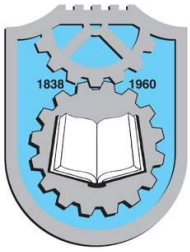
Универзитет у Крагујевцу
ФАКУЛТЕТ ИНЖЕЊЕРСКИХ НАУКА



Заштита података (Технике криптовања)

Асиметрична криптографија
RSA

Професор:
Проф. Владимир Миловановић

Студент:
Матеја Вујсић 617/2017

Крагујевац **, фебруар 2021. године**

САДРЖАЈ

Увод

Појава асиметричне криптографије је највероватније једини револуционарни догађај у целокупној историји криптографске мисли.

Од почетка модерног времена па до тог догађаја, сви криптосистеми заснивали су се на операцијама пермутације и замене. У основи, асиметрична криптографија напушта те устаљене градивне основе криптоалгоритама и прави јединствени заокрет, **укључује теорију бројева и математичке функције**.

Најважнија одлика ове гране криптографије је **коришћење два одвојена(али повезана кључа), приватног и јавног,** као контраст симетричним криптографским алгоритмима који користе један кључ за шифровање-дешифровање. Коришћење два кључа доноси бенефите на пољу поверљивости, дистрибуције кључа и аутентификације.

Међутим, иако је револуционарни приступ шифровању, постоји доста заблуда у вези ње. Често се ова грана криптографских алгоритама сматра супериорном на пољу ефикасности, сигурности, и размени кључева у односу на грану криптографских алгоритама који се темеље на коришћењу симетричне енкрипције. Са становишта криптоанализе, не постоји ниједан параметар који показује да је једна грана супериорнија од друге, стога не требају се имплементирати a priori. Заправо, сигурност неке криптографске шеме лежи у дужини кључа и количини израчунавања потребном да се разбије неки шифрован податак.

Далеко највише коришћена асиметрична шема која се данас користе јесте **RSA** шема, објављена 1977. године.

Кроз овај семинарски рад пролази се кроз основна начела асиметричне криптографије, начина функционисања асиметричних криптографских шема и криптоанализе уопштено и кроз пример RSA алгоритма.

1. Принципи асиметричне криптографије

Концепт асиметричне криптографије замишљен је да премости два највећа проблема симетричне енкрипције.

Први проблем је дистрибуција кључа, која код ове врсте шифровања захтева или да учесницима у комуникацији већ познат заједнички кључ (1) или да постоји трећи члан који учесницима комуникације дистрибуира кључ (2). Само постојање трећег члана комуникације је већ проблематично и као такав нарушава приватност разговора, пошто и он зна дељени кључ.

Пионири ове области криптографије попут **Мартина Хелмана** и **Витфилда Дифиеа** о проблему постојања центра за дистрибуцију дељеног кључа (2) постављали су следеће питање: „Која је бенефит дизајнирања непробојног криптосистема ако су корисници истог приморани да поделе кључ комуникације са трећим ентитетом за ког знамо да може бити компромитован? “

Други проблем, који није повезан са првим, је постојање дигиталних потписа. Као што се претпостављало, коришћење криптографије је превазишло коришћење у војне сврхе, и појавило се у свакодневном свету. Било је потребно осмислити пандан потпису у физичком свету, потпис електронских порука и докумената.

Дифие и Хелман, професори на Стенфорд Универзитету постигли су пробој те 1976. године постављајући метод који једнако решава оба проблема и притом мења све устаљене криптографске основе.

1.1 Асиметрични криптосистеми

Асиметрични алгоритам темељи се на кључу који шифрује податке, и различитом кључу који је повезан са првим за дешифровање података. Кључ за шифровање података назива се **јавни кључ** (енг**. public-key)**. Кључ који дешифрује податке је **приватан кључ** (енг. **private-key**). Ови алгоритми имају следећу карактеристику:

- Посматрајући излаз алгоритма шифровања, знајући алгоритам као и јавни део кључа практично је немогуће одредити приватни кључ а самим тим и дешифровати поруку.

Неки алгоритми, као што је RSA, такође додају и:

- Један од два кључа се користи за шифровање, док се други користи за дешифровање података.

Асиметрична криптографска шема темељи се на шест основних ствари:

- **Отворени текст** (енг. **Plain-Text**): Податак који је улаз у алгоритам шифровања и који је разумљив и читљив.
- **Алгоритам шифровања**: Шема коју примењујемо на податке које желимо да шифрујемо.
- **Приватни и јавни кључ (**енг. **Public key,** енг**. Private key):** Пар кључева тако изабраних да ако се један изабере да шифрује податке други служи да дешифрује и обрнуто. Приватни и јавни кључ казују алгоритму коју операцију над улазом треба да примени.
- **Шифрована порука (**енг. Chiper**):** Порука која се преноси преко несигурног канала и настаје као излаз из алгоритма шифровања. За исти улаз, два различита кључа произвешће два различита шифрата.
- **Алгоритам дешифровања** : Алгоритам прихвата шифровану поруку и одговарајући кључ, примењује шему и долази до оригиналне поруке.

