МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ

КОМП'ЮТЕРНА КРИПТОГРАФІЯ

Опорний конспект лекцій
(спеціальності 6.050201 – системна інженерія)

ТЕРНОПІЛЬ
ТНЕУ

ЛЕКЦІЯ 1
ОСНОВНІ ПОНЯТТЯ КРИПТОГРАФІЇ

Анотація: У даній лекції визначаються предмет і завдання криптографії, формулюються основоположні визначення курсу і вимоги до криптографічних систем захисту інформації, дається історична довідка про основні етапи розвитку криптографії як науки. Також розглядається приклад найпростішого шифру, на основі якого пояснюються сформульовані поняття і тези.

Предмет і завдання криптографії

Проблемою захисту інформації при її передачі між абонентами люди займаються протягом всієї своєї історії. Людством винайдено безліч способів, що дозволяють в тій чи іншій мірі приховати зміст повідомлень від противника. На практиці виробилося кілька груп методів захисту секретних послань. Назвемо деякі з них, що застосовуються так само давно, як і криптографічні.

Першим способом є фізичний захист матеріального носія інформації від противника. В якості носія даних може виступати папір, комп'ютерний носій (DVD-диск, флеш-карта, магнітний диск, жорсткий диск комп'ютера і т.д.). Для реалізації цього способу необхідний надійний канал зв'язку, недоступний для перехоплення. В різний час для цього використовувалися поштові голуби, спеціальні кур'єри, радіопередачі на секретній частоті. Методи фізичного захисту інформації використовуються і в сучасних автоматизованих системах обробки даних. Так, наприклад, комплексні системи захисту інформації неможливі без систем огорожі і фізичної ізоляції, а також без охоронних систем.

Другий спосіб захисту інформації, відомий з давніх часів - стеганографічний захист інформації. Цей спосіб захисту заснований на спробі приховати від противника сам факт наявності інформації. При стеганографічному методі захисту від супротивника ховають фізичний носій даних або маскують секретні повідомлення серед відкритої, несекретної інформації. До таких способів відносять, наприклад, "захування" мікрофотографії з таємною інформацією в несекретної місці: під маркою на поштовому конверті, під обкладинкою книги і т.д. До стеганографії належать також такі відомі прийоми, як "захування" секретного послання в корінцях книг, в ґудзиках, в підборах, в пломбі зуба і т.д. Деякі з методів були розроблені ще в стародавні часи. Так, наприклад, греки знайшли незвичайне рішення: вони голили наголо голову раба і видряпували на ній своє послання. Коли волосся на голові раба відростало знову, його посилали доставити повідомлення. Одержувач голив голову раба і прочитував текст. На жаль, на відправку повідомлення та отримання відповіді таким способом йшло кілька тижнів. У більш пізні часи в цьому напрямку найбільшого поширення набули хімічні (симпатичні) чорнило. Текст, написаний цими чорнилом між рядків несекретного повідомлення, невидимий. Він з'являвся тільки в результаті застосування певної технології прояви.

В умовах повсюдного використання інформаційних технологій виникають нові стеганографічні прийоми. Наприклад, відомий спосіб, при якому секретне повідомлення ховається в файлі графічного зображення. При використанні цього способу молодший значущий біт в описі кожного пікселя зображення замінюється бітом повідомлення. Розділивши все вихідне повідомлення на біти і розмістивши ці біти по всьому графічного файлу, ми пересилаємо зображення з замаскованим повідомленням одержувачу. Графічне зображення при цьому змінюється не дуже сильно, особливо якщо використовувався режим з великою кількістю квітів, наприклад, з глибиною кольору 24 біта на піксель. Це пов'язано з тим, що людське око не може розрізняти таку велику кількість квітів. В результаті в зображенні розміром всього 32 на 32 точки можна вмістити таємне повідомлення довжиною 1024 біта або 128 байт.

Третій спосіб захисту інформації - найбільш надійний і поширений в наші дні - криптографічний. Цей метод захисту інформації передбачає перетворення інформації для приховування її сенсу від противника. Криптографія в перекладі з грецького означає "тайнопис". В даний час криптографія займається пошуком і дослідженням математичних методів перетворення інформації.

Поряд з криптографією розвивається і вдосконалюється **криптоаналіз** - наука про подолання криптографічного захисту інформації. Криптоаналітики досліджують можливості розшифровки інформації без знання ключів. Успішно проведений криптоаналіз дозволяє отримати ключ шифрування, або відкритий текст, або те й інше разом. Іноді криптографію і криптоаналіз об'єднують в одну науку - **криптології** (kryptos - таємний, logos - наука), що займається питаннями оборотного перетворення інформації з метою захисту від несанкціонованого доступу, оцінкою надійності систем шифрування і аналізом стійкості шифрів.

В даний час криптографія міцно увійшла в наше життя. Перерахуємо лише деякі сфери застосування криптографії в сучасному інформатизованому суспільстві:

- • шифрування даних при передачі по відкритих каналах зв'язку (наприклад, при здійсненні покупки в Інтернеті відомості про угоду, такі як адреса, телефон, номер кредитної картки, зазвичай зашифровуються в цілях безпеки);
- • обслуговування банківських пластикових карт;
- • зберігання і обробка паролів користувачів в мережі;
- • здача бухгалтерських та інших звітів через віддалені канали зв'язку;
- • банківське обслуговування підприємств через локальну або глобальну мережу;
- • безпечне від несанкціонованого доступу зберігання даних на жорсткому диску комп'ютера (в операційній системі Windows навіть є спеціальний термін - шифрована файлова система (EFS)).

До початку XX століття криптографічні методи застосовувалися лише для шифрування даних з метою захисту від несанкціонованого доступу. У двадцятому столітті в зв'язку з розвитком техніки передачі інформації на далекі відстані інтерес до криптографії значно зріс. Завдяки створенню нових криптографічних методів розширився і спектр завдань криптографії. В даний час вважається, що криптографія призначена вирішувати такі завдання:

- • власне шифрування даних з метою захисту від несанкціонованого доступу;
- • перевірка справжності повідомлень: одержувач повідомлення може перевірити його джерело;
- • перевірка цілісності переданих даних: одержувач може перевірити, чи не було повідомлення змінено або підмінено в процесі пересилання;
- • забезпечення неможливості відмови, тобто неможливість як для одержувача, так і для відправника відмовитися від факту передачі.

Системи шифрування варіюються від найелементарніших до дуже складних. І якщо перші не вимагають ніяких математичних знань, то в останніх використовуються поняття, знайомі лише фахівцям в деяких областях математики та інформатики. При використанні криптографічних методів повинні враховуватися витрати на захист інформації та на реалізацію методів нападу. На практиці прагнуть до досягнення компромісу між вартістю шифрування і необхідним ступенем забезпечення безпеки.

Основні визначення

Тепер, дізнавшись призначення криптографії, познайомимося з основними термінами, які будемо використовувати при вивченні криптографічних методів захисту інформації.

Шифр - сукупність заздалегідь обумовлених способів перетворення вихідного секретного повідомлення з метою його захисту.

Вихідні повідомлення зазвичай називають **відкритими текстами**. В іноземній літературі для відкритого тексту використовують термін **plaintext**.

Символ - це будь-який знак, в тому числі буква, цифра або знак пунктуації.

Алфавіт - кількість використовуваних для кодування інформації символів. Наприклад, український алфавіт містить 33 літери від А до Я. Однак цих тридцяти трьох знаків зазвичай буває недостатньо для запису повідомлень, тому їх доповнюють символом пробілу, крапкою, комою і іншими знаками. Алфавіт арабських цифр - це символи 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Цей алфавіт містить 10 знаків і з його допомогою можна записати будь-яке натуральне число. Будь-яке повідомлення може бути записано також за допомогою *двійкового алфавіту*, тобто з використанням тільки нулів і одиниць.

Повідомлення, отримане після перетворення з використанням будь-якого шифру, називається **зашифрованим повідомленням** (закритим текстом, криптограмою). В іноземній літературі для закритого тексту використовують термін **ciphertext**.

Перетворення відкритого тексту в криптограму називається **шифруванням**. Зворотна дія називається розшифрування. В англомовній літературі термінам "**шифрування/розшифрування**" відповідають терміни "**enciphering/deciphering**".

Ключ – інформація, необхідна для шифрування і розшифрування повідомлень.

З точки зору української мови терміни "розшифрування" і "дешифрування" є синонімами. Однак в роботах по криптографії останніх десятиліть часто ці слова розрізняють. Будемо вважати, що терміни "розшифрування" та "дешифрування" не є синонімами. Приймемо, що розшифрування займається легальний одержувач повідомлення (той, хто знає ключ), а людина, якій послання не призначене, намагаючись зрозуміти його зміст, займається дешифруванням.

Система шифрування, або шифросистема, - це будь-яка система, яку можна використовувати для оберненої зміни тексту повідомлення з метою зробити його незрозумілим для всіх, крім тих, кому воно призначене.

Криптостійкістю називається характеристика шифру, що визначає його стійкість до дешифрування без знання ключа (тобто здатність протистояти криптоаналізу).

Таким чином, з урахуванням всіх зроблених визначень можна дати більш точне визначення науці "криптографія". **Криптографія** вивчає побудову і використання систем шифрування, в тому числі їх стійкість, слабкість і ступінь уразливості щодо різних методів розкриття.

Всі методи перетворення інформації з метою захисту від несанкціонованого доступу діляться на дві великі групи: методи шифрування із закритим ключем і методи шифрування з відкритим ключем. **Шифрування з закритим ключем** (шифрування з секретним ключем або симетричне шифрування) використовується людиною вже досить довгий час. Для шифрування і розшифрування даних в цих методах використовується один і той же ключ, який обидві сторони намагаються зберігати в секреті від противника. **Шифрування з відкритим ключем** (асиметричне шифрування) стало використовуватися для криптографічного закриття інформації лише в другій половині XX століття. У цю групу відносяться методи шифрування, в яких для

шифрування і розшифрування даних використовуються два різних ключа. При цьому один з ключів (відкритий ключ) може передаватися по відкритому (незахищеному) каналу зв'язку.

Електронним (цифровим) підписом називається зазвичай приєднаний до повідомлення блок даних, отриманий з використанням криптографічного перетворення. Електронний підпис дозволяє при отриманні тексту іншим користувачем перевірити авторство і достовірність повідомлення.

Криптографічна система захисту інформації - система захисту інформації, в якій використовуються криптографічні методи для шифрування даних.

Вимоги до криптографічних систем захисту інформації

Для розроблюваних в даний час криптографічних систем захисту інформації сформульовані наступні загальноприйняті вимоги:

- • зашифроване повідомлення повинно піддаватися читанню тільки при наявності ключа;
- • знання алгоритму шифрування не повинно впливати на надійність захисту;
- • будь-який ключ з безлічі можливих повинен забезпечувати надійний захист інформації;
- • алгоритм шифрування повинен допускати як програмну, так і апаратну реалізацію.

