Глава 6

Криптография.

6.1 Цели, задачи и основни понятия.

Необходимостта и желанието дадена информация да бъде достъпна само за определен кръг от хора съпътстват човешката цивилизация от възникването ѝ, но дори до преди 30-40 години това бяха приоритети основно на военните и дипломатически служби. Днес нещата са твърде различни. Развитието на компютърните технологии и изграждането на глобална компютърна мрежа драстично промениха ситуацията. Правителствени, обществени и частни организации и фирми съхраняват или обменят информация за хора, продукти, услуги и процеси, които имат съществено влияние върху голям брой граждани. Нейното разкриване или неправомерно използване може да засегне интересите на огромни маси от хора с всички тежки социални последици от това. Всеки бизнес дори и най-легалния има своите тайни. Например при електронната търговия, продавачът не иска да попадне на измамник, а клиентът не желае всички да разберат какво си е купил или да бъде измамен. Всичко това превръща защитата на информация в приоритетна задача не само за правителствените органи, но и за редица обществени организации и частни икономически субекти. Приоритетна задача е разбира се и обществеият контрол върху средствата за защитата за да не се допусне използването им за осъществяване и прикриване на терористични и криминални деяния.

Блок-диаграмата на фиг.1 представя абстрактен модел на комуникационна система с наличие на опонент.

Опонентът (противникът) неправомерно включил се към обмена на информация между легитимните участници може да бъде

• *пасивен*, когато чрез подслушване на канала се опитва да определи ключовете (e, d) или поне да разкрие съобщението m . Пасивното следене на канала може да бъде използвано и за *анализ на трафика*, т.е за наблюдение и анализ на източника, получателя, времето за комуникация и количеството информация, която се обменя.

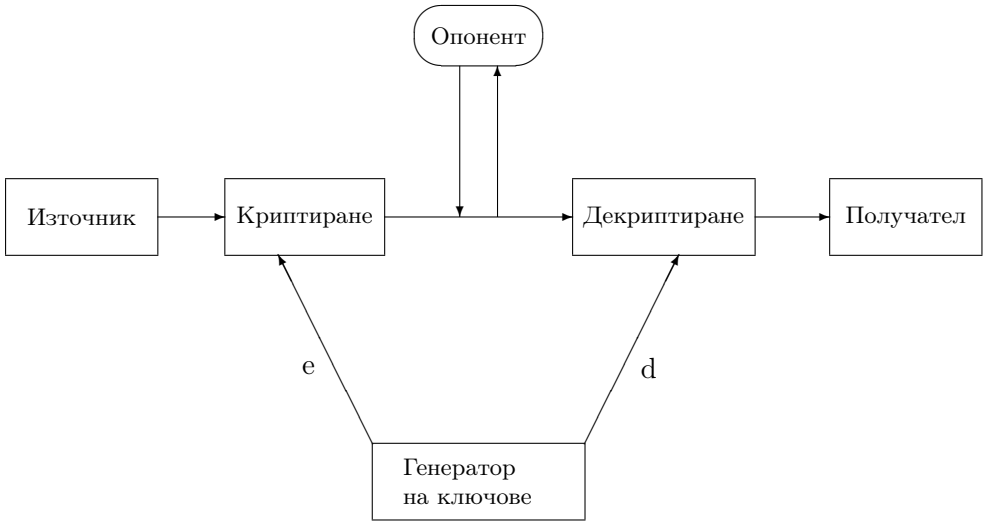
• *активен (tampering)*, когато използва твърде разнообразни форми на противодействие, например като

- блокиране на информационния поток;

- записване на прихванатата информация и излъчването ѝ по-късно, по време на фалшива комуникационна сесия;

- промяна съдържанието на информацията чрез заличаване, вмъкване и/или разместване

на части от нея и др. такива.



Фиг. 4.1. (e и d са съответно криптиращия и декриптиращия ключ)

Това ясно показва, че защитата на информацията включва много и най-разнообразни проблеми. Тя се осъществява на три нива:

• физическо (хардуерно) - преди всичко информацията трябва да бъде физически налична и достъпна. По принцип този кръг проблеми се решават с изграждане на отказоустойчиви устройства за съхранение и обработка на информация, комуникационни възли и софтуерни продукти за изграждане и управление на отказоустойчиви компютърни системи.

• защита при съхранение и предаване на информацията от подслушване, разрушаване или подправяне, както и от грешки породени от естествено зашумяване на комуникационните канали - криптографията и теория на кодирането са теоретичните основи за решаване на такъв кръг задачи. Към това ниво ще причислим и защитата на използваните операционни системи, макар че частично този проблем се отнася и към предното ниво.

• организационни и законодателни мерки, които подпомагат и регламентират горните нива и аспекти на защитата на данни.

Ето някои от задачите и целите спадащи към второто ниво, които трябва да бъдат постигнати с помощта на съвременната криптография:

• *Неприкосновеност и поверителност на информацията (privacy and confidentiality)*: Постигането на тази цел осигурява информацията да остане тайна и недостъпна за всички, освен за тези, които са оторизирани да си служат с нея.

• Осигуряване *цялост на данните (data integrity)*, с което се цели изпратената информацията да не бъде подправяна или подменена в процеса на предаване или обработка.

• *Идентификация (identification or entity authentication)*: Задачата е легално включените в обмена на информация страни (лица, компютърни терминали, кредитни карти и др.)

да се убедят взаимно, че наистина комуникират една с друга, т.е. че отсрещната страна е наистина тази, за която се представя.

• *Автентичност на съобщението (message authentication)*: Получателят на съобщението трябва да се убеди дали подателят му е наистина този, който е заявен в съобщението като такъв. Тази задача е известна още като data origin authentication.

• Осъществяване на *електронен подпис (signature)*: Целта е реализацията на електронен аналог на класическия подпис върху писмени документи. В електронния вариант получателят на даден електронен документ също трябва да е в състояние да убеди трета независима страна ("съдия"), че той действително е изпратен и в точно същия вид от страната, която го е подписала.

• *Неотменимост (неотричаемост) (nonrepudiation)*: Осигурява, че никоя от страните в комуникационния процес не може да отрече реално осъществени свои действия (например, че е изпратила дадено съобщение) и/или поети ангажименти. Това е особено важна цел както от правна така и от комерсиална гледна точка.

• *Управление на ключовете (key management)*: фактор от жизнено значение за сигурността. Управлението на ключовете включва всички аспекти на боравенето с тях, като се започне от създаването им до евентуалното им унищожаване. Най-голямата сложност е при съхраняването и разпространението на ключовете. Тези проблеми се решават отново с криптографски алгоритми.

• *Разпределение на секрета (sharing schemes)*: В ситуация на многостранно сътрудничество дадено свойство (съглашение) действа докато броят на противниците му в групата не надвиши определен праг. Такъв тип задачи възникват при съхранение и управление на ключовете.

• *Анонимност (anonymity)*: Да се прикрие идентичността на определен участник в даден процес.

• *Авторизация (authorization)*: задача, която трябва да осигури прехвърляне правото на друга страна да бъде или да извърши нещо.

• *Контрол на достъпа (access control)*: Да ограничи в определени рамки правата и възможностите, които легитимен ползвател на една компютърна система или мрежа има.

• *Едновременен обмен (simultaneous exchange)*: В ситуация на много участници някакво желано свойство (качество) е в сила, докато нещо друго ценно (например подписа на отсрешната страна) не се получи.

Криптографията, която осигурява методите и средствата за защита заедно с *Криптоанализа*, който има за задача разбиването на шифрите формират науката *Криптология*.

