Асиметрично криптиране

доц. д-р М. Ненова
Доц. д-р Венцислав Трифонов

Симетрична криптография

- Традиционната критпография използва частен**/секретен/единичен ключ**
- Споделян между изпращащия и приемащия
- Ако този ключ е известен комуникацията е компрометирана
- Симетричен е защото участниците в комуникацията са равнопоставени
- Затова изпращащия съобщението не е защитен от получателя, че няма да подмени съобщението и да твърди, че е изпратено така от подателя

РКС: модерната криптография

- През 1970-те криптографията с публичен ключ навлиза все повече.

- Всеки потребител има два ключа взаимозаменяеми;
- Ключът за криптиране е публичен;
- Ключът за декриптиране е секретен.

- Тоест всеки може да изпрати съобщение на Вас, но само Вие можете да го прочетете.

RSA

- Най-широко използвания алгоритъм за РКС е RSA, базира се на трудността да се намерят делителите на произведението от две големи прости числа.

- **Лесна задача**

  Дадени са две големи прости числа p и q
  Да се изчисли

  $$n = p \times q$$

- **Трудна задача**

  Дадено е n
  да се определят p и q.

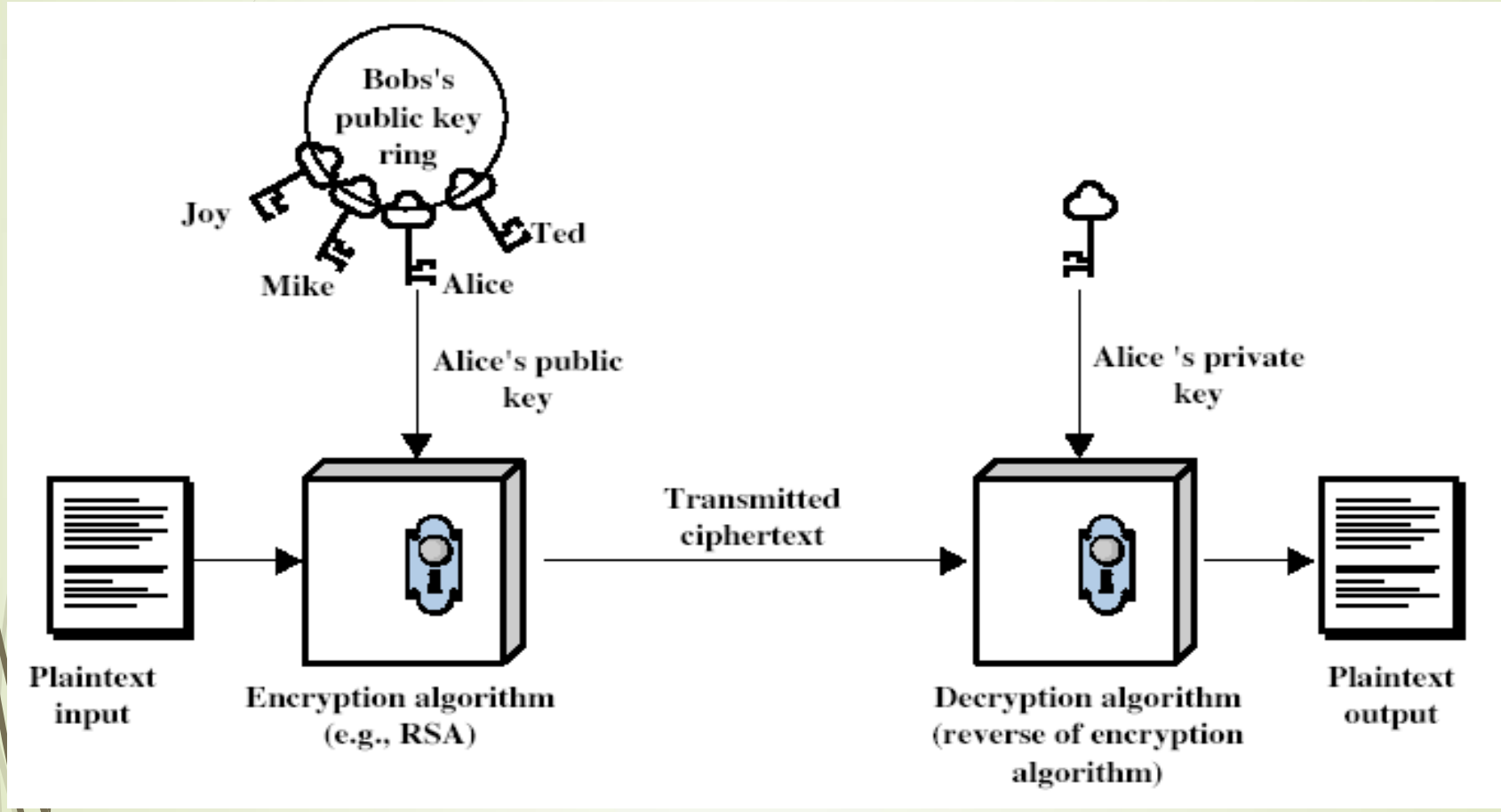
Асиметрична криптография

- Най-значимият напредък през 3000 години история на криптографията
- Използва **два** ключа – публичен & частен ключ
- **Асиметричен**, защото участниците **не са равни**
- Базира се на приложение на теорията на числата
- **Допълва** по-скоро отколкото да измества симетричната криптография