Відомості з історії криптографії

Історично криптографія розвивалася як практична дисципліна, що вивчає і розробляє способи шифрування письмових повідомлень. У істориків є дані, що криптографічні методи застосовувалися в Стародавньому Єгипті, Індії, Месопотамії. Так, наприклад, в записах єгипетських жерців є відомості про системи і способи складання шифрованих послань.

Стародавні греки залишили документальні підтвердження про різні вживані ними шифрувальні системи. Греками, а вірніше спартанцями, під час численних воєн застосовувалося одне з перших шифрувальних пристроїв - Скитала. Скитала представляла собою циліндричний жезл певного діаметру. На Скиталу виток до витка намотувалася вузька смужка папірусу (або шкіряного ременя). На намотаній стрічці вздовж осі жезла писали відкрите повідомлення. Потім стрічку розмотували і переправляли адресату. Після зняття папірусу з жезла виходило наче літери повідомлення написані в безладді поперек стрічки. Якщо папірус потрапляв в руки супротивника, то секретне повідомлення прочитати було неможливо. Для отримання вихідного тексту була необхідна Скитала точно такого ж діаметру - на неї намотувалася отримана смужка папірусу, рядки повідомлення поєднувалися, і в результаті можна було прочитати таємне послання. Ключем в даному методі шифрування був діаметр Сцитали. Цікаво, що винахід дешифрувального "пристрою" приписується Арістотелю.

В различные времена криптографией занимались многие политики и ученые. Среди них Пифагор, Аристотель, Платон, Галилей, Д. Порта, Д. Кардано, Л. да Винчи, Ф. Виет, Д. Валлис, Б. Паскаль, И. Ньютон, Ф. Бекон, Х. Гольбах, Ф. Эпинус, Л. Эйлер, П.Ф. Шиллинг, Ч. Беббидж и другие.

У XX столітті з'явилися нові можливості по передачі інформації на великі відстані з великою швидкістю. У зв'язку із застосуванням радіозв'язку розширилися можливості доступу до шифрованої інформації в процесі її передачі. Науково-технічний прогрес перетворив криптографію, яка стала спочатку електромеханічною, а потім електронною. У XX столітті виникає спеціалізація в криптографічної діяльності. З'являються фахівці з шифрування, по перехопленню зашифрованих повідомлень, по дешифруванню шифрів противника

У 20-х роках XX століття для автоматизації процесу шифрування з'явилися численні механічні пристрої. Зокрема, широко використовувалися роторні шифрувальні машини, в яких

для виконання операцій заміни символів застосовувалися механічні колеса - ротори. Шифрувальні машини перетворювали відкритий текст в зашифрований, що складається з символів того ж алфавіту. Після перетворення зашифрована інформація могла передаватися різними способами, наприклад, по радіоканалу. У всіх розвинених країнах створювалися високошвидкісні шифромашини, які широко застосовувалися під час Другої світової війни і пізніше.

В середині XX століття розробкою криптографічних алгоритмів стали займатися професійні математики та фахівці в галузі інформатики. Істотний вплив на розвиток криптографії справила робота американського інженера-математика К. Шеннона "Теорія зв'язку в секретних системах", в якій були сформульовані і математично доведені умови "нерозкриваємості" шифрів.

З 50-х років XX століття в криптографії використовується електронна обчислювальна техніка. Починається створення так званих блокових шифрів, які дозволяють обробляти інформацію цілими фрагментами або блоками. Спочатку для операцій блочного шифрування розробляли апаратні пристрої з жорсткою логікою, проте стрімкий розвиток можливостей обчислювальної техніки дозволив створити програмні аналоги блочних систем шифрування. Криптографічні програмні і апаратні засоби стали використовуватися в цивільних цілях, наприклад, в комерційних системах передачі інформації.

З розвитком інформаційних технологій криптографія не тільки набула нових сфер застосування, а й зазнала значних змін. У стародавні часи в процесі обміну зашифрованими повідомленнями брало участь тільки дві сторони, тому ключем шифрування необхідно було забезпечити тільки ці дві сторони. В сучасних інформаційних системах в процесі передачі інформації задіяно безліч абонентів, і всі вони зацікавлені в надійних і зручних каналах отримання ключів шифрування. Проблема розподілу ключів була вирішена в двадцятому столітті завдяки винаходу нового принципу шифрування - асиметричного шифрування або шифрування з відкритим ключем (70-ті роки XX ст.). Засновниками цього методу шифрування вважаються У. Діффі і М. Хеллмана. В асиметричних алгоритмах шифрування використовуються спеціальні математичні функції - односторонні функції. Відкриття асиметричних криптосистем дозволило ще більше розширити сфери застосування криптографії. Саме шифрування з відкритим ключем лежить в основі процедур формування цифрового підпису та перевірки автентичності, а отже, і в основі принципів роботи банківських пластикових карт, "електронних" грошей та інших сучасних технологій.

Криптографічні атаки

Інформація в процесі зберігання, передачі і перетворення піддається впливу різних атак. Атаки здійснюються противниками (опонентами, перехоплювачами, ворогами і т.д.). Основними порушеннями безпеки є розкриття інформаційних цінностей (втрата конфіденційності), модифікація без дозволу автора (втрата цілісності) або неавторизована втрата доступу до цих цінностей (втрата доступності) .

Атаки можуть бути пасивними і активними. **Пасивної** називається атака, при якій супротивник не має можливості змінювати передані повідомлення. При пасивній атаці можливо лише прослуховування повідомлень, що передаються, їх дешифрування і аналіз трафіку. При **активній** атаці противник має можливість модифікувати передані повідомлення і навіть додавати свої повідомлення.

Криптографічні атаки можна класифікувати за кількістю і типом інформації, доступної для криптоаналізу противником. За цією класифікацією виділяють такі види атак.

Атака на основі шифротексту є в тому випадку, коли противник має для аналізу шифротексти різних невідомих відкритих текстів, зашифровані на одному і тому ж ключі. Завдання криптоаналитика полягає в отриманні відкритого тексту якомога більшого числа повідомлень або в отриманні ключа, використаного при шифруванні. Отриманий ключ буде потім використаний для дешифрування інших повідомлень.

Атака на основі відомого відкритого тексту має місце в тому випадку, якщо криптоаналитик отримує в своє розпорядження будь-які відкриті тексти, що відповідають раніше переданим зашифрованим повідомленням. Зіставляючи пари "текст-шифротекст", противник намагається дізнатися секретний ключ, щоб з його допомогою дешифрувати всі наступні повідомлення. Деяким здасться, що противнику досить складно дістати в своє розпорядження кілька пар "текст-шифротекст". Насправді практично завжди можливо дістати такі шматочки відкритого тексту і шифротекста. Криптоаналітика може мати інформацію про формат перехопленого зашифрованого файлу: наприклад, знати, що це файл із зображенням JPEG, документ Word або Excel, файл бази даних або щось ще. Всі ці та багато інших форматів містять певні стандартні заголовки або фрагменти. Таким чином, фахівець з криптоаналізу зможе сформувати необхідні дані для проведення атаки на основі відомого відкритого тексту.

Можливий ще більш "серйозний" для передавальних сторін варіант - *це атака на основі обраного відкритого тексту*. В цьому випадку криптоаналитик має можливість не тільки використовувати надані йому пари "текст-шифротекст", але і сам формувати потрібні йому тексти і шифрувати їх за допомогою того ключа, який він хоче дізнатися. Відомо, що під час другої світової війни американці, підкупивши охорону, викрали шифрувальну машину в японському посольстві на два дні і мали можливість формувати і подавати їй на вхід різні тексти і отримувати відповідні шифровки. (Вони не могли зламати машину з метою безпосереднього визначення закладеного в неї секретного ключа, так як це було б помічено і спричинило б за собою зміну всіх ключів.)

Довгий час розробники криптосистем намагалися зробити свої алгоритми шифрування невразливими по відношенню тільки до атак по шифротексту і забезпечувати організаційно неможливість атак з відкритого або заданого тексту. Для цього тримали в таємниці алгоритми шифрування, пристрої шифрувальних машин, ретельно перевіряли на надійність персонал, що має доступ до криптосистем.

Однак ще в XIX столітті фахівці в області криптографії припустили, що секретність алгоритму шифрування не є гарантією від злому. Більш того, в подальшому було зрозуміле, що по-справжньому надійна система шифрування повинна залишатися захищеною, навіть якщо супротивник повністю дізнався алгоритм шифрування. Секретність ключа повинна бути достатня для хорошого шифру, щоб зберегти стійкість до спроб злому. Цей фундаментальний принцип вперше був сформульований в 1883 Керкхоффсом (A. Kerckhoffs) і зазвичай називається **принципом Керкхоффса**.

Розробники сучасних криптографічних систем використовують саме такий підхід, передбачаючи можливість атак за обраним текстом. В даний час створювані алгоритми шифрування всебічно вивчаються великим числом фахівців, оцінюються за різними показниками, в тому числі і по можливості протистояти атакам по обраному тексту.

Приклади найпростіших шифрів

Шифри перестановки та простої заміни

При шифруванні перестановкою символи відкритого тексту переставляються за визначеним правилом в межах блоку цього тексту. Шифри перестановки є найпростішими та найдревнішими шифрами.

Пізніше почали використовувати шифр частоколу. Наприклад:

р п о р ф я
к и т г а і

Отримується шифртекст: рпосфякитгаі. Ключем є висота частоколу. Зокрема, для частоколу висотою 3 маємо (ліворуч):

и г і и о а я
р п о р ф я р т р і
к т а к п г ф

Отримується шифротекст: игірпорфякта. Це складний частокіл. При використанні простого частоколу отримуємо: иоаяртрікпгф. аналогічно можна використати частокіл і більшої висоти.

З кінця 14 ст. виникли шифруючі таблиці. Наприклад, відкритий текст записується в таблицю по стовбцях, а читається по рядках. Ключем є розмір таблиці. Їх удосконаленням стали шифруючі таблиці з ключовими словами, коли стовбці та рядки переставляються у відповідності до цих ключових слів.

		л	і	т	о
		2	1	4	3
з	2	п	р	и	л
и	3	і	т	а	ю
м	4	в	о	с	ь
а	1	м	о	г	о

		і	л	о	т
		1	2	3	4
з	2	р	п	л	и
и	3	т	і	ю	а
м	4	о	в	ь	с
а	1	о	м	о	г

1	о	м	о	г
2	р	п	л	и
3	т	і	ю	а
4	о	в	ь	с

В середні віки використовувалося також шифрування за допомогою магічних квадратів. Це квадратні таблиці з вписаними в клітинки послідовностями натуральних чисел, починаючи з 1, щоб їх сума по стовбцях, рядках та діагоналях дорівнювала одному і тому самому числу.

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

о	и	р	м
і	о	с	ю
в	т	а	ь
л	г	о	п

Якщо не враховувати повороти, то існує тільки один квадрат розміром 3x3, 880 – розміром 4x4, біля 250000 – розміром 5x5.