Всеки механизъм, процес или структура опиращи се на достиженията на съвременната криптография привличат и използват нейните базисни средства (методи, алгоритми и др.), наричани най-общо *криптографски примитиви*. Естествено, възниква въпроса за оценка на тяхната ефиктивност и полезност. Формулирани са редица практически и математически критерии за оценка на криптографски примитиви. Последните се основават на теорията на вероятностите и теорията на сложността на алгоритми, а по-долу излагаме критерии за оценка от практическа гледна точка.

Още в 1883 г. Керкхоф формулира съвкупност от изисквания за една криптосистема, наричани **Kerckhoffs' desiderata**:

1. да бъде, ако не теоретически, то поне практически неразбиваема;

2. компрометирането на един неин детаил да не създава проблеми на кореспондиращите;

3. ключът трябва да се запомня лесно и без да се записва както и лесно да се сменя;

4. криптограмите трябва да могат да се изпращат с телеграф (това е "върха" на комуникациите тогава);

5. апаратурата за криптиране трябва да бъде портативна и да позволява само един човек да работи с нея;

6. системата трябва да бъде лесна, да не изисква нито знанието на голямо количество правила, нито умствено пренапрежение.

Тези препоръки са актуални и днес, а ето и някои съвременни практически критерии:

• ниво на секретност.

Определя се от работния фактор (т.е. необходимите време и изчислителски ресурси) за компрометиране на криптографския механизъм при най-добрата известна в момента на оценка атака, както и предполагаемите в близко бъдеще развитие на компютърните технологии и методи за криптоанализ.

• функционалност.

Как се съгласува с други примитиви при изграждане на механизми, каква и колко съществена е ролята му. За решаване на какви криптографски задачи даденият примитив е по подходящ и ефективен.

• методи за опериране.

Оценява се поведението по отношение на сигурността при различни начини на прилагане, свързване с други примитиви, вида на входните данни и др.

• производителност и бързина.

Например скоростта на криптиране измерена в бит/сек. Към критериите от този тип спадат, например, необходимото време да се създаде електронен подпис (ЕП), времето за проверката му, времето, необходимо за генериране на основните параметри на системата и ключовете. Оценяването винаги се прави за фиксирана форма на опериране.

• внедримост.

Оценява каква е сложността при хардуерна и/или софтуерна реализация на съответния криптографски механизъм. Небходимите ресурси (дисково пространство, памет и др) за съхраняване на ЕП както и всички данни и/или резултати получени по време на създаване на електрония подпис, издаване на сертификат и др.

Интересна от практическа гледна точка представлява така наречената "Ad hoc security". При нея се прави оценка на ресурсите на потенциалния противник и те се сравняват с ресурсите необходими за разбиване на криптосистемата.

Дефиниция 6.1.1 *Под* ***криптосистема*** *се разбира съвкупността от три множества* $\mathbf{M}$, $\mathbf{C}$ *и* $\mathbf{K}$ *от редици, съответно над азбуки* $\boldsymbol{A_1}$, $\boldsymbol{A_2}$ *и* $\boldsymbol{A_3}$, *и двойка* $(\boldsymbol{E}, \boldsymbol{D})$ *множества от изображения* $\boldsymbol{E} = \{\mathcal{E}_e : \mathbf{M} \to \mathbf{C} \mid e \in K\}$ *и* $\boldsymbol{D} = \{\mathcal{D}_d : \mathbf{C} \to \mathbf{M} \mid d \in K\}$, *които притежават свойството, че за всяко* $e \in K$ *(криптиращ ключ) съществува единствен* $d \in K$ *(декриптиращ ключ) така, че*

$$\mathcal{D}_d(\mathcal{E}_e(m)) = m, \quad \textit{за всяко} \quad m \in \mathbf{M}.$$

Множествата $\mathbf{M}$, $\mathbf{C}$ *и* $\mathbf{K}$ *се наричат съответно* ***пространство от съобщенията ((plaintext) message space), пространство от шифротекста (ciphertext space)***

и ***пространство от ключовете (key space)****, а елементите на* **E** *и* **D** ***криптиращи*** *и* ***декриптиращи*** *трансформации.*

Забележка 6.1 *Понятието криптографски алгоритъм в литературата се употребява в широк смисъл като синоним на криптосистема, а в тесен за означаване на криптиращата и декриптираща трансформации.*

Криптосистемите се делят на две групи:

• ***Симетрични***: $e = d$ или "изчислително лесно" може да се извлече d от e.

Примери на такива криптосистеми са AES, DES, RC4, Stream ciphers и др.

• ***Асиметрични*** (наричани още *двуключови, с публичен ключ)*:

когато е "изчислително невъзможно" да се определи d, знаейки само e. Ключът e използван за шифриране може да бъде направен публично известен докато този за дешифриране d трябва да се държи в тайна.

Примери на такива криптосистеми са RSA, Elliptic curve cryptography (ECC), тези на Rabin, ElGamal и McEliece.

Понятието "изчислително невъзможно" трябва да се разглежда в контекста на Теория на сложността на алгоритми (theory of complexity). "Невъзможно/лесно" означава, че не съществува/съществува алгоритъм, който дава решение на проблема за време, зависещо полиномиално от параметъра (параметрите) на проблема. Когато такъв алгоритъм не съществува, се казва още, че задачата е в класа **NP**. Прецизирането на тези понятия излиза извън рамките на тези лекции и ние ще се опрем на интуитивния им смисъл. На читатели, които искат да се запознаят по-подробно с горните понятия и съпътстващите ги проблеми препоръчваме да прочетат някоя книга по теория на сложността на алгоритми (например [**?**]).

Дефиниция 6.1.2 *Нека X и Y са произволни множество.* ***Еднопосочна функция (one-way function)*** *се нарича функция $f : X \to Y$, такава че е "лесно" да се пресметне $f(x)$ за всяко $x \in X$, докато за случайно $y \in Im(f)$ е "изчислително невъзможно" да се намери x, такова че $f(x) = y$.*

Пример 6.1.1 Нека g е примитивен корен по модул простото число p. Функцията $f : Z_p^* \to Z_p^*$ дефинирана с

$$f(x) = g^x \pmod{p}$$

е пример за еднопосочна функция.

Изчисляването на $f(x)$ не представлява трудност, но намирането на $\operatorname{ind} a$, за произволно $a \in \mathbb{Z}_p^*$, дори за не големи p, вече създава проблеми. Както отбелязахме по-горе в настоящото изложение няма да разглеждаме сложността на тази операция, но препоръчваме на читателя да направи таблица с индексите относно простото число $p = 53993$ (твърде малко от криптографска гледна точка), за да добие представа за сложността на проблема.

При системите с публичен ключ се използва най-често следният подклас от еднопосочни функции.

Дефиниция 6.1.3 ***Еднопосочна функция със секрет (Trapdoor one-way)*** *се нарича еднопосочна функция $f : X \to Y$ със свойството, че при зададена допълнителна (тайна) информация (trapdoor information) за всяко $y \in Im(f)$ става "възможно" да се намери x, така че $f(x) = y$.*

6.2 Криптографски примитиви и механизми.

Основните криптографски примитиви, използвани при изграждане на системи за защита на информацията са следните:

- симетрични криптографски алгоритми
- асимитрични криптографски алгоритми
- управление създаване, съхранение и разпределение на ключове
- хеш-функции
- генератори на случайни числа

В частност те са в основата на механизмите за създаване на електронен подпис и генерирането на частните и публичните ключове.