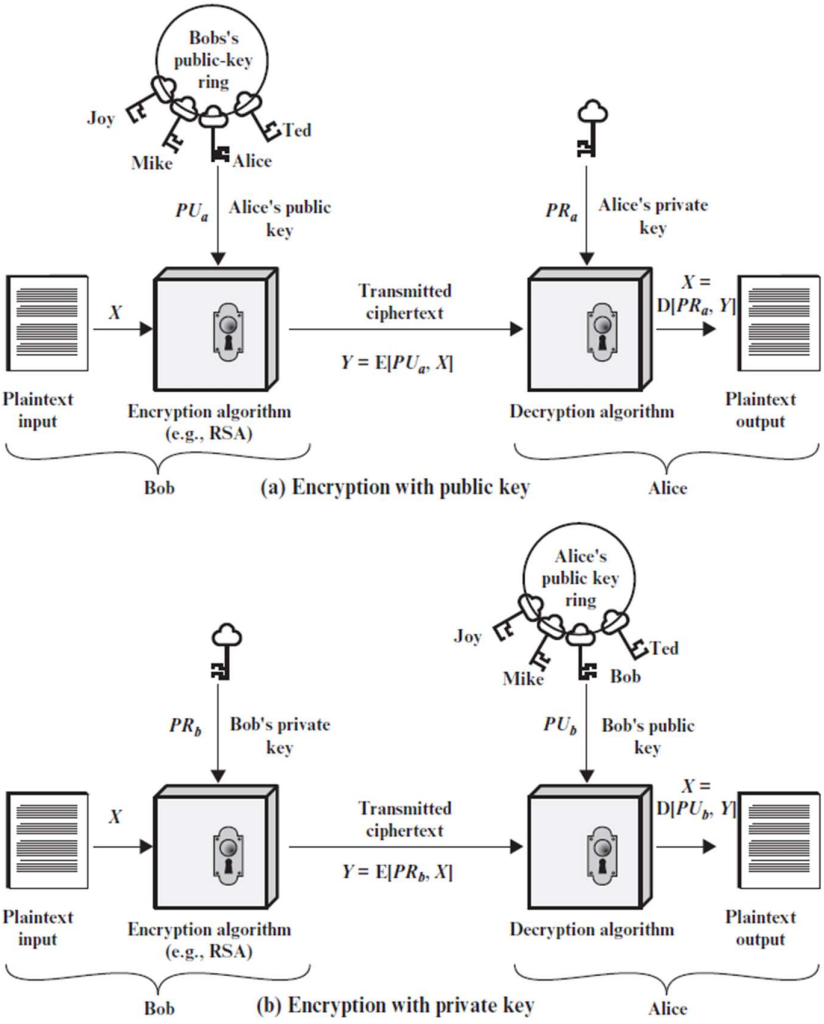
У основни кораци су следећи:

1. Сваки корисник генерише пар кључева који се користе за шифровање и дешифровање порука.

2. Корисник смешта један од два кључа у јавни регистар или други доступни фајл. Други кључ се држи тајности. Као што слика1 приказује, сваки корисник чува колекцију јавних кључева других корисника.

3. У колико Боб жели да пошаље шифровану поруку Алиси, шифрује је уз помоћ Алисиног јавног кључа.

4. Када Алис добије шифровану поруку, поруку дешифрује уз помоћ њеног приватног кључа. Било који ентитет који пресретне шифровану поруку није у могућности да је дешифрује, зато што Алис приватни кључ држи у тајности.



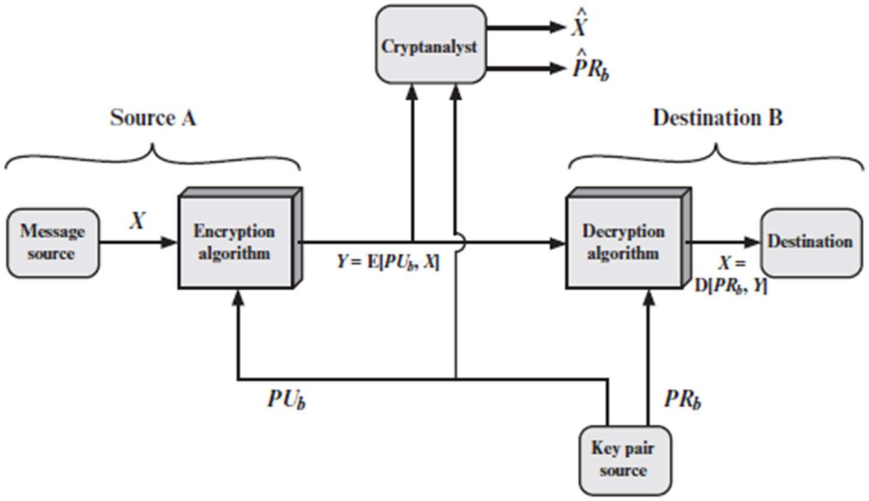
Слика1. (а)Шифровање јавним(б)Шифровање приватним кључем

Оваквим приступом сваком учеснику комуникације доступни су јавни кључеви, а приватни кључеви су генерисани на локалним машинама и њихова природа налаже да они немају потребе да се дистрибуирају. Све док се приватни кључ држи у тајности, долазећа комуникација је сигурна. У свако доба систем може да промени приватни кључ, са тим и јавни кључ, и да све учеснике о обавести о том догађају.