Тепер, коли дані основні визначення, розглянемо одну з найпростіших систем шифрування, яка носить ім'я "шифр Юлія Цезаря". Передбачається, що знаменитий римський імператор і полководець, що жив в 1 столітті до нашої ери, використовував цей шифр в своєму листуванні.

Шифр Цезаря стосовно російської мови приклад 1.1 полягає в наступному. Кожна буква повідомлення замінюється на іншу, яка в російському алфавіті відстоїть від вихідної на три позиції далі. Таким чином, буква А замінюється на Г, Б на Ґ і так далі аж до букви Щ, яка замінялася на Я, потім Ь на А, Ю на Б і, нарешті, Я на В.

```
АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ
а, б, в, г, ґ, д, е, є, ж, з, и, і, ї, й, к, л, м, н, о, п, р, с, т, у, ф, х, ц, ч, ш, щ, ь, ю, я
```

Лістинг 1.1. Вихідний алфавіт

Так, наприклад, слово ЗМІНА після шифрування методом Цезаря перетвориться в ЇПКРГ.

Це не дуже складний метод, тим більше що при шифруванні повідомлень з декількох слів відразу стає зрозумілим, скільки слів містив вихідний текст. Крім того, можна отримати деяку інформацію з аналізу повторів букв в зашифрованому повідомленні. Наприклад, в

зашифрованому КГПЗРГ одна з букв повторюється двічі. Проте, Цезар увійшов в історію криптографії, а "шифр Юлія Цезаря", як його досі називають, є прикладом однієї з перших систем шифрування.

Для розшифрування повідомлення ЇПКРГ необхідно знати тільки сам алгоритм шифрування. Будь-яка людина, що знає спосіб шифрування, легко може розшифрувати секретне повідомлення. Таким чином, ключем в даному методі є сам алгоритм.

Яким чином можна вдосконалити шифр Цезаря? Можна було б спробувати розширити алфавіт з 33 до 36 символів і більше за рахунок включення розділових знаків і пробілів. Це збільшення алфавіту замаскувало б довжину кожного окремого слова.

У криптографії прийнято вважати, що противник може знати використаний алгоритм шифрування, характер переданих повідомлень і перехоплений шифротекст, але не знає секретний ключ. Як уже згадувалося вище, це називається принципом Керкхоффса. Іноді це правило здається "перестраховкою", але така "перестраховка" аж ніяк не зайва, якщо, наприклад, передаються дані оборонного або державного характеру.

Вдосконалюємо шифр Цезаря з урахуванням правила Керкгоффса.

Припустимо, що букви зсуваються не так на два знака вправо, а на ***n*** (0 <n <33). У цьому випадку в системі шифрування з'являється ключ - число ***n*** - параметр зсуву. Відправник і одержувач можуть якимось чином домовлятися (наприклад, особисто) і іноді міняти значення ключа. Так як ***n*** може приймати різні значення, знання однієї тільки алгоритму не дозволить противнику розшифрувати секретне повідомлення.

Яким же чином може діяти в тому випадку зловмисник, щоб дізнатися зміст повідомлення? Нехай, наприклад, перехоплено секретне повідомлення ЧСЮЕЮ'. Противнику відомо, що ключ (параметр зсуву ***n***) може набувати значень від 1 до 32. Намагаючись знайти значення секретного ключа, ми будемо проводити атаку по шифротексту. Розглянемо спосіб послідовного перебору всіх можливих ключів (це так званий метод "грубої сили"). Запишемо на 32 рядках всі варіанти, які виходять зрушенням кожної літери на 1, 2, 3, ..., 32 позиції відповідно. Цю операцію можна проводити вручну, а можна скласти нескладну програму, яка запише всі варіанти перебору параметра ***n*** в файл. Одна з цих 32 рядків буде містити вихідне повідомлення (таблиця 1.1).

а, б, в, г, ґ, д, е, є, ж, з, и, і, ї, й, к, л, м, н, о, п, р, с, т, у, ф, х, ц, ч, ш, щ, ь, ю, я

Таблиця 1.1. Перебор вариантов для поиска ключа при использовании метода Цезаря

Перехваченная криптограмма ЧСЮЭЮЪ			
1	ШТЯЮЯЫ	17	ЗВОНОК
2	ЩУАЯАЬ	18	ИГПОПЛ
3	ЪФБАБЭ	19	ЙДРПРМ
4	ЫХВБВЮ	20	КЕСРСН
5	ЬЦГВГЯ	21	ЛЁТСТО
6	ЭЧДГДА	22	МЖУТУП
7	ЮШЕДЕБ	23	НЗФУФР
8	ЯЩЁЕЁВ	24	ОИХФХС
9	АЪЖЁЖГ	25	ПЙЦХЦТ
10	БЫЗЖЗД	26	РКЧЦЧУ

11	ВЬИЗИЕ	27	СЛШЧШФ
12	ГЭЙИЙЁ	28	ТМЩШЩХ
13	ДЮКЙКЖ	29	УНЪЩЪЦ
14	ЕЯЛКЛЗ	30	ФОЫЪЫЧ
15	ЁАМЛМИ	31	ХПЬЫЬШ
16	ЖБНМНЙ	32	ЦРЭЬЭЩ

Ми бачимо, що єдине слово, яке має сенс, - це ЗВОНОК. Це слово розташовується на 17-му місці. Отже, якщо зашифровані текст зрушити на 17 позицій вперед вийде відкритий текст. Це означає, що для отримання шифрованого тексту відкритий текст потрібно зрушити на (33-17) = 16 позицій. Таким чином, отримали, що при шифруванні ключ *n=16*.

Так як ні при якому іншому зсуві не вийшло осмисленого повідомлення, то, швидше за все, ми правильно дешифрували це повідомлення. Таке припущення про єдине рішення цілком обгрунтовано, коли вихідне повідомлення складено на одному з природних мов (в розглянутому прикладі - російською) і містить більше п'яти-шести знаків. Але якщо повідомлення дуже коротке, можливих рішень може бути кілька. Єдине рішення також дуже важко знайти, якщо вихідне повідомлення, складається, наприклад, з цифр.

Так, наприклад, нехай вихідний алфавіт складається з арабських цифр, тобто має вигляд:

```
0 1 2 3 4 5 6 7 8 9.
```

Один з абонентів бажає переслати іншому секретний код замка, що складається з п'яти цифр і рівний 12345. Відправник і одержувач заздалегідь домовилися про те, що ключ шифрування ***n*** дорівнюватиме 3. Відправник шифрує обраним ключем вихідне повідомлення 12345, отримує 45678 і переправляє отримане значення своєму абоненту . Можливо, противник перехопить криптограму і спробує розкрити її, використовуючи, як і раніше, метод послідовного перебору. Так як вихідний алфавіт складався з 10 символів, то значення ключа може лежати в діапазоні від 1 до 9 випишемо, як і раніше всі варіанти, які виходять зрушенням кожного знака перехопленого повідомлення на 1, 2, 3, ..., 9 позицій відповідно (таблиця 1.2).

Таблиця 1.2. Перебор вариантов для вскрытия зашифрованного кода замка

Перехваченная криптограмма 45678	
1	56789
2	67890
3	78901
4	89012
5	90123
6	01234
7	12345
8	23456
9	34567

Видно, що всі отримані варіанти рівнозначні і зловмисник не може зрозуміти, яка саме комбінація істинна. Аналізуючи шифротекст, він не може знайти значення секретного ключа. Звичайно, один з наведених у таблиці варіантів підійде до кодовому замку, але в настільки простому методі шифрування не можна розраховувати на більшу таємничість.

У першому прикладі повідомлення - текст російською мовою, тому воно підпорядковується численним правилами, різні літери і їх поєднання мають різні ймовірності і, зокрема, багато набори букв взагалі заборонені. (Ця властивість називається надмірністю тексту). Тому-то і вдалося легко підібрати ключ і дешифрувати повідомлення, тобто надмірність дозволила "зламати" шифр. На противагу цьому, в другому прикладі всі комбінації цифр допустимі. "Мова" кодового замка не містить надмірності. Тому навіть простий шифр, застосований до повідомлень цією мовою, стає нерозшифровуваним в разі атаки тільки по шифротексту. Якщо ж ми маємо можливість проводити атаку і з відкритого тексту, тобто маємо пари "відкрите повідомлення" - "зашифроване повідомлення", то розкриття стає абсолютно простим як у випадку використання символів-букв, так і в разі символів-цифр.

Наведені прості приклади показують, що ймовірність успішного криптоаналізу залежить від багатьох чинників: від системи шифрування, від довжини перехопленого повідомлення, від мови і алфавіту вихідного повідомлення.

Криптографічний протокол

В сучасної криптографії велика увага приділяється не тільки створенню і дослідженню шифрів, а й розробці криптографічних протоколів.

Криптографічний протокол - це така процедура взаємодії двох або більше абонентів з використанням криптографічних засобів, в результаті якої абоненти досягають своєї мети, а їх противники - не досягають. В основі протоколу лежить набір правил, що регламентують використання криптографічних перетворень і алгоритмів в інформаційних процесах. Кожен криптографічний протокол призначений для вирішення певної задачі.

Будь-який протокол має такі властивості:

- • при виконанні протоколу важливий порядок дій; кожна дія має виконуватися в свою чергу і тільки після закінчення попереднього;
- • протокол повинен бути несуперечливим;
- • протокол повинен бути повним, тобто для кожної можливої ситуації має бути передбачена відповідна дія.

Властивості протоколу нагадують властивості алгоритму. Насправді протокол - це і є алгоритм дії декількох сторін в певній ситуації. Учасники протоколу повинні знати протокол і виконувати повністю всі його етапи. Учасниками криптографічного протоколу зазвичай є абоненти деякої системи зв'язку. Учасники протоколу можуть не довіряти один одному, тому криптографічні протоколи повинні захищати їх учасників не тільки від зовнішнього противника, а й від нечесних дій партнерів.

Криптографічні протоколи - порівняно молода галузь криптографічної науки. Перші протоколи з'явилися в другій половині XX століття. З тих пір ця область криптографії бурхливо розвивалася, і на даний момент є вже кілька десятків різних типів криптографічних протоколів. Всі ці типи можна умовно розділити на дві групи: прикладні протоколи і примітивні. *Прикладний протокол* вирішує конкретну задачу, яка виникає (або може виникнути) на практиці. *Примітивні* ж протоколи використовуються як своєрідні "будівельні блоки" при розробці прикладних протоколів. Ми будемо розглядати тільки примітивні криптографічні протоколи, які при деякій адаптації до реальних систем зв'язку можуть використовуватися на практиці.

Розглянемо призначення деяких видів протоколів.

