسل جدید از پردازنده‌ها افزایش می‌یابد، ضمن اینکه قدرت پردازشی هکرها زیاد می‌شود!

حملات ممکن بر RSA

- **حملات ریاضی**
  - تجزیه پیمانه n و در نتیجه محاسبه φ(n)
  - محاسبه φ(n) به صورت مستقیم
  - محاسبه d بدون استفاده از φ(n)
- در حال حاضر سختی همه راه‌های فوق معادل سختی مساله تجزیه اعداد بزرگ حاصل از ضرب دو عامل اول است.
- الگوریتم‌های مختلفی برای مساله تجزیه ارائه شده است (بهترین آنها LS است).
- در حال حاضر RSA با کلید ۱۰۲۴ تا ۴۰۹۶ بیت امن است.

Twenty Years of Attacks on the RSA Cryptosystem 1999, by Dan Boneh

حملات ممکن بر RSA

حمله زمانی

- زمان اجرای عملیات رمزگذاری یا رمزگشایی رمز می‌تواند اطلاعاتی را در مورد کلید افشاء کند.

راه‌های مقابله با حملات زمانی

- استفاده از توان رساندن با زمان ثابت محاسباتی
- اضافه کردن تاخیرهای تصادفی
- قرار دادن اعمال اضافی و گمراه کننده در بین محاسبات

فهرست مطالب

- مبانی رمزنگاری کلید عمومی
- مقایسه با رمزنگاری سنتی و متقارن
- کاربردهای رمزنگاری کلید عمومی
- الگوریتم رمز RSA
- **الگوریتم رمز دیفی-هلمن**

الگوریتم دیفی–هلمن

- توسط Diffie و Hellman در سال ۱۹۷۶ ارائه شد.
- برای تبادل کلید مورد استفاده قرار می‌گیرد.
- طرفین بر روی مقادیر q و توافق می‌کنند.
- q یک عدد اول و یک مولد برای این عدد است.
- امنیت روش مبتنی بر مشکل بودن لگاریتم گسسته است.

الگوریتم دیفی–هلمن

A

B

عمومی : α , q

مقدار تصادفی X_A را انتخاب می‌کند

مقدار تصادفی X_B را انتخاب می‌کند

$$Y_A = \alpha^{X_A} \bmod q$$

$$Y_B = \alpha^{X_B} \bmod q$$

$$K_{AB} = (Y_B)^{X_A} \bmod q$$

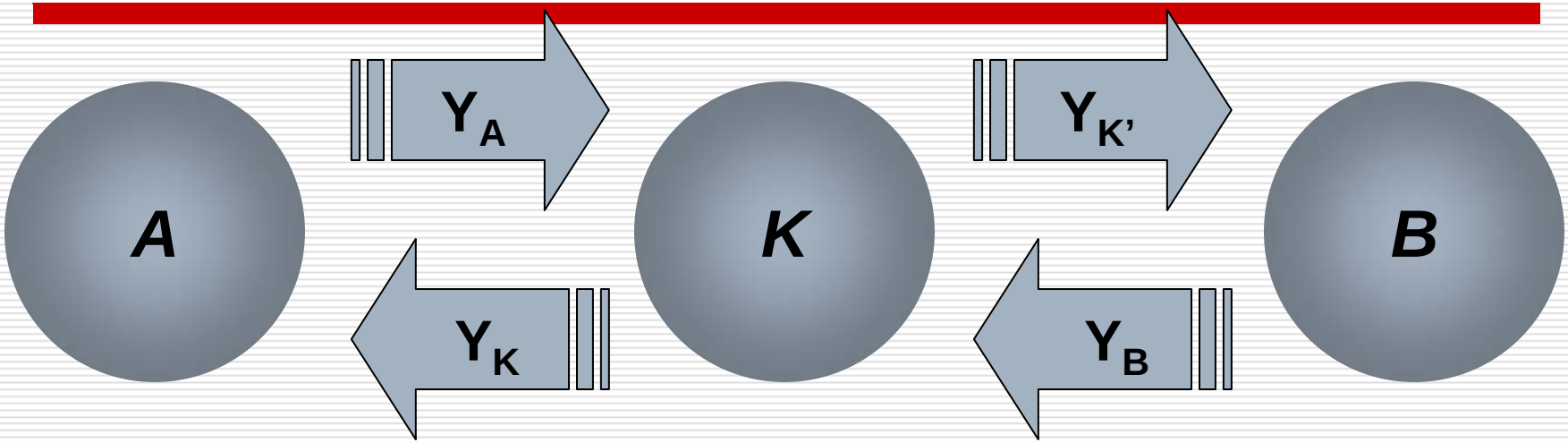
$$K_{AB} = (Y_A)^{X_B} \bmod q$$

کلید مشترک عبارت است از $\alpha^{(X_A \times X_B)} \bmod q$

حمله مرد میانی

- مهاجم به عنوان کانال ارتباطی میان طرفین عمل می‌کند.
- از نوع حملات فعال محسوب می‌شود.
- الگوریتم دیفی-هلمن را تهدید می‌کند.

حمله مرد میانی



$$K_1 = \alpha^{(X_A \times X_K)} \bmod q$$

$$K_2 = \alpha^{(X_A \times X_{K'})} \bmod q$$

A گمان می‌کند کلید K_1 را با B به اشتراک گذاشته است.

B گمان می‌کند کلید K_2 را با A به اشتراک گذاشته است.

کاربردهای برخی الگوریتم‌های کلید عمومی

الگوریتم	رمزگذاری/ رمز گشایی	امضاء رقمی	توزیع کلید
RSA	√	√	√
Diffie-Hellman	×	×	√
DSS	×	√	×
Elliptic Curve	√	√	√

پایان

مرکز امنیت شبکه شریف

http://nsc.sharif.edu

پست الکترونیکی

m_amini@ce.sharif.edu

درستی RSA

- بر اساس تئوری اولر
  - اگر $gcd(a,n)=1$ باشد، آنگاه $a^{\varphi(n)} \bmod n = 1$
- در RSA داریم:
  - $n=p.q$
  - $\varphi(n)=(p-1).(q-1)$
  - $d \equiv e^{-1} \pmod{\varphi(n)}$ و لذا $e.d=1+k.\varphi(n)$
- بنابراین
  - $C^d = M^{e.d} = M^{1+k.\phi(n)} = M^1.(M^{\phi(n)})^k = M^1.(1)^k = M^1 = M \bmod n$