Симетрични алгоритми	**Асиметрични алгоритми**
Потребно за рад	***Потребно за рад***
• Један кључ и један алгоритам за шифровање и дешифровање. • Мора се разменити кључ и алгоритам шифровања.	• Један алгоритам се користи за шифровање једним, сличан за дешифровање другим кључем. • Прималац и пошиљалац морају да знају одговарајуће кључеве.
Потребно за сигурност	***Потребно за сигурност***
• Кључ мора да се држи у тајности. • Немогуће је дешифровати поруку уколико се не зна кључ шифровања. • Чак и уколико је позната шифрована порука и алгоритам шифровања, не би требало да је могуће открити тајни кључ.	• Један од два кључа се држи у тајности. • Практично је немогуће дешифровати поруку без знања једног од кључа, • Чак и ако је познат јавни кључ и алгоритам шифровања немогуће је ефикасно установити други део кључа.

Табела1. Разлика између асиметричних и симетричних алгоритама

Табела1. осликава разлику симетричних и асиметричних криптографских шема кроз кључне криптографске аспекте. Како би се избегла конфузија између криптографских шема које користимо, обично када се каже тајни кључ(енг. secret key) мисли се на кључ код симетричних алгоритама наспрам приватног кључа код асиметричних шема.



Слика2. Асиметрични криптосистеми - Поверљивост

Корисник А(Слика2.) жели да пошаље поруку X, X = $[X_1, X_2, X_3, \dots, X_m]$. M-елемената поруке X су слова алфабета(који је коначан). Порука је насловљена за корисника B. Корисник B генерише пар кључева, јавни кључ PU_b и приватни кључ PR_b. Приватни кључ корисник B чува у тајности док је јавни кључ доступан свима па самим тим и кориснику A.

Са познавањем јавног кључа PU_b и поруке X као улаза, корисник A формира шифровану поруку Y = $[Y_1, Y_2, Y_3, \dots, Y_m]$:

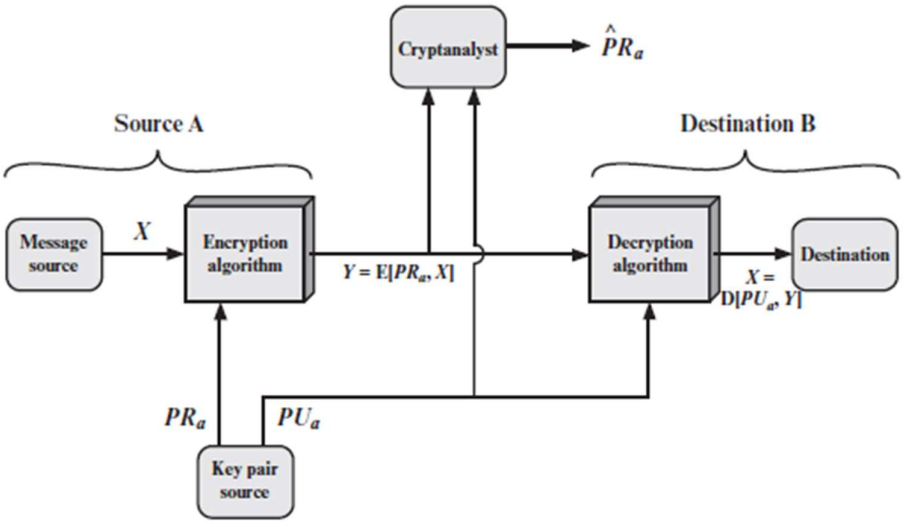
$$\mathbf{Y = E(P}\boldsymbol{U_b, X)}$$

Прималац, који у свом поседу има PR_b, кључ који одговара PU_b, је у могућности да нађе инверзну функцију:

$$\boldsymbol{X} \mathbf{= D(}\boldsymbol{PR_b, Y}\mathbf{)}$$

Неки нежељени посматрач пресреће шифровану поруку Y и доступан му је PU_b. Покушаће да поврати X и/или PR_b. Такође се претпоставља да посматрач зна и алгоритам шифровања E и алгоритам дешифровања D. Уколико је посматрач заинтересован за конкретну поруку покушаће да направи $\check{X}$, који генерише идентичну шифровану поруку. Уколико жели да има увид у долазећу комуникацију, што је најчешће, покушаће да пронађе PR_b.

Поменуто је да да у пару одговарајућих кључева, један служи за шифровање док други служи за дешифровање. Ова функционалност омогућава имплементацију на различите начине као и примену у различите сврхе. Једна оваква имплементација користи се да пружи аутентичност.