1. *Протоколи конфіденційної передачі повідомлень*. Завдання конфіденційної передачі повідомлень полягає в наступному. Є два учасники протоколу, які є абонентами мережі зв'язку. Учасники з'єднані деякою лінією зв'язку, по якій можна пересилати повідомлення в обидві сторони. Лінію зв'язку може контролювати противник. У одного з абонентів є конфіденційне повідомлення ***m***, і завдання полягає в тому, щоб це повідомлення конфіденційним же чином передати другому абоненту. Протоколи цього типу, напевно, з'явилися раніше від інших криптографічних протоколів, так як завдання конфіденційної передачі повідомлень - історично перша задача, яка вирішувалася криптографією.
2. *Протоколи аутентифікації і ідентифікації*. Вони призначені для запобігання доступу до деякої інформації осіб, які не є її користувачами, а також запобігання доступу користувачів до тих ресурсів, на які у них немає повноважень. Типова сфера застосування - організація доступу користувачів до ресурсів деякої великої інформаційної системи.
3. *Протоколи розподілу ключів* необхідні для забезпечення секретними ключами учасників обміну зашифрованими повідомленнями.
4. *Протоколи електронного цифрового підпису* дозволяють ставити під електронними документами підпис, аналогічну звичайної підписи на паперових документах. В результаті виконання протоколу електронного цифрового підпису до переданої інформації додається унікальне числове додаток, що дозволяє перевірити її авторство.
5. Протоколи забезпечення невідслідкованості ("Електронні гроші"). Під електронними грошима в криптографії розуміють електронні платіжні засоби, що забезпечують невідслідкованості, тобто неможливість простежити джерело пересилання інформації.

Розглянемо найпростіший протокол для обміну конфіденційними повідомленнями між двома сторонами, які будемо називати абонент №1 і абонент №2. Нехай абонент №1 бажає передати зашифроване повідомлення абоненту №2. У цьому випадку їх послідовність дій повинна бути наступною.

1. Абоненти вибирають систему шифрування (наприклад, шифр Цезаря із зсувом на ***n*** позицій).
2. Абоненти домовляються про ключі шифрування.
3. Абонент №1 шифрує вихідне повідомлення за допомогою ключа обраним методом і отримує зашифроване повідомлення.
4. Зашифроване повідомлення пересилається абоненту №2.
5. Абонент №2 розшифровує зашифроване повідомлення за допомогою ключа і отримує відкрите повідомлення.

Цей протокол досить простий, однак він може дійсно використовуватися на практиці. Криптографічні протоколи можуть бути простими і складними в залежності від призначення.

Раніше ми ввели поняття криптографічної атаки і розглянули типи атак на криптографічний алгоритм. У багатьох випадках атака може бути спрямована не на алгоритм шифрування, а на протокол. Тому навіть наявність абсолютно надійного алгоритму шифрування не гарантує повної безпеки абонентам системи зв'язку. Відомі випадки, коли застосовуються на практиці криптографічні протоколи містили вади, що допускають шахрайство сторін або розтин з боку активного зломщика. Звичайно, криптографічні протоколи не повинні допускати таких можливостей порушників. Саме тому в даний час криптографічні протоколи є предметом ретельного аналізу з боку фахівців.

Ключові терміни

Ciphertext – зашифроване повідомлення (закритий текст, криптогрма)

Deciphering – розшифрування.

Enciphering – перетворення відкритого тексту в криптограму (зашифрування).

Plaintext – вихідне повідомлення або відкритий текст.

Активна криптографічний атака - при такій атаці противник має можливість модифікувати передані повідомлення і навіть додавати свої повідомлення.

Алфавіт - кінцева кількість використовуваних для кодування інформації символів.

Ключ - інформація, необхідна для шифрування і розшифрування повідомлень.

Криптоаналіз - наука про подолання криптографічного захисту інформації.

Криптографічний система захисту інформації - система захисту інформації, в якій використовуються криптографічні методи для шифрування даних.

Криптографічний протокол - алгоритм взаємодії двох або більше абонентів з використанням криптографічних засобів, в результаті якої абоненти досягають своєї мети, а їх противники - не досягають.

Криптографія вивчає побудову і використання систем шифрування, в тому числі їх стійкість, слабкості і ступінь уразливості щодо різних методів розкриття.

Крипостійкість - характеристика шифру, що його стійкість до дешифрування без знання ключа (тобто здатність протистояти криптоанализу).

Пасивна криптографічний атака - атака, при якій супротивник не має можливості змінювати передані повідомлення. При пасивної атаці можливо лише прослуховування повідомлень, що передаються, їх дешифрування і аналіз трафіку.

Принцип Керкхоффса - правило розробки криптографічних систем, згідно з яким в засекреченому вигляді тримається ключ шифрування, а інші параметри системи шифрування можуть бути відкриті без зниження стійкості алгоритму. Іншими словами, при оцінці надійності шифрування необхідно припускати, що противник знає про використовувану системі шифрування все, крім застосовуваних ключів. Вперше даний принцип сформулював в XIX столітті голландський криптограф Огюст Керкхоффс.

Символ - це будь-який знак, в тому числі буква, цифра або знак пунктуації.

Система шифрування, або шифросистема, - це будь-яка система, яку можна використовувати для оборотного зміни тексту повідомлення з метою зробити його незрозумілим для всіх, крім тих, кому воно призначене.

Шифр - сукупність заздалегідь обумовлених способів перетворення вихідного секретного повідомлення з метою його захисту.

Шифрування з закритим ключем (симетричне шифрування) - методи оборотного перетворення даних, в яких використовується один і той же ключ, який обидві сторони інформаційного обміну повинні зберігати в секреті від противника. Всі відомі з історії шифри, наприклад, шифр Цезаря - це шифри з закритим ключем.

Шифрування з відкритим ключем (асиметричне шифрування) - методи шифрування, в яких для шифрування і розшифрування даних використовуються два різних ключа. При цьому один з ключів (відкритий ключ) може передаватися по відкритому (незахищеному) каналу зв'язку. Шифрування з відкритим ключем використовується на практиці лише з другої половини XX століття.

Електронний (цифровий) підпис - приєднується до повідомлення блок даних, отриманий з використанням криптографічного перетворення. Електронний підпис дозволяє при отриманні тексту іншим користувачем перевірити авторство і достовірність повідомлення.

Короткі підсумки

Криптографія в перекладі з грецького означає "тайнопис". В даний час криптографія займається пошуком і дослідженням математичних методів перетворення інформації. Паралельно розвивається і вдосконалюється криптоаналіз - наука про подолання криптографічного захисту інформації.

Криптографія вирішує наступні завдання: шифрування даних з метою захисту від несанкціонованого доступу; перевірка справжності повідомлень; перевірка цілісності переданих даних; забезпечення неможливості відмови.

Для сучасних криптографічних систем захисту інформації сформульовані наступні вимоги:

- • зашифроване повідомлення повинно піддаватися читання тільки при наявності ключа;
- • знання алгоритму шифрування не повинно впливати на надійність захисту;
- • будь-який ключ з безлічі можливих повинен забезпечувати надійний захист інформації;
- • алгоритм шифрування повинен допускати як програмну, так і апаратну реалізацію.

Криптографічні алгоритми можуть бути реалізовані апаратно або програмно. На розробку апаратного пристрою необхідні істотні витрати, однак при масовому випуску пристрою ці витрати окупаються. Апаратна реалізація криптографічного методу відрізняється високою швидкістю обробки даних, простотою в експлуатації, захищеністю. Програмні реалізації криптографічних алгоритмів відрізняються істотно меншою швидкодією. Випускаються також і комбіновані модулі шифрування, так звані програмно-апаратні засоби.

Інформація в процесі зберігання, передачі і перетворення піддається впливу атак. Основними порушеннями інформаційної безпеки є розкриття інформаційних цінностей (втрата конфіденційності), модифікація без дозволу автора (втрата цілісності) або неавторизована втрата доступу до цих цінностей (втрата доступності).

Криптографічні атаки можуть бути пасивними і активними. Залежно від кількості і типу інформації, доступної для криптоаналізу, виділяють атаки на основі шифротекста, атаки на основі відомого відкритого тексту, атаки на основі обраного відкритого тексту.

Голландський криптограф Огюст Керкхоффс сформулював принцип, званий в даний час **принципом Керкхоффса** - правило розробки криптографічних систем, згідно з яким в засекреченому вигляді тримається ключ шифрування, а інші параметри системи шифрування можуть бути відкриті без зниження стійкості алгоритму. Іншими словами, при оцінці надійності шифрування необхідно припускати, що противник знає про використовувану систему шифрування все, крім застосовуваних ключів. Принцип Керкхоффса спрямований на те, щоб зробити безпеку алгоритмів і протоколів незалежною від їх секретності; відкритість не повинна впливати на безпеку. Більшість широко використовуваних систем шифрування, відповідно до

принципу Керкхоффса, використовують відомі, що не становлять секрету криптографічні алгоритми.

В сучасної криптографії велика увага приділяється розробці криптографічних протоколів, тобто процедур або алгоритмів взаємодії абонентів з використанням криптографічних засобів. В основі протоколу лежить набір правил, що регламентують використання криптографічних перетворень. Деякі види протоколів: протоколи конфіденційної передачі повідомлень; протоколи аутентифікації і ідентифікації; протоколи розподілу ключів; протоколи електронного цифрового підпису; протоколи забезпечення невідслідковуваності.

Питання для самоперевірки

1. Назвіть проблеми, при вирішенні яких можуть використовуватися криптографічні методи.
2. У чому відмінність криптографії від стеганографії?
3. Які завдання вирішує сучасна криптографія?
4. Сформулюйте вимоги до криптографічних систем захисту інформації.
5. Дайте визначення поняттям: алфавіт, криптограма, криптографічний система, криптографічний протокол, символ, шифр, електронна (цифрова) підпис.
6. У чому полягає правило шифрування методом Цезаря?
7. Чому неможливо розкрити криптограму, що містить код для кодового замка?
8. Чому проблема використання криптографічних методів у інформаційних системах стала зараз особливо актуальною?
9. Що таке криптографічний атака?
10. Які типи криптографічних атак існують?
11. Що таке криптографічний протокол?
12. Поясніть призначення наступних криптографічних протоколів:
- обміну конфіденційними повідомленнями;
- формування електронного цифрового підпису;
- розподілу ключів.Упражнения для самопроверки

Вправи для самоперевірки

1. Визначити ключі Цезаря, якщо відомі наступні пари відкритий текст – шифротекст (вихідний алфавіт:**А Б В Г Ґ Д Е Є Ж З И І Ї Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ь Ю Я**):
    - АПЕЛЬСИН - ПДХБМЄЩГ
    - МАНДАРИН – СДТИДХЛТ
2. Розшифруйте наступні повідомлення, зашифровані шифром Цезаря, і визначте ключ n, 0<n<33, якщо відомо, що вихідні повідомлення складені з алфавіту:

А Б В Г Ґ Д Е Є Ж З И І Ї Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ь Ю Я:

    - УЧЦЬШЛУЮ
    - ИБЗСЕЧ

Лекція 2
НАЙПРОСТІШІ МЕТОДИ ШИФРУВАННЯ ІЗ ЗАКРИТИМ КЛЮЧЕМ

Загальна схема симетричного шифрування

Класична, або одноключова криптографія спирається на використання симетричних алгоритмів шифрування, в яких шифрування і розшифрування відрізняються тільки порядком виконання і напрямом деяких кроків. Ці алгоритми використовують один і той же секретний елемент (ключ), і друга дія (розшифрування) є простим зверненням першого (шифрування). Тому зазвичай кожен з учасників обміну може як зашифрувати, так і розшифрувати повідомлення. Схематична структура такої системи представлена на рис. 2.1.