По-долу ще разгледаме малко по-подробно първите три, а сега да дефинираме и посочим ролята на последните две.

Дефиниция 6.2.1 ***Хеш-функция (hash function)*** *се нарича еднопосочна функция h, която изобразява редица (от битове) с произволна дължина в редица с фиксирана дължина n (например, $n = 64, 128, 160$) и притежава свойството, че е свободна от колизии, т.е. изчислителски невъзможно е да се намерят две различни съобщения m и m', така че $h(m) = h(m')$.*

Примери на хеш-функции са MD4, MD5, SHA-1, RIPEMD-160. Препоръчително е да се използват хеш-функции с $n = 128$ или 160 бита, за да са свободни от колизии и същевременно n да не е твърде голямо, за да не се забавя обработката на информацията. Но поради увеличаване производителността на компютрите в крайна фаза на стандартизация е дори SHA-512 ($n = 512$).

Основните приложения на хеш-функциите са за проверка цялостта на данните и създаване на кратка добавка, която да бъде подписана при реализация на електронен подпис, както и при някои идентификационни протоколи.

Важен клас криптографски примитиви са и ***генераторите на случайни числа***. Те стоят в основата на програмите за пораждане на ключове и тяхното качество е от съществено значение за пораждането на добри неподдаващи се на отгатване ключове.

Под *случайна поредица (низ) от битове* ще разбираме такава редица от 0 и 1, за която знанието на произволно подмножество от елементите ѝ не дава никаква информация за останалите битове. С всяка такава редица може да се свърже случайно число, като се вземе числото, чиито запис в двоична бройна система е тази редица. Под *генератор на случайна двоична редица (случайно число)* разбираме устройство и/или алгоритъм, който поражда случаен низ от битове.

Всеки генератор на случайна редица изисква естествен източник на шум (например, физическо устройство с топлинен шум), който може впоследствие да бъде обработен криптографски (например, подложен на хеш-функция).

Всеки случаен генератор се подлага на най-разнообразни тестове (чиито резултат се дава най-често с вероятността редицата да е случайна) доколко случайни редици произвежда. Най-простите примери за такива тестове са проверка за равенство между броя на единиците и нулите, и проверка за автокорелационните свойства на генерираната редица. Един универсален тест (Maurer) е степента на компресиране на редицата. Колкото по-малко може да се компресира, толкова по-случайна е тя.

Когато се изискват случайни редици с много голяма дължина, се използват така наречените псевдослучайни редици. *Генератор на псевдослучайна редица* наричаме детерминиран алгоритъм, който от напълно случаен низ от k бита поражда двоична редица с дължина на порядъци по-голяма от k, която с голяма вероятност издържа статистически тестове за случайност. Доколкото изходът на такъв генератор е еднозначно определен по началната редица, k трябва да бъде достатъчно голямо, за да не може с пълно изчерпване да се генерират всички възможни псевдослучайни редици.

6.2.1 Симетрични криптосистеми.

Предимства:

- Голяма скорост на криптиране и декриптиране както при хардуерна, така и при софтуерна реализация.
- Лесни за реализация.
- Използват се като съставни части на най-разнообразни криптографски примитиви.

Недостатъци:

- Секретният ключ е само един и всяка от страните, участващи в процеса, може да го компрометира случайно или целенасочено.
- При комуникация по двойки в мрежа от n участника са необходими $n(n-1)/2$ ключа.
- Изисква често смяна на ключа (при всяка сесия), което поражда проблеми с разпространението на ключовете.
- Не са удобни за използване в механизми за електронен подпис, защото изискват много големи ключове за проверяващата трансформация и въвличане на Трета доверена страна (TTP).

В $ 5.2.3 е описан един механизъм (използващ Теория на числата) как да се реши проблемът с управлението на ключовете за симетрични криптосистеми.

Най-употребяваните симетрични криптосистеми са **блоковите криптосистеми (block ciphers)**.

Дефиниция 6.2.2 *Криптосистема, чието пространство от съобщенията* $\mathbf{M}$ *се състои от редици* $m = (m_1, \ldots, m_n)$ *с дължина* n*, т.е. разделя открития текст на блокове от по* n *символа на азбуката, се нарича* ***блоков шифър с дължина*** n.

символ	число	символ	число	символ	число	символ	число	символ	число
0	00	й	20	я	40	У	60	V	80
1	01	к	21	А	41	Ф	61	*	81
2	02	л	22	Б	42	Х	62	=	82
3	03	м	23	В	43	Ц	63	+	83
4	04	н	24	Г	44	Ч	64	/	84
5	05	о	25	Д	45	Ш	65	^	85
6	06	п	26	Е	46	Щ	66	!	86
7	07	р	27	Ж	47	Ъ	67	?	87
8	08	с	28	З	48	Ь	68	$	88
9	09	т	29	И	49	Ю	69		
интервал	10	у	30	Й	50	Я	70		
а	11	ф	31	К	51	,	71		
б	12	х	32	Л	52	.	72		
в	13	ц	33	М	53	-	73		
г	14	ч	34	Н	54	:	74		
д	15	ш	35	О	55	;	75		
е	16	щ	36	П	56	"	76		
ж	17	ъ	37	Р	57	(	77		
з	18	ь	38	С	58	)	78		
и	19	ю	39	Т	59	I	79		

Таблица 5.1.

Повечето добре известни симетрични криптосистеми са блокови. Такива са и следните класически криптосистеми:

Заместващи шифри (Substitution ciphers)

Дефиниция 6.2.3 *Критосистема, която попада в една от следващите категории, се нарича* ***заместващ шифър (субституция)****:*

Проста субституция *Нека е даден блокова криптосистема с дължина n и пространство от ключовете $\mathbf{K}$, състоящо се от пермутации на азбуката. Криптирането с ключа $e \in \mathbf{K}$ се извършва по правилото*

$$\mathcal{E}_e(m) = (e(m_1), e(m_2), \ldots, e(m_n)).$$

Хомофоник (homophonic) *Всеки символ от азбуката се изобразява $a \longrightarrow H(a)$ в множество от редици с дължина t, като $H(a) \cap H(b) = \emptyset$, за $a \neq b$.*

Многоазбукови (polyalphabetic) *За разлика от простата субституция, ключовете $e = (\pi_1, \ldots, \pi_n)$ представляват наредени $n-$орки от пермутации на азбуката и*

$$\mathcal{E}_e(m) = (\pi_1(m_1), \pi_2(m_2), \ldots, \pi_n(m_n)).$$

Разместващи (пермутационни) шифри (Transposition ciphers)

Дефиниция 6.2.4 *Нека $\mathbf{M}$ е пространството от съобщения на блоков шифър с дължина n. Криптосистемата се нарича* ***разместваща (пермутационна)****, ако всеки ключ е еднозначно определя пермутация $\pi \in S_n$ на $\{1, 2, \ldots, n\}$, такава че*

$$\mathcal{E}_e(m) = (m_{\pi(1)}, \ldots, m_{\pi(n)}).$$

Използваните днес блокови шифри принадлежат на класа от така наречените ***итеративни шифри от тип на Фейстел (Feistel)***, но тази тематика остава в страни от целите на нашето изложение.

Друг много използван тип симетрични криптосистеми са ***поточните шифри (stream ciphers)***. При тях криптирането представлява прибавяне по модул 2 на бинарна редица към бинарния запис на открития текст.