Слика3. Асиметрични криптосистем - аутентичност

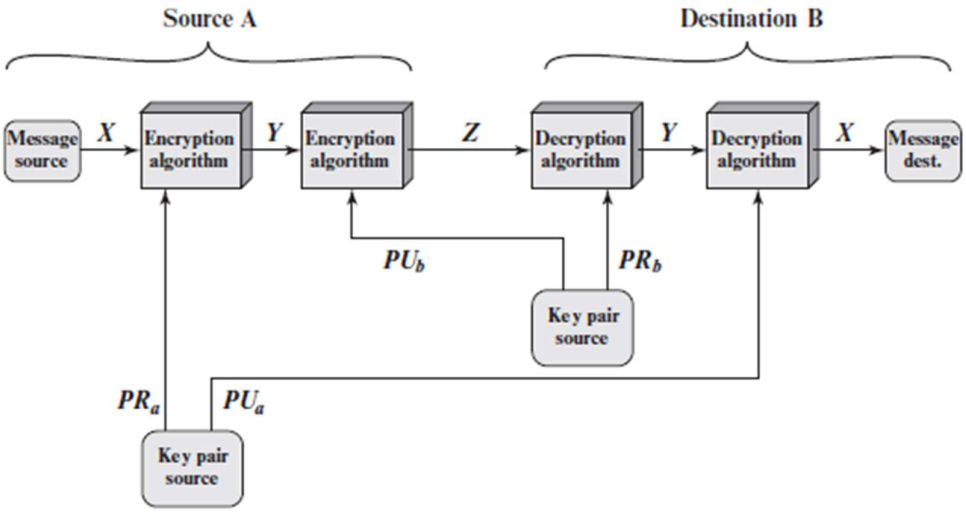
$$\boldsymbol{Y = \mathrm{E}(PR_a, X)}$$

$$\boldsymbol{X = \mathrm{D}(PU_a, Y)}$$

У овом случају, А припрема поруку за В и шифрује је користећи свој приватни кључ, након што је пошаље кориснику В. Пошто је порука шифрована приватним кључем PR_{a}, једино је могла доћи од корисника А. Каже се да је шифрована порука **дигитално потписана**(енг. **digital signature).** Такође, садржај поруке је немогуће изменити без познавања PR_{a}, па је порука сигурна и у смислу од кога је дошла и садржаја. Иако је порука сигурна у ова два смисла, често овакав приступ дигиталном потпису није ефикасан, из простог разлога што морамо имати оригиналну поруку и шифровану верзију како би одредили аутентичност што захтева доста простора. Једно просто решење овог проблема налаже да се уместо шифровања целог документа, шифрују само неколико блокова који су функција поруке. Ови блокови називају се аутентикатори(енг. **authenticators),** и морају имати функционалност такву да се са мењањем садржаја поруке мења се и њихова вредност. Они као такви служе да потврде порекло и садржај поруке.

Горе наведени криптосистем пружа аутентичност поруке али не осигурава поверљивост. Једноставно свако ко има јавни део кључа може да прислушкује и дешифрује комуникацију.

Могуће је осигурати поверљивост а притом и задржати аутентичност саме поруке, користећи дупло шифровање асиметричном шемом:



Слика4. Асиметрични криптосистем - Аутентичност и поверљивост

$$Z = \mathrm{E}(PU_b, \mathrm{E}(PR_a, X))$$

$$X = \mathrm{D}(PU_a, \mathrm{D}(PR_b, Z))$$

Уколико желимо да осигурам и аутентичност и поверљивост, пре него што шифрујемо поруку са јавним кључем примаоца, шифрујемо поруку са приватним кључем пошиљаоца. То обезбеђује да је порука дигитално потписана. Шифрована порука може бити дешифрована једино од оне стране којој је порука намењена, пошто само та страна има одговарајући приватан кључ. Иако овакав дизајн криптосистема доноси и поверљивост и аутентичност, негативна страна је та што се алгоритамска асиметрична шема извршава четири, уместо два пута у свакој комуникацији.

1.2 Примена асиметричних криптосистема

Сваки асиметрични криптосистем карактерише употреба криптографског алгоритма са два кључа, од којих се један држи у тајности док је други јавно доступан. У зависности који ефекат пошиљалац жели да постигне, он шифрује поруку уз помоћ свог приватног кључа и/или уз помоћ јавног кључа примаоца.

У најужем смислу, у зависности за шта се користе, класификација асиметричних криптографских алгоритама урађена је на следећи начин:

- **Шифровање/дешифровање** – Пошиљалац шифрује поруку уз помоћ јавног кључа примаоца. Прималац дешифрује поруку уз помоћ приватног кључа
- **Дигитални потписи** – Пошиљалац „потписује“ поруку својим приватним кључем. Потписивање се остварује тако што се алгоритам примењује на цео садржај поруке или на неколико блокова које представљају функцију поруке.
- **Размена кључева(енг. key-exchange) –** Две стране покушавају да сигурно размене сесијски кључ(**енг. session-key**) који је заправо тајни кључ симетричне криптографске шеме који важи за одређену трансакцију или за сесију и који важи само одређен временски период.

Неки асиметрични криптографски алгоритми погодни су за имплементацију свих наведених функција па спадају у све три категорије, док су неки специјализовани само за једну или две функције. Табелом2 дати су неки асиметрични алгоритми и ситуације у којима се могу искористити.