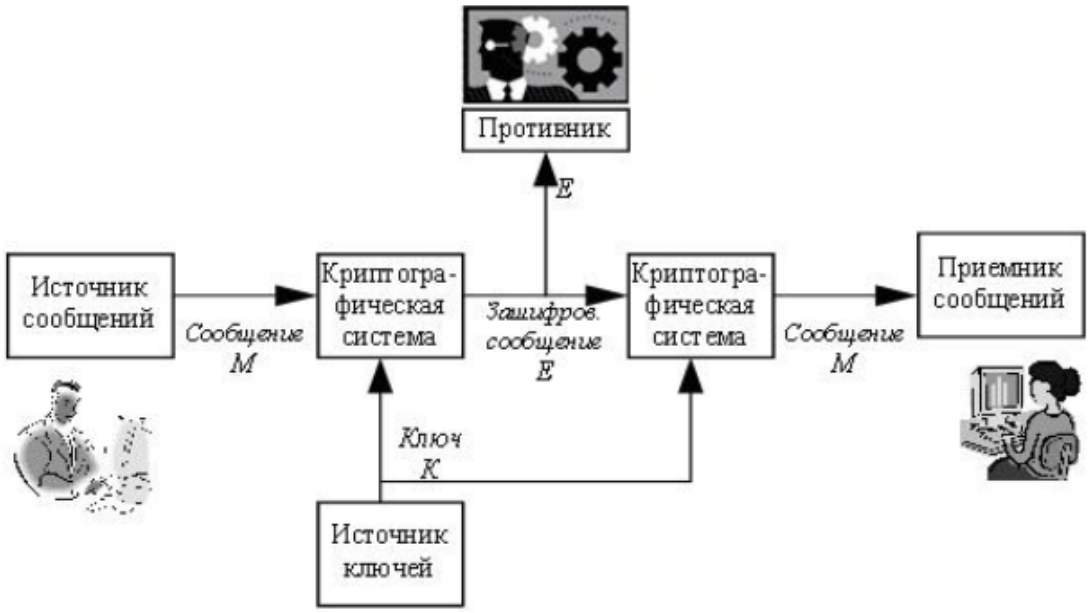


Рис. 2.1. Загальна структура секретної системи, що використовує симетричне шифрування

На передавальній стороні є джерело повідомлень і джерело ключів. Джерело ключів вибирає конкретний *ключ* ***K*** серед усіх можливих ключів даної системи. Цей *ключ* ***K*** передається деяким способом приймаючій стороні, причому передбачається, що його не можна перехопити, наприклад, ключ передається спеціальним кур'єром (тому симетричне шифрування називається також шифруванням із закритим ключем). Джерело повідомлень формує деяке *повідомлення* ***M,*** яке потім шифрується з використанням обраного ключа. В результаті процедури шифрування виходить зашифроване *повідомлення* ***E*** (зване також криптограмою). Далі криптограма ***E*** передається по каналу зв'язку. Так як канал зв'язку є відкритим, незахищеним, наприклад, радіоканал або комп'ютерна мережа, то передане повідомлення може бути перехоплено противником. На приймаючій стороні криптограму ***E*** за допомогою ключа розшифровують і отримують вихідне повідомлення ***M.***

Якщо ***M*** - повідомлення, ***K*** - ключ, а ***E*** - зашифроване повідомлення, то можна записати

$$E = f(M, K)$$

тобто зашифроване повідомлення ***E*** є деякою функцією від вихідного повідомлення ***M*** і ключа ***K***. Використовуваний в криптографічній системі метод або алгоритм шифрування і визначає функцію ***f*** в наведеній вище формулі.

З причини великої надмірності природних мов безпосередньо в зашифроване повідомлення надзвичайно важко внести осмислену зміну, тому класична криптографія забезпечує також захист від нав'язування помилкових даних. Якщо ж природної надмірності

виявляється недостатньо для надійного захисту повідомлення від модифікації, надмірність може бути штучно збільшена шляхом додавання до повідомлення спеціальної контрольної комбінації, названої *імітовставкою*.

Відомі різні методи шифрування із закритим ключем рис. 2.2. На практиці часто використовуються алгоритми перестановки, підстановки, а також комбіновані методи.

Рис. 2.2. Методи шифрування із закритим ключем

У методах перестановки символи вихідного тексту міняються місцями один з одним за певним правилом. У методах заміни (або підстановки) символи відкритого тексту замінюються деякими еквівалентами шифрованого тексту. З метою підвищення надійності шифрування текст, зашифрований за допомогою одного методу, може бути ще раз зашифровано за допомогою іншого методу. В цьому випадку виходить комбінований або композиційний шифр. Застосовувані на практиці в даний час блокові або потокові симетричні шифри також відносяться до комбінованих, так як в них використовується кілька операцій для шифрування повідомлення.

Основна відмінність сучасної криптографії від криптографії "докомп'ютерної" полягає в тому, що раніше криптографічні алгоритми оперували символами природних мов, наприклад, буквами різних алфавітів. Ці букви переставлялися або замінялися іншими за певним правилом. У сучасних криптографічних алгоритмах використовуються операції над двійковими знаками, тобто над нулями і одиницями. В даний час основними операціями при шифруванні також є перестановка або підстановка, причому для підвищення надійності шифрування ці операції застосовуються разом (комбінуються) і багато раз циклічно повторюються.

Принципи побудови сучасних блокових шифрів сформульовані в "Принципи побудови блокових шифрів із закритим ключем", "Алгоритми шифрування DES і AES", "Алгоритм криптографічного перетворення даних ГОСТ 28147-89", а в цій лекції розглядаються шифри підстановки і перестановки, застосовувані людиною з найдавніших часів. Ми повинні познайомитися з цими шифрами, так як процедури підстановки і перестановки використовуються в якості складових операцій і в сучасних блокових шифрах.

Методи заміни

Методи шифрування заміною (підстановкою) засновані на тому, що символи вихідного тексту, зазвичай розділені на блоки і записані в одному алфавіті, замінюються одним або декількома символами іншого алфавіту відповідно до прийнятого правилом перетворення.

Одноалфавітна заміна

Одним з важливих підкласів методів заміни є одноалфавітні (або моноалфавітні) підстановки, в яких встановлюється однозначна відповідність між кожним знаком a_i вихідного алфавіту повідомлень A і відповідним знаком e_i зашифрованого тексту E. Одноалфавітна підстановка іноді називається також простою заміною, так як є найпростішим шифром заміни .

Прикладом одноалфавітної заміни є шифр Цезаря, розглянутий раніше. У розглянутому в "Основні поняття криптографії" прикладі перший рядок є вихідним алфавітом, другий (з циклічним зрушенням на k вліво) - вектором замін.

У загальному випадку при одноалфавітній підстановці відбувається однозначна заміна вихідних символів їх еквівалентами з вектора замін (або таблиці замін). При такому методі шифрування ключем є використовувана таблиця замін.

Підстановка може бути задана за допомогою таблиці, наприклад, як показано на рис. 2.3.

Відкр. текст	Шифр 1	Шифр 2		Відкр. текст	Шифр 1	Шифр 2		Відкр. текст	Шифр 1	Шифр 2
А	В	^		**І**	Т	№		**У**	Д	∑
Б	І	@		**Ї**	Ц	$		**Ф**	И	☼
В	О	)		**Й**	.	-		**Х**	У	♫
Г	А	+		**К**	Ж	=		**Ц**	Н	☻
Ґ	Щ	<		**Л**	Г	(		**Ч**	Ю	₣
Д	П	>		**М**	Л	?		**Ш**	Є	Ω
Е	К	#		**Н**	Х	%		**Щ**	Ш	€
Є	Б	■		**О**	С	©		**Ь**	Е	π
Ж	Ї	*		**П**	Ь	!		**Ю**	Ф	έ
З	пробіл	♥		**Р**	Ч	§		**Я**	Ґ	Δ
И	Р	♠		**С**	З	®		**пробіл**	Й	¥
				Т	М	♦		**.**	Я	♣

Рис. 2.3. Приклад таблиці замін для двох шифрів

У таблиці на рис. 2.3 насправді об'єднані відразу дві таблиці. Одна (шифр 1) визначає заміну українських букв вихідного тексту на інші українські букви, а друга (шифр 2) - заміну букв на спеціальні символи. Вихідним алфавітом для обох шифрів будуть великі українські літери, пробіл і крапка.

Зашифроване повідомлення з використанням будь-якого шифру моноалфавитної підстановки виходить таким чином. Береться черговий знак з вихідного повідомлення. Визначається його позиція в стовпці "Одкр. Текст" таблиці замін. У зашифроване повідомлення вставляється шифрований символ з цього ж рядка таблиці замін.

Спробуємо зашифрувати повідомлення "ВИШЛІТЬ ПІДМОГУ" с використанням цих двох шифрів (рис. 2.4). Для цього беремо першу букву вихідного повідомлення "В". У таблиці на рис. 2.3 в стовпці "Шифр 1" знаходимо для букви "В" замінний символ. Це буде буква "О". Записуємо букву "О" під літерою "В". Потім розглядаємо другий символ вихідного повідомлення - букву "И". Знаходимо цю букву в стовпці "Одкр. Текст" і з шпальти "Шифр 1" беремо букву, що стоїть на тому ж рядку, що і буква "И". Таким чином отримуємо другий символ зашифрованого повідомлення - букву "Р". Продовжуючи діяти аналогічно, зашифровувати все вихідне повідомлення (рис. 2.4).

Відкрите повідомлення														
В	И	Ш	Л	І	Т	Ь		П	І	Д	М	О	Г	У
Зашифроване повідомлення з використанням шифру 1														
О	Р	Є	Г	Т	М	Е	Й	Ь	Т	П	Л	С	А	Д
Зашифроване повідомлення з використанням шифру 2														
)	♠	Ω	(	№	♦	π	¥	!	№	>	?	©	+	∑

Рис. 2.4. Пример шифрования методом прямой замены

Отриманий таким чином текст має порівняно низький рівень захисту, так як вихідний і зашифрований тексти мають однакові статистичні закономірності. При цьому не має значення, які символи використані для заміни - примішані символи вихідного алфавіту чи таємничі знаки.

Зашифроване повідомлення може бути розкрите шляхом так званого частотного криптоаналізу. Для цього можуть бути використані деякі статистичні дані мови, якою написано повідомлення.