Асиметрична криптография

- **Публичен ключ/два ключа/асиметрична** криптография използва два ключа:
  - **Публичен ключ**, който може да се знае от всеки и може да се използва за **криптиране на съобщението** и за **проверка на подписи**
  - **Частен ключ**, известен само на получателя, изпозлван за **декриптиране на съобщението** и **подписването** (създаването) на подписи
- **асиметрично** е защото
  - Този, който криптира съобщението или проверява подписите не може да декриптира съобщението или да създаде подписи

Асиметрична криптография



Защо се използва симетричната криптография?

- Разработена е за да реши два основни проблема:
  - **Изпращане на ключа** – как да се направи сигурна комуникация без да има доверие в центъра за разпределение на ключовете със собствения ключ
  - **Цифрови подписи** – как да се провери съобщението дали наистина идва от дадения източник
- разработен от Whitfield Diffie & Martin Hellman във Университета в Станфорд през 1976
  - Известен само на ограничен кръг хора

Асиметрична криптография

- Алгоритмите за асиметрична криптография се базират на:
  - **изчислителна невъзможност** да се намери ключа за декриптиране, знаейки само алгоритъма и криптиращия ключ
  - лесно от изчислителна гледна точка да се криптира/ декриптира съобщението със съответния ключ, ако той се знае
  - един от двата ключа се използва за криптиране, а другият за декриптиране

Приложение на асиметричната криптография

- Могат да се разделят на три категории:
  - **криптиране/декриптиране** (осигурява секретност)
  - **цифрови подписи** (осигурява автентификация)
  - **предаване на ключове** (за сесийни ключове)
- Някой алгоритми са подходящи за всички горепосочени, други са специфични

Сигурност при асиметричните схеми

- Както и при симетричните, схемите за **brute force** атака са винаги теоретично възможни
- Но тук използваните ключове са много големи (>512bits)
- Сигурността е базирана на **достатъчно големи** разлики в трудността между **лесните** (криптиране и декриптиране) и **трудните** (криптоанализ) проблеми
- За труден се смята проблем, който е трудно да се реализира практически
- Изисква използването на **много големи числа**
- Затова е и бавен в сравнение със симетричните схеми

RSA

- Rivest, Shamir & Adleman от MIT през 1977
- Най-известната и широко използвана схема
- Базирана на повдигането на степен в крайни полета (Galois) с цели числа **modulo** от прости числа
  - заб. Това изисква $O((\log n)^3)$ операции (лесно)
- Използва големи цели числа (напр. 2048bits)
- Сигурност дължаща се на ресурса необходим за разделянето на множители на големи числа
  - заб. Това изисква $O(e^{\log n \log \log n})$ операции (изключително трудно)

RSA генериране на ключа

- Всеки потребител генерира двойка публичен и частен ключ, така:
- Избират се две големи пройзволни прости числа p,q
- Изчислява се тяхното произведение $N=p.q$
  - като $\varnothing(N)=(p-1)(q-1)$
- Избира се пройзволно e – криптраща експонента, такава, че

$$1<e<\varnothing(N),\ \gcd(e,\varnothing(N))=1$$

- Решава се уравнението за да се намери декриптиращия ключ d
  - $e.d=1 \mod \varnothing(N)$ and $0\leq d\leq N$
- Обявява се техния публичен ключ: KU={e,N}
- Запазва се скрит частния ключ: KR={d,p,q}

RSA Приложение

- За да се криптира съобщението M изпращащия:
  - Взема **публичния ключ** на получателя $KU=\{e,N\}$
  - Изчислява: $C=M^e \bmod N$, където $0 \leq M < N$
- За да се декриптира съобщението/криптирания текст C :
  - използва получателя частния си ключ $KR=\{d,p,q\}$
  - изчислява: $M=C^d \bmod N$
- Трябва съобщението да е по-малко от произведението N

Прости числа

- Простите числа имат делител 1 и самите себе си
  - Те не могат да се запишат като произведение от други числа
  - Забележка: 1 е просто число, но е неприложимо
- Така 2,3,5,7 са прости 4,6,8,9,10 не са
- Простите числа са основата на теорията на числата
- Това са числата по малки от 200 и прости:

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61
67 71 73 79 83 89 97 101 103 107 109 113 127 131
137 139 149 151 157 163 167 173 179 181 191 193
197 199
```

Разлагане на множители

- За да се разложи число `n` на множители се представя във вида: `n=a × b × c`
- Като тази операция е сложна в сравнение с умножението на числата
- **Разложението** на прости множители на числото `n` е:
  - `91=7×13 ; 3600=2`^{4}`×3`^{2}`×5`^{2}

$$a = \prod_{p \in P} a^{a_p}$$

Функция на Ойлер ø(n)

- При аритметична опреация **modulo n**
- **Пълният набор от резидиуми е**: `0..n-1`
- **Редуцираният набор от резидиуми,** които са взаимно прости на n
  - Тоест за n=10,
  - Пълният набор от резидиуми е {0,1,2,3,4,5,6,7,8,9}
  - Редуцираният набор от резидиуми е {1,3,7,9}
- Броят на елементите в редуцирания набор се нарича Ойлерова функция **Euler Totient Function ø(n)**

RSA Пример

1. Избор на прости числа: $p=17$ & $q=11$
2. Изчисление на: $n = pq = 17 \times 11 = 187$
3. Изчисляваме: $\varnothing(n)=(p-1)(q-1)=16 \times 10=160$
4. Избираме e: $\gcd(e,160)=1$; избираме $e=7$
5. Определяме d: $de=1 \bmod 160$ и $d < 160$
   Стойността е: $d=23$ след като $23 \times 7=161= 10 \times 16+1$
6. Публичният ключ е $KU=\{7,187\}$
7. Пазим секретният ключ $KR=\{23,17\}$

RSA Пример

- Пример за RSA криптиране и декриптиране е:
- За съобщение $M = 88$ (като $88<187$)
- Криптиране:

$$C = 88^7 \bmod 187 = 11$$

- Декриптиране:

$$M = 11^{23} \bmod 187 = 88$$

RSA генериране на ключа

- потребителите на RSA трябва:
  - Определяне на две прости случайни числа - `p, q`
  - Избор на `e` или `d` и се изчисляват
- Простите числа `p,q` не трябва лесно да се получават от произведението `N=p.q`
  - Означава, че трябва да е огромно/голямо
  - Да е устояло на тестовете за вероятност
- Експонентите `e, d` са инверсни, използва се инверсен алгоритъм за да се получи едното или другото

Сигурност на RSA / Атаки

- Има три подхода за атака на RSA:
  - преминаване през всички комбинации на ключа/**brute force** (невъзможно при определен размер на числата);
  - математически атаки (базирани на трудността на изчислението на ø(N), чрез модул от N);
  - времеви атаки (докъто тече декриптирането).

Разделяне на множители

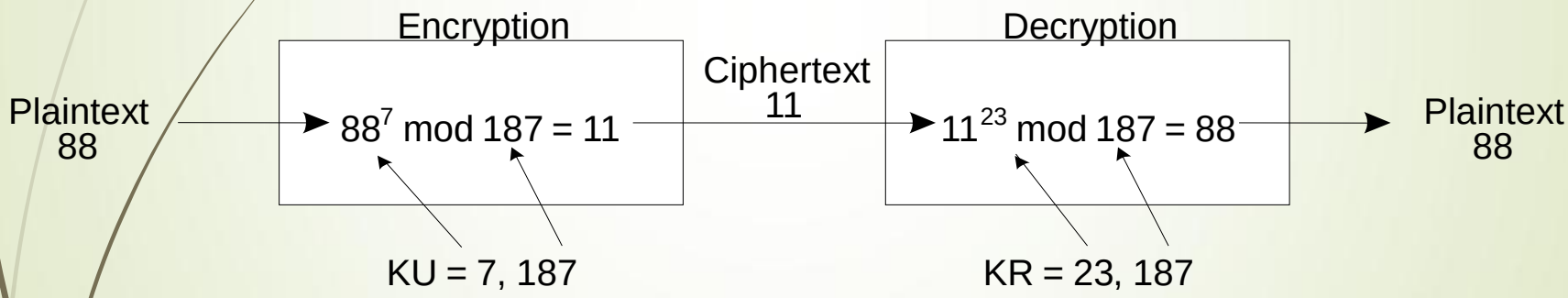
- Математическият подход има три форми:
  - Разделянето на $N=p.q$, така да се намери $\varnothing(N)$ и после d
  - Определяне на $\varnothing(N)$ директно и намирането на d
  - Намирането на d директно
- факторизиране
  - Наблюдава се бавен напредък през годините
    - От Авг-99 най-доброто е 130 цифрено число (512) bit с GNFS
  - Най-голям успех има чрез доказаните алгоритми
    - cf “Quadratic Sieve” to “Generalized Number Field Sieve”
  - Блокиране на пробив при 1024+ bit RSA
    - Доказване, че p, q са с подобен размер и да се нагодят другите константи

Времеви атаки

- разработени са в средата на 1990-те
- използват времеви вариации в операциите
- Влияят върху размера на операндите базирани на времето, което отнема
- RSA изпозлва времето отнемащо **степенуването**
- контрамерки
  - Използване на „constant exponentiation time“
  - Добавяне на случайни закъснения
  - Неопределности използвани в изчисленията

Задачи

1. Криптирайте и декриптирайте използвайки алгоритъм RSA, показан на фигурата, за следните данни:
   ① p = 3; q = 11, e = 7; M = 5
   ② p = 5; q = 11, e = 3; M = 9



2. В система базирана на RSA, прихващате шифриран текст *C = 10* изпратен на потребител, чийто публичен ключ е *e = 5*, *n = 35*. Какъв е явния текст M?