Алгоритам	**Шифровање/дешифровање**	**Дигитални потпис**	**Размена кључа**
RSA	ДА	ДА	ДА
Ecliptic Curve	ДА	ДА	ДА
Diffie-Hellman	НЕ	НЕ	ДА
DSS	НЕ	ДА	НЕ

Табела2. Асиметрични алгоритми и функције

1.3 Концепти рада

Дифије и Хелман поставили су услове које криптографски алгоритам мора да задовољи да би постао асиметрични криптографски алгоритам. Услови су:

- Корисник B лако прорачунава(генерише) пар кључева (јавни кључ PU_b, приватни кључ PR_b).
- Корисник A, који је пошиљалац, знајући PU_b, лако примењује алгоритам на оригинални текст и генерише шифровани:

$$\mathbf{C = E(}\boldsymbol{PU_b}\mathbf{, M)}$$

- Корисник B шифровани текст, уз помоћ кључа који чува у тајности PR_b, лако примењује шему дешифровања и долази до оригиналне поруке.

$$\mathbf{M = D(}\boldsymbol{PR_b}\mathbf{, C) = D[}\boldsymbol{PR_b}\mathbf{, E(}\boldsymbol{PU_b}\mathbf{,M)]}$$

- Уколико је познат PU_b, немогуће је одредити PR_b.
- Немогуће је закључити нешто о оригиналном тексту на основу PU_b и шифрованог текста.
- И један и други кључ могу да послуже шифровању поруке, са тим да други кључ служи за дешифровање.

Као доказ да су услови тешко остварив сведоче само пар алгоритама(RSA, Diffie-Hellman, Ecliptic Curve, DSS) који су достигли општу примену у дигиталном свету. Срж асиметричног криптографског алгоритма лежи у добром избору једносмерне функције(**one-way function**). Једносмерна функција је функција која има такву инверзну функцију чију је вредност готово немогуће израчунати, док се за њу врло лако израчунава вредност :

$$\boldsymbol{Y} = \mathbf{f(}\boldsymbol{X}\mathbf{)} \quad \textbf{лако}$$

$$\boldsymbol{X} = \mathbf{f}^{-1}\mathbf{(}\boldsymbol{Y}\mathbf{)} \ \textbf{неизводљиво}$$

Генерално, термин лако користи се за проблем који се може решити у полиномијалном времену кроз функцију величине улаза. Термин неизводљиво је много теже одредити. Генерално каже се да је решење проблема неизводљиво, ако време његовог израчунавања сигурно траје дуже од извршавања полиномијалног алгоритма. Једносмерна функција лако се израчунава у једном смеру, са друге стране немогуће је израчунати инверзију без познавања још неких информација. Управо те информације су кључне да инверзија може да се израчуна у полиномијалном времену. У суштини једносмерна функција треба да

буде слична функцијама које немају инверзију, и њена дизајн мора да прати следеће карактеристике:

$$Y = f_k(X) \quad \textbf{лако, познати } k \textbf{ и } X$$
$$X = f_k(Y) \quad \textbf{лако, познати } k \textbf{ и } Y$$
$$X = f_k^{-1}(Y) \quad \textbf{неизводљиво, непознато k}$$

1 .4 Анализа криптосистема

Као и код симетричног типа шифровања, асиметричан алгоритам је такође рањив исцрпним (енг. Brute Force) нападима. Овај проблем се решава употребом дугих кључева. Међутим мора се обратити пажња на следеће: резултат математичке функције, која је у основи асиметричног криптографског алгоритма, дуже се рачуна са порастом броја битова у кључу. То значи да кључ не сме бити ни предугачак, шифровање би било непрактично, а ипак не сме бити ни прекратак, што доводи до БФ рањивости. Употреба дугих кључева у овим алгоритмима резултира лошим перформансама и шифровање/дешифровање траје дуго. Зато се ови алгоритми користе за размену кључева и дигиталне потписе.

Још једна форма напада јесте покушај да се на основу јавног кључа, доступног свима израчуна приватан кључ. Овакав покушај, са математичког становишта је могућ, међутим више у теорији него у пракси. Време израчунавања приватног кључа на основу јавног кључа, захтева време на просечном рачунару, које се мери у деценијама.

2.RSA алгоритам

RSA алгоритам је можда највише коришћени асиметрични криптографски алгоритам. Свакако је један од најпознатијих. Објављен је 1977. године од стране тима којег су чинили математичари Роналда Ривеста(енг. Ronald Rivest), Адија Шамира(енг. Adi Shamir) и Ленарда Адлемана(енг. Lenard Adleman). Име алгоритма чине прва слова презимена његових изумитеља. Алгоритам се заснива на простим бројевима и на тежини свођења великих бројева на производ простих бројева.

Трио математичара који је осмислио овај алгоритам, Ривест, Шамир, Адлеман, су добро позната и веома цењена имена у криптографском свету. Можда најзвучније име овог тима

је Ривест, професор на америчком универзитету MIT у Бостону, који је поред овог алгоритма осмислио и RC2, RC4, RC5, MD4, MD5.

Као то што обично бива у историји криптографије, успоставило се да је сличан криптографски систем осмишљен раније, али је дуго био под ознаком „Тајно“. Систем који је осмислио Клифорд Кокс(енг. Cliford Cocks), службеник енглеске Владе, који је дуго времена радио у одсеку за комуникације. Ознака тајности са његовог рада скинута је тек почетком 90-тих година прошлог века.

2.1 Опис алгоритма

За потребе генерисања приватног и јавног кључа, најпре се случајним путем бирају два велика проста броја, **p и q.** Мора се бирати тако да се производ ових бројева буде представљен жељеном прецизношћу(рецимо 2048-бита или 4096- бита али најчешће n се представља са 1024 бита или 309 децималних цифара).