Відомо, що в текстах російською мовою найбільш часто зустрічаються символи О, І. Трохи рідше зустрічаються букви Е, А. З приголосних найчастіші символи Т, Н, Р, С. У розпорядженні криптоаналітиків є спеціальні таблиці частот зустрічальності символів для текстів різних типів - наукових, мистецьких і т.д.

Криптоаналітика уважно вивчає отриману криптограму, підраховуючи при цьому, які символи скільки разів зустрілися. Спочатку найбільш часто зустрічаються знаки зашифрованого повідомлення замінюються, наприклад, буквами О. Далі виробляється спроба визначити місця для букв І, Е, А. Потім підставляються найбільш часто зустрічаються приголосні. На кожному етапі оцінюється можливість "поєднання" тих чи інших букв. Наприклад, в російських словах важко знайти чотири поспіль голосні літери, слова в російській мові не починаються з літери И і т.д. Насправді для кожного природної мови (російської, англійської і т.д.) існує безліч закономірностей, які допомагають розкрити фахівцеві зашифровані противником повідомлення.

Можливість однозначного криптоаналізу безпосередньо залежить від довжини перехопленого повідомлення. Подивимося, з чим це пов'язано. Нехай, наприклад, в руки криптоаналітиків потрапило зашифроване за допомогою деякого шифру одноалфавітной заміни повідомлення:

ТНФЖ.ИПЩЪРЪ

Це повідомлення складається з 11 символів. Нехай відомо, що ці символи складають ціле повідомлення, а не фрагмент більш великого тексту. У цьому випадку наше зашифроване повідомлення складається з одного або декількох цілих слів. У зашифрованому повідомленні символ Ъ зустрічається 2 рази. Припустимо, що у відкритому тексті на місці зашифрованого знака Ъ стоїть голосна О, А, І чи Є. Підставимо на місце Ъ ці букви і оцінимо можливість подальшого криптоаналізу (таблиця 2.1)

Таблица 2.1. Варианты первого этапа криптоанализа

Зашифрованное сообщение										
Т	Н	Ф	Ж	.	И	П	Щ	Ъ	Р	Ъ
После замены Ъ на О										

								О		О
После замены Ъ на А										
								А		А
После замены Ъ на И										
								И		И
После замены Ъ на Е										
								Е		Е

Всі наведені варіанти заміни можуть зустрітися на практиці. Спробуємо підібрати якісь варіанти повідомлень, враховуючи, що в криптограмі інші символи зустрічаються по одному разу (таблиця 2.2).

Таблица 2.2. Варианты второго этапа криптоанализа										
Зашифрованное сообщение										
Т	Н	Ф	Ж	.	И	П	Щ	Ъ	Р	Ъ
Варианты подобранных дешифрованных сообщений										
Ж	Д	И		С	У	М	Р	А	К	А
Д	Ж	О	Н	А		У	Б	И	Л	И
В	С	Е	Х		П	О	Б	И	Л	И
М	Ы		П	О	Б	Е	Д	И	Л	И

Крім представлених в таблиці 2.2 повідомлень можна підібрати ще велику кількість відповідних фраз. Таким чином, якщо нам нічого не відомо заздалегідь про зміст перехопленого повідомлення малої довжини, дешифрувати його однозначно не вийде.

Якщо ж у руки криптоаналітиків потрапляє досить довге повідомлення, зашифроване методом простої заміни, його зазвичай вдається успішно дешифрувати. На допомогу фахівцям з розкриття криптограм приходять статистичні закономірності мови. Чим довше зашифроване повідомлення, тим більша ймовірність його однозначного дешифрування.

Цікаво, що якщо спробувати замаскувати статистичні характеристики відкритого тексту, то завдання розшифрування шифру простої заміни значно ускладниться. Наприклад, з цією метою можна перед шифруванням "стискати" відкритий текст з використанням комп'ютерних програм-архіваторів.

З ускладненням правил заміни збільшується надійність шифрування. Можна заміняти не окремі символи, а, наприклад, дволітерні поєднання - біграми. Таблиця замін для такого шифру може виглядати, як на таблиця 2.3.

Таблица 2.3. Пример таблицы замен для двухбуквенных сочетаний			
Откр. текст	**Зашифр. текст**	**Откр. текст**	**Зашифр. текст**
аа	кх	бб	пш
аб	пу	бв	вь
ав	жа	...	...
...	...	яэ	сы
ая	ис	яю	ек
ба	цу	яя	рт

Оцінимо розмір такої таблиці замін. Якщо вихідний алфавіт містить N символів, то вектор замін для біграммного шифру повинен містити N^2 пар "відкр. текст - зашифров. текст". Таблицю замін для такого шифру можна також записати і в іншому вигляді: заголовки стовпців відповідають першій букві біграми, а заголовки рядків - другій, причому елементи таблиці заповнені замінюють символами. В такій таблиці буде N рядків і N стовпців (таблиця 2.4).

Таблица 2.4. Другой вариант задания таблицы замен для биграммного шифра

	а	**б**	**...**	**я**
а	кх	цу	...	...
б	пу	пш	...	...
в	жа	вь	...	...
...	...	...	...	...
ю	...	...	...	ек
я	ис	...	...	рт

Можливі варіанти використання триграммного або взагалі *n*-граммного шифру. Такі шифри володіють вищою криптостійкістю, але вони складніші для реалізації і вимагають набагато більшої кількості ключової інформації (великий обсяг таблиці замін). В цілому, всі *n*-грамні шифри можуть бути розкриті за допомогою частотного криптоаналізу, тільки використовується статистика зустрічаємості не окремих символів, а сполучень з *n* символів.

Пропорційні шифри

До одноалфавітних методів підстановки відносяться **пропорційні** або **монофонічні шифри**, в яких зрівнюється частота появи зашифрованих знаків для захисту від розкриття за допомогою частотного аналізу. Для знаків, що зустрічаються часто, використовується відносно велике число можливих еквівалентів. Для менш використовуваних вихідних знаків може виявитися достатнім одного або двох еквівалентів. При шифруванні заміна для символу відкритого тексту вибирається або випадковим, або певним чином (наприклад, по порядку).

При використанні пропорційного шифру в якості заміни символам зазвичай вибираються числа. Наприклад, поставимо у відповідність буквам української мови тризначні числа, як зазначено на таблиця 2.5.

Таблица 2.5. Таблиця замін для пропорційного шифру

Символ	Варіанти заміни				Символ	Варіанти заміни				
А	760	128	350	201	Н	763	756	212		
Б	101				О	757	213	765	133	353
В	210	106			П	743	766			
Г	351				Р	134	532			
Ґ	378				С	800	767	105		
Д	129				Т	759	135	214		
Е	761	130	802	352	У	544				
Є	126				Ф	560				
Ж	102				Х	768				
З	753				Ц	545				

И	750	864			Ч	215				
І	762	211	131		Ш	103				
Ї	136				Щ	752				
Й	561				Ь	562				
К	754	764			Ю	570				
Л	132	354			Я	216	104			
М	755	742			Пробіл	751	769	758	801	849 035…

В такому випадку повідомлення: ВЕЛИКИЙ СЕКРЕТ

може бути зашифровано наступним способом:

```
2107611327507548645617518001307541348027 59
```

В даному прикладі варіанти замін для букв, що повторюються вибиралися попорядку.

Пропорційні шифри більш складні для розкриття, ніж шифри простої одноалфавітної заміни. Однак, якщо є хоча б одна пара "відкритий текст - шифротекст", розтин проводиться тривіально. Якщо ж у наявності є тільки шифротекст, то розтин ключа, тобто знаходження таблиці замін, стає більш трудомістким, але теж цілком здійсненним.

Багатоалфавітної підстановки

З метою маскування природної частотної статистики вихідної мови застосовується *багатоалфавітна підстановка*, яка також буває декількох видів. У багатоалфавітних підстановках для заміни символів вихідного тексту використовується не один, а кілька алфавітів. Зазвичай алфавіти для заміни утворені із символів вихідного алфавіту, записаних в іншому порядку.

Прикладом багатоалфавітної підстановки може служити схема, заснована на використанні таблиці Віжинера. Цей метод, відомий уже в XVI столітті, був описаний французом Блез Вижинером в "Трактаті про шифри", що вийшов в 1585 році.

У цьому методі для шифрування використовується таблиця, що представляє собою квадратну матрицю з числом елементів ***NxN***, де ***N*** - кількість символів в алфавіті (таблиця 2.6). У першому рядку матриці записують букви в порядку черговості їх у вихідному алфавіті, у другому - ту ж послідовність букв, але з циклічним зсувом вліво на одну позицію, в третьому - із зсувом на дві позиції і т. д.

Таблица 2.6. Подготовка таблицы шифрования
АБВГДЕ.....ЭЮЯ
БВГДЕЖ.....ЮЯА
ВГДЕЖЗ.....ЯАБ
ГДЕЖЗИ.....АБВ
ДЕЖЭИК.....БВГ
ЕЖЗИКЛ.....ВГД
.....
ЯАБВГД.....ЬЭЮ

Для шифрування тексту вибирають ключ, що представляє собою деяке слово або набір символів вихідного алфавіту. Далі з повної матриці виписують підматрицю шифрування, що включає перший рядок і рядки матриці, початковими буквами яких є послідовно букви ключа (наприклад, якщо вибрати ключ "весна", то таблиця шифрування буде такою (таблиця 2.7).

Таблиця 2.7. Первый этап шифрования – составление подматрицы шифрования

АБВГДЕЖЗИКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ
ВГДЕЖЗИКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯАБ
ЕЖЗИКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯАБВГД
НОПРСТУФХЦЧШЩЪЫЬЭЮЯАБВГДЕЖЗИКЛМ
СТУФХЦЧШЩЪЫЬЭЮЯАБВГДЕЖЗИКЛМНОПР

У процесі шифрування (рис. 2.5) під кожною буквою шифрованого тексту записують букви ключа, що повторюють ключ необхідне число разів. Потім шифрований текст по таблиці шифрування (таблиця 2.7) замінюють буквами, розташованими на перетинах ліній, що з'єднують букви тексту першого рядка таблиці і букви ключа , що знаходиться під нею.

Наприклад, під першою літерою початкового тексту "М" записана буква "В" ключа. У таблиці кодування знаходимо стовпець, що починається з "М" і рядок, що починається з "В". На їх перетині розташовується буква "О". Вона і буде першим символом зашифрованого повідомлення (на рис. 2.5 ця літера виділена прямокутної рамочкою). Наступна буква вихідного повідомлення - "Е", символ ключа - теж "Е". Знаходимо перетинання рядка, що починається з "Е", і шпальти, що починається з "Е". Це буде буква "Л" - другий символ зашифрованого повідомлення.