Ще илюстрираме заместващите шифри с една криптосистема основана на афинната трансформация

$$\begin{pmatrix} x \\ y \end{pmatrix} \Longrightarrow \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

В нашия случай ще изберем

$$\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 2 & 13 \\ 17 & 22 \end{pmatrix} \quad \text{и} \quad \mathbf{B} = \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} -7 \\ 5 \end{pmatrix},$$

които ще представляват ключа. На елементите на матриците ще гледаме като на елементи на $\mathbb{Z}_{89}$, т.е. действията ще ги извършваме по модул 89. Съответствието, при което на буквите и другите текстови символи се съпоставят числа по модул 89 се задава с Таблица 5.1.

Да криптираме текста: “Теория на числата”. На него му съответства редицата от двуцифрени числа 59 16 25 27 19 40 10 24 11 10 34 19 28 22 11 29 11 10. Накрая допълваме текста с интервал (10) за да станат числата четен брой. Първите две числа се преобразуват в

$$\begin{pmatrix} 2 & 13 \\ 17 & 22 \end{pmatrix} \begin{pmatrix} 59 \\ 16 \end{pmatrix} + \begin{pmatrix} -7 \\ 5 \end{pmatrix} = \begin{pmatrix} 52 \\ 25 \end{pmatrix}$$

Извършвайки последователно пресмятанията получаваме 52 25 38 45 17 51 58 80 56 56 41 22 68 75 36 29 56 56 Следователно шифротекстът е

ЛоьДжКСVППАлЬ;щтПП

Декриптирането се извършва като се приложи обратната трансформация:

$$\begin{pmatrix} x \\ y \end{pmatrix} \Longrightarrow \mathbf{A^{-1}} \begin{pmatrix} x \\ y \end{pmatrix} - \mathbf{A^{-1}B} = \begin{pmatrix} 22 & -13 \\ -17 & 2 \end{pmatrix} \left[\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} -7 \\ 5 \end{pmatrix} \right]$$

Разгледаният тип криптосистеми са предложени в 1931 от Хил (Hill). Поради линейността си те нямат криптографска стойност днес, макар че все още се използват (алгоритми използващи същия принцип) в някои комуникационни системи за да увеличат ентропията на предаваната информация.

6.2.2 Асиметрични криптосистеми.

Предимства:

- Двойката ключове (частен и публичен) могат да бъдат ползвани многократно и през дълъг период (дори няколко години).

- В контраст на симетричния случай, при мрежа от много участници са необходими толкова двойки ключове, колкото са участниците.

- Позволяват създаване на ефективни схеми за електронен подпис с проверяваща функция, изискваща ключ с доста по-малки размери от съответните, построени на симетрични алгоритми.

Недостатъци:

- Относително по-бавни в сравнение със симетричните.

- При използване за криптиране ключът им е много по-дълъг от този на симетричните.

- Дължината на добавения електронен подпис е относително голяма.

Към асиметричните криптосистеми спадат RSA, RSA(Rabin-Williams modification), ElGamal, Elliptic Curves Cryptography и др.

Криптосистема RSA:
Носи името на създателите си: Rivest, Shamir, Adleman. Описанието ѝ може да се намери в почти всички книги по криптография (виж [**?**]) както и в стандартите PKCS #1 [**?**], ISO 11166 [**?**, **?**]. Сигурността на RSA се основава на трудността да се разложи едно естествено число n на прости множители. За достатъчно големи n съвременните математически методи и компютърна техника не могат да се справят с тази задача.

Нека $n = pq$ е произведение на две големи прости числа и e, $1 < e < \varphi(n)$, е случайно число, такова че $(e, \varphi(n)) = 1$ ($\varphi(n)$ е функцията на Ойлер от n). RSA ползва два ключа:

ПУБЛИЧЕН КЛЮЧ: (n, e), състоящ се от модул n и експонента e;
ЧАСТЕН КЛЮЧ: d, $1 < d < \varphi(n)$, такова че $ed \equiv 1 \pmod{\varphi(n)}$.
КРИПТИРАНЕ: Съобщението се превръща в поредица от числа $m_1, m_2, \ldots$, всяко от които се криптира по правилото:

$$m_i \longrightarrow c_i \equiv m_i^e \pmod{n}.$$

Поредицата $c_1, c_2, \ldots$ представлява шифротекста.
ДЕКРИПТИРАНЕ: Откритият текст се получава с

$$m_i \equiv c_i^d \pmod{n}.$$

Наистина $c_i^d \equiv m_i^{ed} \pmod{n}$. Но от $ed \equiv 1 \pmod{(p-1)}$ и теоремата на Ферма следва, че $m_i^{ed} \equiv m_i \pmod{p}$. Аналогично $m_i^{ed} \equiv m_i \pmod{q}$. Тъй като p и q са различни прости числа, то $m_i^{ed} \equiv m_i \pmod{pq}$.

Криптосистемата RSA се използва най-вече за създаване на електронен подпис, при разпространение и съхраняване на ключове за симетрични алгоритми и за аутентификация. Понастоящем модулът n трябва да бъде число поне от 768 бита (в двоичен запис). При приложения, които изискват по-дълъг срок на надежност (месеци или години, както е при електронния подпис), ***трябва да се ползват модули с дължина поне 1024 бита***. Числата p и q също трябва да удовлетворяват някои ограничения, които са описани в стандартизационните документи.

Експонентата и частният ключ не е препоръчително да бъдат много малки, защото при определени обстоятелства може да се разкрие съобщението m. Има

реализации, при които $e = 3$, но в тези случаи, особено за малки m, сигурността е слаба. В тези случаи към съобщението се добавя случайна информация, за да се избегне малко m. Ако вместо малки експоненти се вземат такива, които имат малко единици в двоичното им представяне, криптирането също ще изисква по-малко ресурси.

Вариант на RSA е криптосистемата на Rabin-Williams. Сигурността ѝ се основава на трудността на разлагане на прости множители и на намирането квадратен корен по модул съставно n.

Пример 6.2.1 Нека $p = 73$ и $q = 109$. Тогава $n = pq = 7957$, а $\varphi(n) = 7776$. Да изберем за експонента $e = 17$. Публичния ключ ще бъде $(7776, 17)$. Тъй като $17 \cdot 7489 \equiv 1 \pmod{7776}$, то частния ключ е $d = 7489$.
Да криптираме съобщението "Кой е Ойлер?". За целта всеки два последователни текстови символа ще разглеждаме като двузначни числа в 89-ична бройна система, т.е. от Ко|й |е |Ой|ле|р? получаваме $51 \cdot 89 + 25 \mid 20 \cdot 89 + 10 \mid \ldots \mid 27 \cdot 89 + 87$. Следователно на нашия текст съответства съвкупност от 6 числа: 4564 | 1790 | 1434 | 4915 | 1974 | 2490. Сега всяко от тези числа трябва да подигнем на степен 17 по модул 7957. Да забележим, че $a^{17} = (((a^2)^2)^2)^2 \cdot a$, т.е. повдигането може да извършим с последователни повдигания в квадрат и едно умножение. След изпълнение на операциите получаваме

$$4564^{17} \equiv 7859 \pmod{7957},\ 1790^{17} \equiv 3552 \pmod{7957},\ \ldots\ ,\ 2490^{17} \equiv 2911 \pmod{7957}$$

И така шифротекстът е набора от шесте числа

$$7859 \mid 3552 \mid 1885 \mid 2339 \mid 1557 \mid 2911.$$

Ако всяко от тях представим в 89-ична бройна система и използваме Таблица 5.1 ще получим текстови запис на шифротекста:

$$\text{р\$} \mid \text{*ю} \mid \ \text{ек} \mid \text{оп} \mid \text{Гж} \mid \text{Цх}$$

Трябва да отбележим, че $88 \cdot 89 + 88 = 7920 < 7957$, то след повдигане в степен по модул n може да се получи число, което е с 3 цифри в 89-ична бройна система. Затова, ако шифротекстът ще се преобразува в текстова форма, то някакво разделяне (както горе) трябва да се направи. Тази възможност също създава неудобства при използване на RSA. Затова обикновено се ползва за електронен подпис или криптиране на кратки съобщения (например парола или ключ за симетрична криптосистема), така че да не се налага разбиване на текста на блокове.