Затим се уводи број **n,** који се дефинише као:

$$\boldsymbol{n = pq}$$

Следећи корак је производ Ојлерова Фи функција за вредности p и q. Ојлерова Фи функција даје број узајамно простих бројева са неким бројем а који су притом мањи од њега. Два броја су узајамно проста ако немају заједнички садржалац. Такође јасно је да за прост број, број узајамно простих бројева мањих од тог броја, је тај број минус 1. Када помножимо два проста броја, добијемо неки сложен број за кога није баш тако лако одредити колико број бројева који су узајамно прости са њим. Међутим, Ојлер је доказао да уколико помножимо два проста броја, Фи функција тог броја је производ Фи функција та два проста броја. Зато уводимо број **m**:

$$\boldsymbol{m = (p - 1)(q - 1)}$$

Број **m је Ојлерова Ф функција за број n.**

Сада се уводи још један број, са ознаком **e.** Законитост избора броја е је да је број узајамно прост са **m**. Број **m** је прост број и углавном свака имплементација овог алгоритма бира број **e = 65,537(2**16+1) или број 17 или број 3. Сматра се да је избор овог простог број довољно сложен а уједно и брз за израчунавање.**

Остаје само да се уведе још један број, **d** , а тај број преставља приватан кључ. Бира се на следећи начин, а тражи се проширеним Еуклидовим алгоритмом:

$$\text{пронаћи број } \boldsymbol{d}, \text{ такав да } \boldsymbol{de \bmod m = 1}$$

Дакле, број **d** је приватан кључ. Да би смо шифровали поруку једноставно узмемо поруку у отвореном тексту подигнемо је на степен е и одредимо остатак при дељењу са **m:**

$$\text{шифрована порука} = \text{порука}^{\text{e}} \ \boldsymbol{\bmod n}$$

Да би смо поруку дешифровали једноставно шифровани текст подигнемо на степен d и одредимо остатак при дељењу са n.

$$\text{порука} = \text{шифрована порука}^{\boldsymbol{d}} \ \boldsymbol{\bmod n}$$

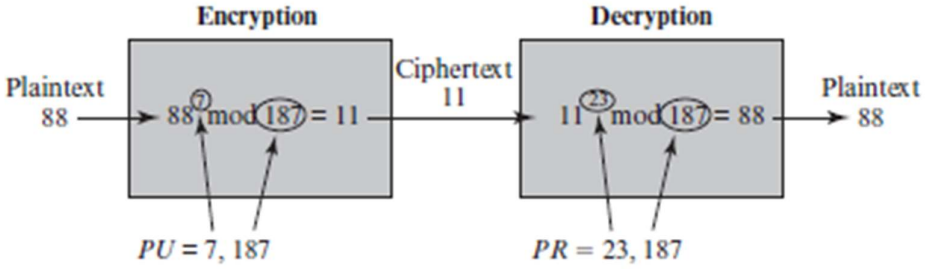
Како би што вернији приказали концепт рада алгоритма навешћемо један баналан пример са малим простим бројевима. Наравно, алгоритам RSA користи неупоредиво веће бројеве:

1. Узимао два различита проста броја **p = 61, q = 53.**
2. Рачунамо број **n = 61*53 = 3233.**
3. Рачунамо **Φ(n) = (p - 1)(q - 1) = 3120.**
4. Узимамо узајамно прост број e, 1< e <3120 , рецимо **e = 17**
5. Израчунавамо инверзну модуларну функцију за **d ,** па испада да је **d=2753**
6. Пратећи кораке јавни кључ је (**n=3233, e=17**). Функција шифровања је поруке **m** је $m^{17} (\text{mod } 3233)$.
7. Приватни кључ је онда (**n = 32333, d = 2753**). Функција дешифровања шифроване поруке **c**, $c^{2753} (\text{mod } 3233)$.

Поступак шифровања велике поруке се своди на следећи начин. Порука се подели у блокове. Бинарна вредност блока мора да буде мања од бинарне вредности броја n. То у пракси значи да величина блока у битовима буде већа или једнака log2(n)+1. Због тога се

овај алгоритам не користи да би шифровао хард-диск или рецимо неке велике улазне поруке.

На следећој слици показано је илустративно шифровање и дешифровање поруке за улазну поруку 88, приватни кључ (23,187) и јавни кључ (7,187).



Слика5. Рад RSA алгоритма кроз пример

2.2 Комплексност израчунавања

Постоје два аспекта са којих се овај проблем може посматрати: један је време потребно за генерисање кључева, други је време потребно за шифровање/дешифровање блокова поруке. Најпре се осврћемо на шифровање и дешифровање.

Шифровање и дешифровање неког броја своди се на подизање неког целог броја на степен неког другог целог броја и тражења остатка при дељењу са n. Пошто алгоритам користи велике бројеве, неке вредности између израчунавања биће толико велике да ће имплементација постати непрактична.