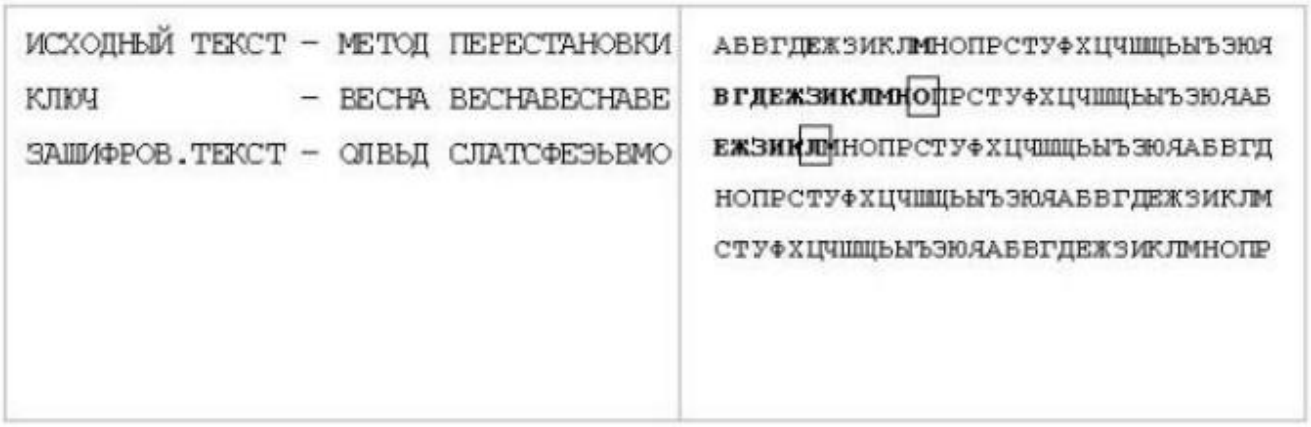


Рис. 2.5. Механізм шифрування багатоалфавітною заміною

ВИХІДНИЙ ТЕКСТ	М	Е	Т	О	Д		П	Е	Р	Е	С	Т	А	Н	О	В	К	И
КЛЮЧ	В	Е	С	Н	А		В	Е	С	Н	А	В	Е	С	Н	А	В	Е
ЗАШИФРОВАНИЙ ТЕКСТ	о	ї	и	в	д		с	ї	ж	у	с	ф	е	д	в	в	м	м

а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я
в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б
е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д
с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р

н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м

Рис. 2.5. Механізм шифрування багатоалфавітною заміною

Розглянемо на прикладі процес розшифрування повідомлення за методом Віжинера. Нехай є зашифроване за допомогою ключа ВЕСНА повідомлення ІЕЙЧСФНДЖОТТСИІК (пропуски при шифруванні пропущені). Розшифровка тексту виконується в наступній послідовності (таблиця 2.8):

- над літерами зашифрованого тексту зверху послідовно записують букви ключа, повторюючи ключ необхідне число разів;
- в рядку підматриці таблиці Віжинера для кожної букви ключа відшукується буква, відповідна знаку зашифрованого тексту. Що знаходиться над нею буква першого рядка і буде знаком розшифрованого тексту;
- отриманий текст групується в слова за змістом.

Таблиця 2.8 – Механізм розшифрування

КЛЮЧ	В	Е	С	Н	А	В	Е	С	Н	А	В	Е	С	Н	А	В
ЗАШИФРОВАНИЙ ТЕКСТ	і	е	й	ч	с	ф	н	д	ж	о	т	т	с	и	і	к
РОЗШИФРОВАНИЙ ТЕКСТ	З	А	Х	И	С	Т	І	Н	Ф	О	Р	М	А	Ц	І	Ї

Розкрити шифр Віжинера, тим же способом, що і шифр одноалфавітної заміни, неможливо, оскільки одні й ті ж символи відкритого тексту можуть бути замінені різними символами зашифрованого тексту. З іншого боку, різні літери відкритого тексту можуть бути замінені однаковими знаками зашифрованого тексту.

Особливість даного методу багатоалфавітної підстановки полягає в тому, що кожен із символів ключа використовується для шифрування одного символу вихідного повідомлення. Після використання всіх символів ключа, вони повторюються в тому ж порядку. Якщо використовується ключ з десяти букв, то кожна десята буква повідомлення шифрується одним і тим же символом ключа. Цей параметр називається періодом шифру. Якщо ключ шифрування складається з одного символу, то при шифруванні буде використовуватися один рядок таблиці Вижинера, отже, в цьому випадку ми отримаємо моноалфавітну підстановку, а саме шифр Цезаря.

З метою підвищення надійності шифрування тексту можна використовувати поспіль два або більше зашифрування за методом Вижинера з різними ключами (складовою шифр Вижинера).

На практиці крім методу Вижинера використовувалися також різні модифікації цього методу. Наприклад, шифр Вижинера з перемішаним один раз алфавітом. У цьому випадку для розшифрування повідомлення одержувачу необхідно окрім ключа знати порядок проходження символів у таблиці шифрування.

Ще одним прикладом методу многоалфавитной підстановки є шифр з біжучим ключем або книжковий шифр. У цьому методі один текст використовується як ключ для шифрування іншого тексту. В епоху "докомп'ютерної" криптографії в якості ключа для шифру з біжучим ключем вибирали якусь досить товсту книгу; від цього і відбулося друга назва цього шифру. Періодом в такому методі шифрування буде довжина обраного в якості ключа твору.

Методи многоалфавитной підстановки, в тому числі і метод Вижинера, значно важче піддаються "ручному" криптоанализу. Для розкриття методів багатоалфавітної заміни розроблені спеціальні, досить складні алгоритми. З використанням комп'ютера розтин методу многоалфавитной підстановки можливо досить швидко завдяки високій швидкості проведених операцій і розрахунків.

Методи гамірування

Ще одним окремим випадком багатоалфавітної підстановки є гамірування. У цьому способі шифрування виконується шляхом складання символів вихідного тексту і ключа по модулю, рівному числу букв в алфавіті. Якщо у вихідному алфавіті, наприклад, 33 символи, то додавання проводиться за модулем 33. Такий процес складання початкового тексту і ключа називається в криптографії накладанням гами.

Нехай символам вихідного алфавіту відповідають числа від 0 (А) до 32 (Я). Якщо позначити число, відповідне вихідного символу, ***x***, а символу ключа - ***k***, то можна записати правило гамірування наступним чином:

$z=x+k(modN)$,

де **z** - закодований символ, ***N*** - кількість символів в алфавіті, а додавання по модулю ***N*** - операція, аналогічна звичайному додаванню, з тією відмінністю, що якщо звичайне сумування дає результат, більший або рівний ***N***, то значенням суми вважається залишок від ділення його на ***N***. Наприклад, нехай складемо по модулю 33 символи Г (3) і Ю (31):

$3 + 31 \pmod{33} = 1$,
тобто в результаті отримуємо символ Б, відповідний числу 1.

Найбільш часто на практиці зустрічається двійкове гамірування. При цьому використовується двійковий алфавіт, а додавання проводиться по модулю два. Операція додавання за модулем 2 часто позначається $\oplus$, тобто можна записати:

$$z = x + k(mod2) = x \oplus k.$$

Операція додавання за модулем два в алгебрі логіки називається також "виключає АБО" або по-англійськи XOR.

Розглянемо приклад. Припустимо, нам необхідно зашифрувати десяткове число 14 методом гамірування з використанням ключа 12. Для цього спочатку необхідно перетворити вихідне число і ключ (гамму) в двійкову форму: $14_{(10)} = 1110_{(2)}$, $12_{(10)} = 1100_{(2)}$. Потім треба записати отримані двійкові числа один під одним і кожну пару символів скласти по модулю два. При складанні двох двійкових знаків виходить 0, якщо вихідні двійкові цифри однакові, і 1, якщо цифри різні:

$$0 \oplus 0 = 0$$
$$0 \oplus 1 = 1$$
$$1 \oplus 0 = 1$$
$$1 \oplus 1 = 0$$

Складемо по модулю два двійкових числа 1110 и 1100:

```
Исходное число 1 1 1 0
Гамма          1 1 0 0
Результат      0 0 1 0
```

В результаті складання отримали двійкове число 0010. Якщо перевести його в десяткову форму, отримаємо 2. Таким чином, в результаті застосування до числа 14 операції гамірування з ключем 12 отримуємо в результаті число 2.

Яким же чином виконується розшифрування? Зашифроване число 2 представляється в двійковому вигляді і знову проводиться додавання по модулю 2 з ключем:

```
Зашифрованное число 0 0 1 0
Гамма               1 1 0 0
Результат           1 1 1 0
```

Переведемо отримане двійкове значення 1110 в десятковий вигляд і отримаємо 14, тобто вихідне число.

Таким чином, при гамірованні за модулем 2 потрібно використовувати одну і ту ж операцію як для зашифрування, так і для розшифрування. Це дозволяє використовувати один і той же алгоритм, а відповідно і одну і ту ж програму при програмній реалізації, як для шифрування, так і для розшифрування.

Операція додавання за модулем два дуже швидко виконується на комп'ютері (на відміну від багатьох інших арифметичних операцій), тому накладання гами навіть на дуже великий відкритий текст виконується практично миттєво.

Завдяки зазначеним перевагам метод гамірування широко застосовується в сучасних технічних системах сам по собі, а також як елемент комбінованих алгоритмів шифрування.

Сформулюємо, як виробляється гамірування за модулем 2 в загальному випадку:

• символи вихідного тексту і гамма представляються в двійковому коді і розташовуються один під іншим, при цьому ключ (гамма) записується стільки разів, скільки буде потрібно;

• кожна пара двійкових знаків складається по модулю два;

• отримана послідовність двійкових знаків кодується символами алфавіту відповідно до обраного коду.

На рис. 2.6 показано, як застосовується гамірування до тексту з російськими символами. Символи кодуються відповідно до прийнятого кодування, а потім проводиться складання по модулю 2.

При використанні методу гамірування ключем є послідовність, з якою проводиться додавання - гама. Якщо гамма коротше, ніж повідомлення, призначене для зашифровування, гамма повторюється необхідну кількість раз. Так у прикладі на рис. 2.6 довжина вихідного повідомлення дорівнює дванадцяти байтам, а довжина ключа - п'яти байтам. Отже, для зашифровування гамма повинна бути повторена 2 рази повністю і ще один раз частково.

Исходный текст: *Гаммирование*

Исходный текст в шестнадцатеричном виде:

83 A0 AC AC A8 E0 AE A2 A0 AD A8 A5

Гамма (Ключ): *Весна (82 A5 E1 AD A0)*

Гаммирование

Исх. биты	1000	0011	1010	0000	1010	1100
Гамма	1000	0010	1010	0101	1110	0001
Результат	0000	0001	0000	0101	0100	1101

Исх. биты	1010	1100	1010	1000	1110	0000
Гамма	1010	1101	1010	0000	1000	0010
Результат	0000	0001	0000	1000	0110	0010

Исх. биты	1010	1110	1010	0010	1010	0000
Гамма	1010	0101	1110	0001	1010	1101
Результат	0000	1011	0100	0011	0000	1101

Исх. биты	1010	1101	1010	1000	1010	0101
Гамма	1000	0010	1010	0101	1110	0001
Результат	0010	1111	0000	1101	0100	0101

Закодированный текст в шестнадцатеричном виде:

01 05 4D 01 08 62 0B 43 0D 2F 0D 45

Рис. 2.6. Механізм гамірування

Методи перестановки

При використанні шифрів перестановки вхідний потік вихідного тексту ділиться на блоки, в кожному з яких виконується перестановка символів. Перестановки в класичній "докомп'ютерній" криптографії виходили в результаті запису вихідного тексту і читання шифрованого тексту за різними шляхами геометричної фігури.