Криптосистема на ElGamal:
Вариант на тази ситема се използва като асиметричен алгоритъм в механизма за електронен подпис DSA. Сигурността ѝ се основава върху трудността на намиране на дискретен логаритъм в $Z_p^\star$. Публичният ключ на даден потребител U представлява тройка (p, g, E), където g е примитивен корен по модул p и $E = g^a$. Частният ключ е a, $1 \le a \le p - 2$. Ако група ползватели имат едни и същи p и g, то публичният ключ, който се записва в регистъра, е само $E = g^a$. Процедурата по криптирането е следната: Потребител S, който иска да изпрати съобщение m на U избира произволно число k и изпраща двойката $(g^k,\ mE^k)$. За да декриптира съобщениет U изчислява $D = (g^k)^a = E^k$ и след това намира $mE^kD^{-1} \equiv mE^kE^{-k} \equiv m \pmod{p}$.

Простото число p трябва да е с дължина поне 768 бита, като за по-дългосрочна сигурност се препоръчва дължина поне 1024 бита. Като недостатък на криптосистемата на ElGamal може да се посочи, че шифротекста е два пъти по-дълъг от откритият текст.

При варианта, фиксиран в DSA (FIPS 186), цикличната група не е $Z_p^\star$, а нейна подгрупа от ред простото число q с дължина 160 бита и делящо $p-1$. Затова публичният ключ включва и q.

Криптосистемата на ElGamal естествено се обобщава като $Z_p^\star$ се замени с произволна крайна циклична група от достатъчно голям ред. Частен случай на обобщената криптосистема на ElGamal е криптоситемата, основана на елиптични криви.

6.2.3 Управление на ключове. Протокол на Diffie-Hellman.

Управлението на ключовете включва тяхното създаване, съхранение, разпределение и унищожаване. Пропуски в предаването и съхранението на ключа стават най-често причина за компроментиране на дадена система за защита. Описаният протокол представлява един математически подход при решаването на този тип проблеми в защитата на информация.

Протокол на Diffie-Hellman (DH): Нека **A** и **B** са двама участника (например терминални устройства или потребители) в комуникационен процес, които трябва да договорят ключ за предстоящата им комуникационна сесия. На **A** и **B** са известни (обикновено те са публично известни или са заложени във всяко устройство) базисните параметри на протокола, които са постоянни за всички сесии. Това са:

- голямо просто число p - поне 512 бита, обикновено с дължина 1024 бита, и
- цяло число $g:\ 1 < g < p-1$, за което $g^k \not\equiv 1 \pmod p$ за всяко $k < p-1$.

Протоколът се състои от следните стъпки:

1. **A** генерира случайно число $a : 1 < a < p-1$, и изпраща на **B** числото $x \equiv g^a \pmod p$.
2. **B** генерира случайно число $b : 1 < b < p-1$, и изпраща на **A** числото $y \equiv g^b \pmod p$.
3. **A** изчислява $K \equiv y^a \equiv g^{ab} \pmod p$.
4. **B** изчислява $K \equiv x^b \equiv g^{ab} \pmod p$.
5. Всеки от **A** и **B** прилагат хеш-функция H към (двоичния запис на) K и намират $S = H(K)$.

Получената бинарна поредица S е сесийния ключ. Случайните числа a и b се пазят в тайна от **A** и B, съответно, и се унищожават веднага след намирането на ключа или поне в края на сесията. Не трябва да съществува възможност те да бъдат изпратени или извлечени от устройството, което пресмята S. Ключат S също се унищожава веднага след приключване на сесията.

С цел да се намали времето за изчисление обикновено се реализира модифициран вариант на DH. Модулът се избира от вида $p = 2tq + 1$, където q е също просто число с дължина 160 бита (трябва да е поне 128 бита), а за g се взема число с показател q по модул p, т.е. $g = \alpha^{2t}$, където α е примитивен корен по модул p. За да се избегнат някои атаки t може да се избере също просто.

6.3 Електронен подпис.

Всеки механизъм за електронен подпис (МСЕП) включва *алгоритъм за генериране на подписа* и *проверяващ автентичността му алгоритъм.* Алгоритъмът за генериране на подписа, заедно с методите за привеждане на съобщението в подходящ за подписване вид, формират *процеса на цифрово (електронно) подписване*, а проверяващият алгоритъм заедно със средствата за разкриване на съобщението - *процеса на проверка на цифровия подпис.*

Формализираното описание на механизма, по който един участник U в компютърна комуникационна система подписва своите съобщения (електронни изявления) включва:

• пространство от съобщенията $\mathbf{M}$ и пространство от *подписите* $\mathbf{S}$

• трансформация $\mathcal{S}_U : \mathbf{M} \to \mathbf{S}$, която е известна само на U

• проверяваща трансформация $\mathcal{V}_U : \mathbf{M} \times \mathbf{S} \to \{true, false\}$,

като са в сила следните свойства:

• $s \in \mathbf{S}$ е валиден подпис тогава и само тогава, когато $\mathcal{V}_U(m, s) = true$

• изчислително невъзможно е за всеки освен U да намери за всяко $m \in \mathbf{M}$ такова $s \in \mathbf{S}$, че $\mathcal{V}_U(m, s) = true$.

Подписване:

1. Пресмятане на $s = \mathcal{S}_U(m)$
2. Изпращане на (m, s).

Процедура за проверка:

1. Получаване на $\mathcal{V}_U$ (от U или от регистър)
2. Пресмяне на $u = \mathcal{V}_U(m, s)$
3. Приемане на подписа при $u = true$ и отхвърляне в противния случай.

На практика МСЕП трябва да притежава следните свойства:

- лесно да се създава ЕП, т.е. $\mathcal{S}_U$ да бъде достатъчно бърза и удобна за прилагане;