Стога, користе се неке законитости модуларне аритметике као што је следећа теорема:

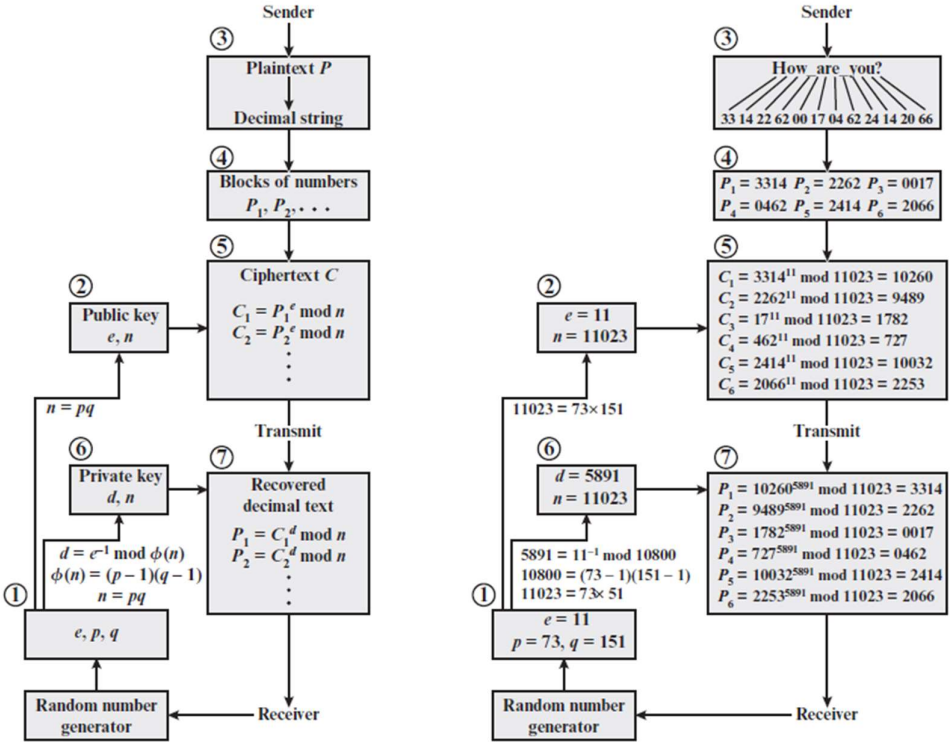
$$[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$$

На основу ове законитости можемо да рачунамо модуо међубројева и тиме убрзамо имплементацију.

Други принцип који убрзава шифровање и дешифровање је упрошћавање степеновања. Рецимо да желимо да израчунамо 16 степен броја x. Уколико то урадимо на најједноставнији начин, потребно је 15 множења бројева, односно:

$$\boldsymbol{x^{16} = x * x * x * x * x * x * x * x * x * x * x * x * x * x * x * x}$$

Међутим, израчунавање жељеног степена броја можемо остварити и са четири множења, ако у множење убацимо квадрате сваког парцијалног резултата, формирајући тако (x^2, x^4, x^8, x^{16}).



Сликаб. (а) општа секвенца шифровања/дешифровања (б) пример

Рецимо да желимо да израчунамо x^{11} mod n, где је x и n неки позитивни бројеви, то је $x^{1+2+8} = (x^1)(x^2)(x^8)$, затим израчунавамо x mod n, x^2 mod n, x^8 mod n, и њихов производ множимо и одређујемо остатак при дељењу са n.

У случају општих бројева, a^b mod n, где су a, b, n позитивни бројеви a број b је представљен бинарно као:

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

Тада се a^b може записати као:

$$a^b = a^{\left(\sum_{b_i \neq 0} 2^i\right)} = \prod_{b_i \neq 0} a^{(2^i)}$$

А $a^b \bmod n$ као:

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

Пре коришћења алгоритма RSA, корисник мора да генерише пар кључева. Генерисање кључева укључује одређене задатке:

- Одабир два проста броја p и q.
- Одабир или e или d и израчунавање другог.

Најпре се бирају прости бројеви p и q. Пошто је вредност n=pq позната свима, па и потенцијалним нападачима, да би се избегло израчунавање бројева p и q на основу броја n, прости бројеви се бирају из неког великог сета(бројеви p и q су велики прости бројеви). Са друге стране, одабир p и q мора да буде временски довољно ефикасан процес.

Све до сада није установљена техника за случајни одабир великих простих бројева, стога у одабиру мора се прибећи неким алтернативним методама. Та алтернативна метода је најчешће случајно бирање великог непарног броја и тестирање да ли је број прост. Ако није, сукцесивно се бира следећи и процедура се понавља. Тестови за испитивање да ли је број прост су тестови који се заснивају на вероватноћи. Један најпознатији тест који се најчешће користи за просте бројеве је Miller-Rabin-ов тест.

Пошто се долазак до бројева p и q нагађа, овај процес зна бити дуготрајан. Међутим, генерисање кључева се не користи често; једино кад је потребно генерисати нови пар кључева.

Теорија бројева тврди да се прост број у просеку може наћи на ln(N) корака од неког великог целог броја N. Уколико ову чињеницу искористимо при тражењу p и q, потребно је да у тестирамо ln(N) бројева, тачније ln(N)/2 пошто одбацујемо све парне бројеве. За пример узмимо неки број реда величине 2^{200}, дакле потребно је испитати $\ln(2^{200})/2 = 70$ случајева за проналазак.

За завршетак генерисања пара кључева остаје још само одређивање броја d, или e у зависности како се изабере. Они се одређују проширеним Еуклидовим алгоритмом. Процедура се своди на одабир серије случајних бројева, који се затим тестирају да ли су узајамно прости са Φ(n). Први број који се пронађе се користи. Ово је релативно брз процес, пошто се показало да је вероватноћа да су два случајна броја узајамно проста негде око 60%.

2.3 Сигурност и потенцијалне рањивости

Четири различитих напада на RSA алгоритам. То су:

- **Brute force** – испробавање свих могућих приватних кључева;
- **Математички напади** – покушај проналаска p и q на основу n;
- **Временски напади** – покушај да се нешто закључи на основу времена дешифровања;
- **Напади лажним порукама** – Ова врста напада користи неке мане RSA алгоритма.

Кад је у питању brute force напад, као што је већ речено сигурност се постиже коришћењем дугих кључева, тј. проширивањем броја битова приватног кључа (**d**). Међутим мора се одржавати баланс између броја битова кључа и времена извршавања алгоритма, тј. дужи кључ значи спорији систем.

Постоји више врста математичких напада на RSA:

1. Растурање n на два своја проста чинилаца. Ово омогућава израчунавање m односно Φ(n), а то значи релативно лако израчунавање приватног кључа.
2. Покушаја за директним израчунавањем Φ(n) без познавања p и q, а са Φ(n) и израчунавање приватног кључа.
3. Покушај директног израчунавања d, без познавања Φ(n).

Највише пажње везано за криптоанализу RSA посвећено је проналаску простих чиниоца броја n. Одређивање Φ(n) и растурање n на просте чиниоце су проблеми исте комплексности. Зато је време потребно за проналазак p и q користи као показатељ сигурности RSA алгоритма.

За велики број n и за велике просте чиниоце тог броја, проблем проналаска тих бројева је тежак, али не толико који је у време изума алгоритма био. Те 1977. године изумитељи алгоритма RSA објавили су поруку шифровану, 428-бит јавним кључем у престижном америчком часопису „Mathematical Games". Награда од 100$ следила је оном ко успе да дешифрује текст. Априла 1994. године група истраживача успела је да текст дешифрује, а алгоритам који су користили радио је 8 месеци. До данас, зна се да је успешно факторисан кључ од 768 бита тј. број од 232 цифре (2009. година). За факторизацију користе се математички алгоритми под називом GNFS и SNFS.

Number of Decimal Digits	Number of Bits	Date Achieved
100	332	April 1991
110	365	April 1992
120	398	June 1993
129	428	April 1994
130	431	April 1996
140	465	February 1999
155	512	August 1999
160	530	April 2003
174	576	December 2003
200	663	May 2005
193	640	November 2005
232	768	December 2009

Табела3. Напредак у проблему факторизације кроз време

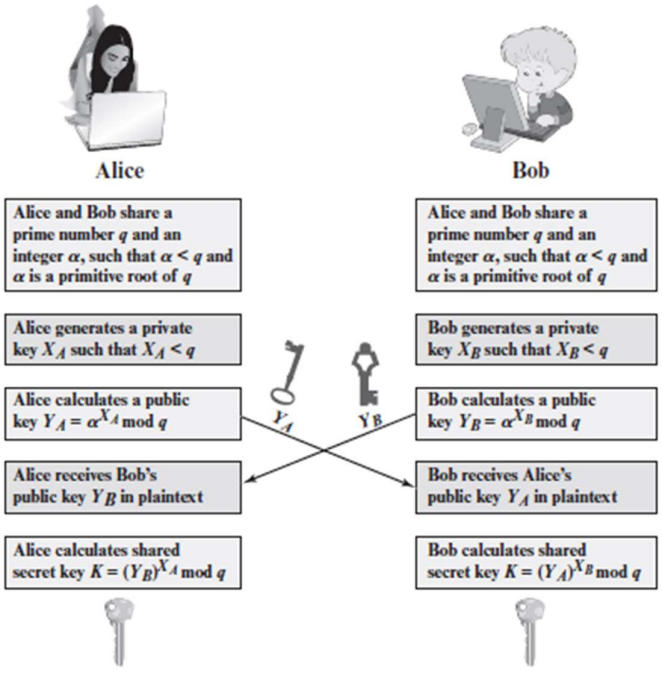
Напад који се темељи на времену потребном компјутеру да дешифрује неку поруку спада у групу напада на основу само шифрованог стекста(енг. chipertext only attack). Може се извући паралела временског напад и провалника који покушава да открије комбинацију сефа на основу времена потребног да власник сефа откључа исти тј. упише комбинацију.

3.Остали алгоритми

3.1 Дифије-Хелман алгоритам за размену кључева

Први објављени асиметрични криптографски алгоритам јавља се у заједничком раду Дифије- Хелман из 1976. године и познатије је као Дифије- Хелманов алгоритам размене кључева.

Сврха алгоритма је да омогући учесницима комуникације да сигурно размене тајни кључ, који ће се касније користити да симетрично шифрује поруке. Алгоритам спада у класу сигурне размене кључева, па се искључиво користи у те сврхе. Темељи се на израчунавању вредности дискретног алгоритма.



Слика7. Секвенца Дифије-Хелман размене кључева

ЛИТЕРАТУРА

[1] Cryptography and Network Security. Principles and Practice 7ed, Stallings W.,Pearson , 2016 година, 283.-308. страна.

[2] Modern Cryptography. Applied Mathematics, Easttom W, Springer, 2020.година, 227.-231. страна.