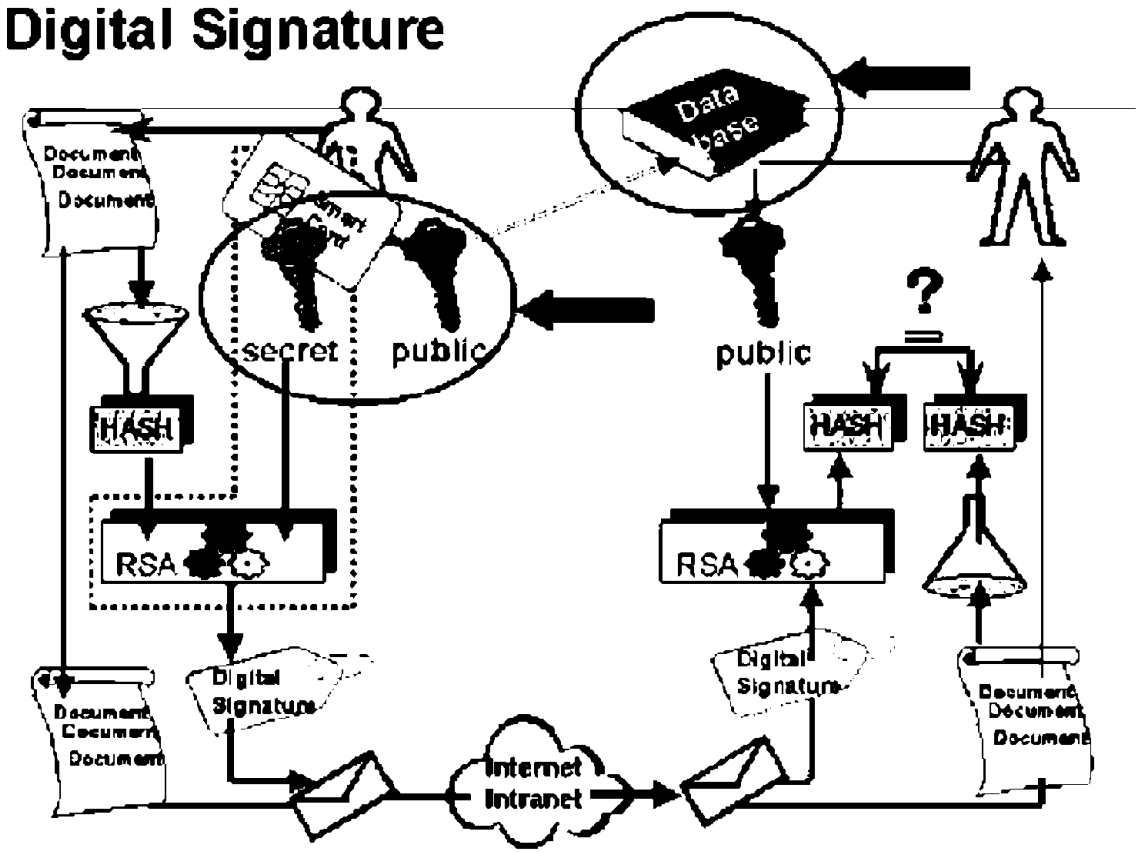
- лесна проверка на ЕП, т.е. $\mathcal{V}_U$ да бъде достатъчно бърза и удобна за прилагане;

- механизмът да остава сигурен поне за периода, през който подписа е валиден, т.е. времето, необходимо за разбиване на системата, да надхвърля този период. Понятието "сигурен" включва практическата невъзможност друг освен автора на подписа да създаде смислени съобщения, които подложени заедно с подписа на проверяващата функция $\mathcal{V}_U$, да дават резултат $true$.

Трябва изрично да отбележим, че необходимостта от електронен подпис предполага две взаимно подозиращи се и/или готови да го оспорят страни. Затова не е възможно използването на общ ключ при реализацията на $\mathcal{S}_U$ и проверяващата трансформация $\mathcal{V}_U$. Това предопределя използването на асиметрични криптографски алгоритми (както е показано във Фиг. 5.1) с два ключа - частен и публичен. В литературата съществуват описания на МСЕП на базата на симетричен алгоритъм, но те изискват въвличане на ТТР в процесите на генериране и проверка на подписа и то в твърде тромави процедури. Затова тези схеми не са и залегнали в международните документи.

Трансформацията $\mathcal{S}_U$ най-често е сложна функция, представляваща композиция (последователно прилагане) на хеш-функция h и криптографска трансформация $\mathcal{E}$. Фигура 5.1 изобразява блоковата структура на механизъм за електронен подпис с такава трансформация $\mathcal{S}_U$. При него съобщението се предава в явен вид. Възможни са, обаче, и реализации, при които то е зашифровано. Подходящи за тази цел са симетричните алгоритми, защото са по-бързи и удобни за реализация. Освен това засекретяване на текста на съобщението

предполага желание за това от двете страни (или нормативни изисквания), което предполага и сътрудничество между страните за грижливо съхранение на общия ключ.



Фиг. 5.1 Блокова схема на система за електронен подпис

Вторият тип механизм за електронен подпис, който предполага извличане на съобщението от самия подпис, се използва само за много кратки съобщения (парола, ключ, идентификационен номер), тъй като прилагането на асиметричен алгоритъм за криптиране на дълги съобщения е бавен процес, изискващ много ресурси. Този механизъм въвлича функция, която създава излишък (разширява съобщението), за да не бъде всеки възможен подпис в действителност подпис на истинско съобщение, т.е. броят на възможните съобщения да бъде много по-малък от този на възможните подписи.
Този тип електронен подпис не е визиран от "Закона за електронния документ и електронния подпис" (ЗЕДЕП), но би могъл да се ползва при разпространение на секретни ключове (за симетрични криптосистеми) или подписване на сертификати.

В някои случаи е необходимо да се подпише дадено съобщение, но без да се разкрива неговото съдържание. В тези случаи се прилага така нареченото ***сляпо подписване (blind signature)***. Този механизъм е реализиран в разпространения протокол електронни разплащания "Digicash". Ето накратко описанието на този механизъм:

Да предположим, че A иска B (частно лице, овластен субект, ТТР или ДУУ и др.) да подпише неговия документ M, но без съдържанието му да стане известно на B. Нека e и d са, съответно, публичния и частен ключ на B. Тогава

1. A генерира случайно число R, изчислява $M_1 = R^e M (mod\ n)$ и го изпраща на B.

2. Използвайки своя частен ключ, d, B пресмята $S = M_1^d = R^{ed}.M^d = R.M^d$ по модул n и го връща на A.

3. При получаването на S, A умножава S с R^{-1} за да намери M^d, което представлява подписания документ.

В литературата са описани и редица други механизми от тип електронен подпис, които са реализирани в специфични приложения . Един такъв механизъм е така наречения *fail-stop* подпис. Той позволява да се открива, ако е направена фалшификация и съответния публичен ключ да се изважда веднага от употреба. Друг тип е така наречения *undeniable signature scheme*, който изисква сътрудничеството на подписващия за проверка на подписа. Реализира се например, когато подписващият осъществява достъп до нещо (например, сейф) в дадено време. Целта е никой да не може да твърди, че този достъп е направен, когато не е осъществяван или във време различно от истинското. Тези и други подобни механизми излизат от рамките на нашия курс и няма да ги разглеждаме подробно.

6.4 Генериране на големи прости числа.

Както видяхме в предходните параграфи много от съвременнитеза асиметрични криптосистеми изискват като ключове или параметри големи прости числа от порядъка на 1024 бита. Методи като решетото на Ератостен и проверки за простотота чрез деление на прости делители $< \sqrt{n}$ са очевидно неефективни дори за 128 битови числа. Затова на основата на резултати от Теория на числата са създадени множество тестове, чрез които се проверява дали дадено нечетно число е просто. Процесът на генериране на голямо просто число представлява поредица от по две стъпки - генериране на голямо нечетно число (най-често случайно) и проверката му чрез тест за простота. Процедурата продължава докато теста даде положителен резултат.

Тестовете за простота се делят на два типа:

- вероятностни
- детерминистични

Най-често се използват първите.

6.4.1 Вероятностни тестове за простота. Тест на Miller-Rabin.

Най-общо тези тестове се описват така:

Дефинира се множество $W(n) \subset \{1, 2, \ldots, n-1\}$, което за всяко нечетно естествено число n притежава следните свойства:

(1) Ако n е просто число, то $W(n) = \emptyset$;

(2) Ако n е съставно число, то $\mid W(n) \mid \geq \frac{n}{2}$

(3) За всяко $a \in \{1, 2, \ldots, n-1\}$ може "лесно" (за полиномиално време) да се определи дали $a \in W(n)$.

При проверка за дадено a тестът дава, че или n е съставно, или с вероятност $\geq \frac{1}{2}$ е просто. Ако се направят t проверки (итерации на теста) и те са положителни, то n е съставно (т.е. тестът е дава грешен резултат) с вероятност $< \left(\frac{1}{2}\right)^t$.

Тест на Miller-Rabin.

Твърдение 6.4.1 *Нека p е нечетно просто число и $p-1 = 2^e r$, където r е нечетно число. Ако a удовлетворява $(a,p)=1$, то или $a^r \equiv 1 \pmod p$, или $a^{2^i r} \equiv -1 \pmod p$ за някое i : $0 \leq i \leq e-1$.*

Доказателство. Съгласно теоремата на Ферма за всяко такова a е в сила $a^{p-1} \equiv 1 \pmod p$. Следователно $x = a^{\frac{p-1}{2}}$ удовлетворява $x^2 \equiv 1 \pmod p$. Но последното влече $a^{\frac{p-1}{2}} \equiv \pm 1 \pmod p$. Ако $a^{\frac{p-1}{2}} \equiv 1 \pmod p$ продължаваме аналогично като разглеждаме $a^{\frac{p-1}{4}}$ и т.н. С това твърдението е доказано.

Горното твърдение дава необходимо, но не и достатъчно, условие едно нечетно число да е просто, на което се основава тестът на Милер. На всяка итерация от теста за дадено a, $(a,n)=1$, се проверява, дали се изпълнява някое от сравненията в Твърдение 6.4.1. Ако резултатът е отрицателен, n е съставно число. Ако е положителен се преминава към следваща итерация (за друго a) докато се извършат първоначално зададен брой от t итерации.

Дефиниция 6.4.2 *Съставно нечетно число n, което издържа теста на Милер за дадено a, $(a,n)=1$, се нарича* ***силно псевдопросто число при основа*** *a.*

Вероятността за грешка след t итерации на теста на Милер-Рабин е $\leq \left(\frac{1}{4}\right)^t$, тъй като съгласно Теорема 6.4.4 вероятността за грешка при една итерация не надминава $1/4$.

Лема 6.4.3 *Нека p е нечетно просто число и $p-1 = 2^k r$. Тогава сравнението*

$$x^{2^i t} \equiv -1 \pmod p \tag{6.1}$$

има точно $2^i(t,r)$ решения, ако $i \leq k-1$ и няма решение при $i \geq k$.

Доказателство. Нека $d = (2^i t,\, \varphi(p)) = (2^i t,\, 2^k r) = 2^m (t,r)$, където $m = \min(k,i)$. Съгласно Теорема 4.1.3 сравнението (6.1) има решение тогава и само тогава, когато

$$(-1)^{\frac{\varphi(p)}{d}} \equiv 1 \pmod p$$

и броят на решенията му е d. Но това е изпълнено тогава и само тогава, когато

$$\frac{2^k r}{2^m (t,r)}$$

е четно число, т.е. $k > m$. Следователно при $i \geq k$ (6.1) няма решение, а при $i \leq k-1$ има точно $2^i(t,r)$ решения.

Теорема 6.4.4 *Ако n е нечетно и съставно естествено число, то n издържа теста на Милер-Рабин най-много за $\frac{1}{4}\varphi(n)$ основи a, $(a,n)=1$, и $1 \leq a \leq n-1$.*

Доказателство. Възможни са два случая.
(1) Нека $p^2 \mid n$ за някое просто число p. Да предположим, че a е основа относно, която n е силно псевдопросто. Тогава от $a^{n-1} \equiv 1 \pmod{n}$ следва, че $a^{n-1} \equiv 1 \pmod{p^2}$. Съгласно Теорема 4.1.3 броят на решенията на последното сравнение е

$$d = (\varphi(p^2), n-1) = (p(p-1), n-1) \quad \text{и} \quad d \mid (p-1),$$

тъй като $p \nmid (n-1)$. Следователно $d \leq p-1$ и отношението на основите a, относно които n е силно псевдопросто към всички възможни p^2-1 по модул p^2 числа е

$$\leq \frac{p-1}{p^2-1} = \frac{1}{p+1} \leq \frac{1}{4}.$$

Да отбележим, че при $n = 3^2$ равенството се достига.

(2) Нека сега n е произведение на различни прости числа. Ще разгледаме само $n = pq$. В общия случай се процедира аналогично. Нека $p-1 = 2^k t_1$ и $q-1 = 2^l t_2$, където t_1, t_2 са нечетни и $k \leq l$. Ако a е основа относно, която n е силно псевдопросто и $n-1 = 2^e t$, то една от следните възможности трябва да е в сила:

(i) $a^t \equiv 1 \pmod{p}$ и $a^t \equiv 1 \pmod{q}$
(ii) $a^{2^j t} \equiv -1 \pmod{p}$ и $a^{2^j t} \equiv -1 \pmod{q}$, за някое $0 \leq j \leq e-1$.

Съгласно Теорема 4.1.3, ако е изпълнено (i) броят на решенията на всяко от сравненията е съответно $(t, p-1) = (t, t_1)$ и $(t, q-1) = (t, t_2)$. Сега Китайската теорема ни дава, че (i) е изпълнено за $(t, t_1)(t, t_2) \leq t_1 t_2$ числа a.
Нека е налице (ii). Съгласно Лема 6.4.3 сравненията на (ii) имат съответно $2^j(t, t_1)$ и $2^j(t, t_2)$ решения за $0 \leq j \leq k-1$. Следователно за фиксирано j броят на търсените числа a удовлетворяващи (ii) е $2^{2j}(t, t_1)(t, t_2) \leq 4^j t_1 t_2$. Сумирайки по j получаваме

$$t_1 t_2 (1 + 4 + \cdots + 4^{k-1}) = t_1 t_2 \frac{4^k - 1}{3}.$$

Следователно отношението R на търсените основи към всички $\varphi(n) = 2^{k+l} t_1 t_2$ възможни основи при $k < l$ е

$$R \leq \frac{t_1 t_2 + \frac{t_1 t_2 (4^k - 1)}{3}}{2^{k+l} t_1 t_2} = \frac{4^k + 2}{3 \cdot 2^{k+l}} \leq \frac{4^k + 2}{6 \cdot 2^{2k}} = \frac{1}{6} + \frac{2}{6 \cdot 4^k} \leq \frac{1}{4}.$$

Нека $k = l$. Тогава не е възможно едновременно $(t, t_1) = t_1$ и $(t, t_2) = t_2$. Наистина да допуснем противното. В такъв случай $t_1 \mid t$ и $t_2 \mid t$. Тъй като $p = 2^k t_1 + 1$ и $q = 2^k t_2 + 1$, то $2^e t = n - 1 = pq - 1 = 2^{2k} t_1 t_2 + 2^k t_1 + 2^k t_2$. Следователно $t_1 \mid t_2$ и $t_2 \mid t_1$, т.е. $t_1 = t_2$. Но тогава $p = q$, което е противоречие. И така, нека за конкретност $(t_1, t) < t_1$. Тъй като t_1 е нечетно, то $(t_1, t) \leq \frac{1}{3} t_1$. Тогава за фиксирано j броят на търсените числа a удовлетворяващи (ii) е $2^{2j}(t, t_1)(t, t_2) \leq 4^j t_1 t_2 / 3$. Следователно

$$R \leq \frac{\frac{t_1 t_2}{3} + \frac{t_1 t_2 (4^k - 1)}{9}}{2^{2k} t_1 t_2} = \frac{4^k + 2}{9 \cdot 4^k} = \frac{1}{9} + \frac{2}{9 \cdot 4^k} \leq \frac{1}{9} + \frac{2}{9 \cdot 4} < \frac{1}{4}.$$

6.4.2 Детерминистични тестове за простота.

При този тип тестове се доказва, че нечетното число n е просто. За целта се проверява, че някое достатъчно условие е изпълнено. Естествено, този тип тестове изискват повече изчислителни ресурси и са по бавни от вероятностните. Те се използват за доказване на простота (ако е необходимо) на числа, които вече са намерени с вероятностни тестове.

Един такъв тест например се базира на следното добре известно твърдение.

Твърдение 6.4.5 *Числото $n \geq 3$ е просто тогава и само тогава, когато съществува a, такова че*

(1) $a^{n-1} \equiv 1 \pmod{n}$ *и*

(2) $a^{\frac{n-1}{q}} \not\equiv 1 \pmod{n}$ *за всеки прост делител q на $n-1$.*

Един тест за проверка дали $2^k - 1$ е просто число на Мерсен се основава на следния резултат.

Теорема 6.4.6 *Нека $k \geq 3$. Числото $M_k = 2^k - 1$ е просто тогава и само тогава, когато са изпълнени следните две условия:*

(a) *k е просто число и*

(b) *рекурентната редица дефинирана с*

$$s_{j+1} \equiv (s_j^2 - 2) \pmod{M_k} \quad \text{за } j \geq 1, \ \text{и} \quad s_1 = 4, \tag{6.2}$$

удовлетворява $s_{k-1} \equiv 0 \pmod{M_k}$.

Доказателство. Да положим $\alpha = 2 + \sqrt{3}$ и $\beta = 2 - \sqrt{3}$. Очевидно, че $\alpha\beta = 1$. С метода на математическата индукция лесно се показва, че общият член на редицата $\{s_n\}$, определена с (6.2), се задава с формулата

$$s_n = \alpha^{2^{n-1}} + \beta^{2^{n-1}}, \ \ n = 1, 2, \ldots, \tag{6.3}$$

Достатъчност. Нека $p \geq 3$ е просто число и $s_{p-1} \equiv 0 \pmod{M_p}$. Тогава $s_{p-1} = \alpha^{2^{p-2}} + \beta^{2^{p-2}}$, откъдето умножавайки по $\alpha^{2^{p-2}}$ получаваме

$$\alpha^{2^{p-1}} + 1 = s_{p-1}\alpha^{2^{p-2}}.$$

В такъв случай условието $M_p \mid s_{p-1}$ влече

$$\alpha^{2^{p-1}} + 1 = M_p \cdot d \cdot \alpha^{2^{p-2}}.$$

Да допуснем, че M_p не е просто. Тогава съществува просто число $q \leq \sqrt{M_p}$ със свойството $q \mid M_p$. Следователно $\alpha^{2^{p-1}} \equiv -1 \pmod{q}$, т.е.

$$\alpha^{2^p} \equiv 1 \pmod{q}. \tag{6.4}$$

Да разгледаме мултипликативната група на пръстена $\mathbb{Z}_q[\sqrt{3}] = \{a + b\sqrt{3} \mid a, b \in \mathbb{Z}_q\}$ Тогава (6.4) означава, че редът на α в $\mathbb{Z}_q^\star[\sqrt{3}]$ е $o(\alpha) = 2^p$. Следователно

$$2^p \leq | \mathbb{Z}_q^\star[\sqrt{3}] | \leq q^2 - 1 \leq M_p - 1 = 2^p - 2,$$

което е противоречие.

Необходимост. Нека $M_p = 2^p - 1$ е просто число на Мерсен. Очевидно p трябва да е просто. Остава да покажем, че е изпълнено и условие (b). За целта да разгледаме $\tau = (1 + \sqrt{3})/\sqrt{2}$ и $\bar{\tau} = (1 - \sqrt{3})/\sqrt{2}$, $\tau\bar{\tau} = -1$. Тъй като $\tau^2 = 2 + \sqrt{3} = \alpha$, то $\tau^{q+1} = \tau^{2^p} = \alpha^{2^{p-1}}$, където $q = M_p$. Тогава

$$(\tau\sqrt{2})^q = 1 + q\sqrt{3} + \binom{q}{2}3 + \cdots + \binom{q}{i}(\sqrt{3})^i + \cdots + 3^{\frac{q-1}{2}}\sqrt{3}.$$

Но q дели $\binom{q}{i}$, тъй като е просто и следователно

$$2^{\frac{q-1}{2}}\tau^q\sqrt{2} = 1 + qa\sqrt{3} + qb + 3^{\frac{q-1}{2}}\sqrt{3}, \tag{6.5}$$

където $a, b \in \mathbb{Z}$. Тъй като $q = 2^p - 1 \equiv -1 - 1 \equiv 1 (\mathrm{mod}\ 3)$, то $\left(\frac{q}{3}\right) = \left(\frac{1}{3}\right) = 1$ и съгласно квадратичния закон за реципрочност

$$\left(\frac{3}{q}\right) = (-1)^{\frac{q-1}{2}}\left(\frac{1}{3}\right) = -1.$$

Тогава критерият на Ойлер ни дава, че

$$3^{\frac{q-1}{2}} \equiv -1 \ (\mathrm{mod}\ q).$$

Аналогично (за $p \geq 3$) е в сила $q \equiv -1 (\mathrm{mod}\ 8)$, т.е. $\left(\frac{2}{q}\right) = 1$ и следователно

$$2^{\frac{q-1}{2}} \equiv 1 \ (\mathrm{mod}\ q).$$

Замествайки в (6.5) получаваме, че съществуват цели числа c, d:

$$(1 + cq)\tau^q\sqrt{2} = 1 + qa\sqrt{3} + qb + (-1 + dq)\sqrt{3},$$

т.е.

$$\sqrt{2}(1 + cq)\tau^q = \bar{\tau}\sqrt{2} + [(a + d)\sqrt{3} + b]q.$$

Умножавайки с $\tau/\sqrt{2}$ получаваме

$$(1 + cq)\tau^{q+1} = -1 + \frac{[(a + d)\sqrt{3} + b](1 + \sqrt{3})}{2} \cdot q.$$

Следователно

$$(1 + cq)\alpha^{2^{p-1}} = -1 + \frac{e + f\sqrt{3}}{2} \cdot q,$$

където $e, f \in \mathbb{Z}$. След умножаване с $2\beta^{2^{p-2}}$ горното равенство и $\alpha\beta = 1$ ни дават

$$2\alpha^{2^{p-2}}(1 + cq) = -2\beta^{2^{p-2}} + (e + f\sqrt{3})\beta^{2^{p-2}}q,$$

т.е.

$$2(\alpha^{2^{p-2}} + \beta^{2^{p-2}}) = -2\alpha^{2^{p-2}}cq + (e + f\sqrt{3})\beta^{2^{p-2}}q.$$

Следователно

$$2s_{p-2} = [-2\alpha^{2^{p-2}}c + (e + f\sqrt{3})\beta^{2^{p-2}}]q.$$

Тъй като лявата страна е цяло число, то и в скобите на дясната страна трябва да е рационално (т.е. цяло) число. Но тогава

$$s_{p-2} \equiv 0 \pmod{q}.$$

С това доказателството е завършено.