Найпростішим прикладом перестановки є *перестановка з фіксованим періодом* ***d***. У цьому методі повідомлення ділиться на блоки по ***d*** символів і в кожному блоці проводиться одна і та ж перестановка. Правило, за яким здійснюється перестановка, є ключем і може бути задане деякою перестановкою перших ***d*** натуральних чисел. В результаті самі літери повідомлення не змінюються, але передаються в іншому порядку.

Наприклад, для d = 6 як ключ перестановки можна взяти 436215. Це означає, що в кожному блоці з 6 символів четвертий символ стає на перше місце, третій - на другому, шостий - на третю і т.д. Нехай необхідно зашифрувати такий текст:

ЭТО_ТЕКСТ_ДЛЯ_ШИФРОВАНИЯ ЦЕЙ_ТЕКСТ_ДЛЯ_ШИФРУВАННЯ

Кількість символів у вихідному повідомленні дорівнює 24, отже, повідомлення необхідно розбити на 4 блоки. Результатом шифрування за допомогою перестановки 436215 буде повідомлення

_ЙЕЕЦТ_ТЛСКДИШР_ЯФНАЯВУН

Теоретично, якщо блок складається з d символів, то число можливих перестановок
d! = 1 * 2 * ... * (d-1) * d.
В останньому прикладі d = 6, отже, число перестановок одно
6! = 1 * 2 * 3 * 4 * 5 * 6 = 720.

Таким чином, якщо противник перехопив зашифроване повідомлення з розглянутого прикладу, йому знадобиться не більше 720 спроб для розкриття вихідного повідомлення (за умови, що розмір блоку відомий противнику).

Для підвищення криптостійкості можна послідовно застосувати до шифрованого повідомленням дві або більше перестановки з різними періодами.

Іншим прикладом методів перестановки є *перестановка по таблиці*. У цьому методі проводиться запис вихідного тексту по рядках деякої таблиці і читання його за стовпцями цієї ж таблиці. Послідовність заповнення рядків і читання стовпців може бути будь-якою і задається ключем.

Розглянемо приклад. Нехай у таблиці кодування буде 4 стовпці і 3 рядки (розмір блоку дорівнює 3 * 4 = 12 символів). Зашифруємо такий текст:

ЦЕЙ ТЕКСТ ДЛЯ ШИФРУВАННЯ

Кількість символів у вихідному повідомленні дорівнює 24, отже, повідомлення необхідно розбити на 2 блоки. Запишемо кожен блок в свою таблицю по рядках (таблиця 2.9).

Таблиця 2.9. Шифрование методом перестановки по таблице

1 блок			
Ц	Е	Й	
Т	Е	К	С
Т		Д	Л
2 блок			
Я		Ш	И
Ф	Р	У	В
А	Н	Н	Я

Потім будемо зчитувати з таблиці кожен блок послідовно по стовпцях:

ЦТТЕЕ ЙКД СЛЯФА РНШУНИВЯ

Можна зчитувати стовпці не послідовно, а, наприклад, так: третій, другий, перший, четвертий:

ЙКДЕЕ ЦТТ СЛШУН РНЯФАИВЯ

У цьому випадку порядок зчитування стовпців і буде ключем.

У випадку, якщо розмір повідомлення не кратний розміру блоку, можна доповнити повідомлення будь-якими символами, не впливають на зміст, наприклад, прогалинами. Однак це робити не рекомендується, оскільки це дає противнику в разі перехоплення криптограми інформацію про розмір використовуваної таблиці перестановок (довжині блоку). Після визначення довжини блоку противник може знайти довжину ключа (кількість стовпців таблиці) серед дільників довжини блоку.

Подивимося, як зашифрувати і розшифрувати повідомлення, що має довжину, що не є кратною розміру таблиці перестановки. Зашифруємо слово

ДИСКОТЕКА

Кількість символів у вихідному повідомленні - 9. Запишемо повідомлення в таблицю по рядках (таблиця 2.10), а останні три клітинки залишимо порожніми.

Таблица 2.10. Шифрование неполного блока методом перестановки по таблице

Д	И	С	К
О	Т	Е	К
А			

Потім будемо зчитувати з таблиці послідовно по стовпцях:

ДОАИТСЕКК

Для розшифрування спочатку визначають число повних стовпців, тобто кількість символів в останньому рядку. Для цього ділять розмір повідомлення (*у нашому прикладі - 9*) на кількість стовпців або розмір ключа (*у прикладі - 4*). Залишок від ділення буде числом повних стовпців: ***9 mod 4 = 1***. Отже, у нашому прикладі був ***1*** *повний стовпець* і три коротких. Тепер можна поставити букви повідомлення на свої місця і розшифрувати повідомлення. Так як ключем при шифруванні було число *1234* (стовпці зчитувалися послідовно), то при розшифруванні перші три символи (ДОА) записуються в перший стовпець таблиці перестановки, наступні два (ИТ) - у другий стовпець, наступні два (СЕ) - в третій, і останні два (КК) - у четвертий. Після заповнення таблиці зчитуємо рядки і отримуємо вихідне повідомлення ДИСКОТЕКА.

Існують і інші способи перестановки, які можна реалізувати програмним і апаратним шляхом. Наприклад, при передачі даних, записаних в двійковому вигляді, зручно використовувати апаратний блок, який перемішує певним чином за допомогою відповідного електричного монтажу біти вихідного *n*-розрядного повідомлення. Так, якщо взяти розмір блоку рівний восьми бітам, можна, наприклад, використовувати такий блок перестановки, як на рис. 2.7.

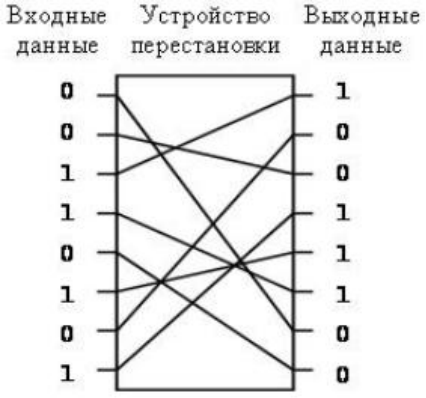


Рис. 2.7. Апаратний блок перестановки

Для розшифрування на приймальній стороні встановлюється інший блок, який відновлює порядок ланцюгів.

Апаратно реалізована перестановка широко використовується на практиці як складова частина деяких сучасних шифрів.

При перестановці будь-якого виду в зашифроване повідомлення будуть входити ті ж символи, що і у відкритий текст, але в іншому порядку. Отже, статистичні закономірності мови

залишаться без зміни. Це дає криптоаналітику можливість використовувати різні методи для відновлення правильного порядку символів.

Якщо у противника є можливість пропускати через систему шифрування методом перестановки спеціально підібрані повідомлення, то він зможе організувати атаку за обраним текстом. Так, якщо довжина блоку в початковому тексті дорівнює ***N*** символів, то для розкриття ключа достатньо пропустити через шифрувальну систему ***N-1*** блоків вихідного тексту, в яких всі символи, крім одного, однакові. Інший варіант атаки по обраному тексту можливий у випадку, якщо довжина блоку ***N*** менше кількості символів в алфавіті. У цьому випадку можна сформувати одне спеціальне повідомлення з різних букв алфавіту, розташувавши їх, наприклад, по порядку проходження в алфавіті. Пропустивши підготовлене таким чином повідомлення через шифрувальну систему, спеціалісту з криптоанализу залишиться тільки подивитися, на яких позиціях опинилися символи алфавіту після шифрування, і скласти схему перестановки.

Ключові терміни

Гамірування - метод шифрування, заснований на "накладенні" гамма-послідовності на відкритий текст. Зазвичай це підсумовування в якому-небудь кінцевому полі (підсумовування по модулю). Наприклад, у полі $GF_{(2)}$ таке підсумовування приймає вигляд звичайного "виключає АБО". При розшифровці операція проводиться повторно, в результаті виходить відкритий текст.

Пропорційні або монофонічні шифри - методи заміни, в яких зрівнюється частота появи зашифрованих знаків.

Шифри заміни (підстановки) засновані на тому, що символи вихідного тексту, зазвичай розділені на блоки і записані в одному алфавіті, замінюються одним або декількома символами іншого алфавіту відповідно до прийнятого правилом перетворення.

Шифр багатоалфавітної заміни (або підстановки) - група методів шифрування підстановкою, в яких для заміни символів вихідного тексту використовується не один, а кілька алфавітів за певним правилом.

Шифри перестановки засновані на тому, що вхідний потік вихідного тексту ділиться на блоки, в кожному з яких виконується перестановка символів. Ключем такого шифру є використовувана при шифруванні перестановочна матриця або вектор, який вказує правило перестановки.

Шифр простий (або одноалфавітної) заміни, простий підстановлювальний шифр, моноалфавітний шифр- група методів шифрування, які зводяться до створення за певним алгоритмом таблиці шифрування, в якій для кожної букви відкритого тексту існує єдина зіставлена їй буква шифротексту. Само шифрування полягає в заміні букв згідно з таблицею. Для розшифровки досить мати ту ж таблицю, або знати алгоритм, за якою вона генерується.

Симетричне шифрування (шифрування із закритим ключем) - методи оборотного перетворення даних, в яких використовується один і той же ключ, який обидві сторони інформаційного обміну повинні зберігати в секреті від противника. Всі відомі з історії шифри, наприклад, шифр Цезаря - це шифри із закритим ключем.

Короткі підсумки

Симетричні шифри - спосіб шифрування, в якому для шифрування і розшифрування застосовується один і той же криптографічний ключ. Ключ шифрування повинен зберігатися в секреті обома сторонами.

Відомі різні методи шифрування із закритим ключем. На практиці часто використовуються алгоритми перестановки, підстановки, а також комбіновані методи.

У методах перестановки символи вихідного тексту міняються місцями один з одним за певним правилом.

У методах заміни (або підстановки) символи відкритого тексту замінюються деякими еквівалентами шифрованого тексту. Шифр простий (або одноалфавітної) заміни - група методів шифрування, які зводиться до створення за певним алгоритмом таблиці шифрування, в якій для кожної букви відкритого тексту існує єдина зіставлена їй буква шифртексту. Само шифрування полягає в заміні букв згідно з таблицею. Для розшифровки досить мати ту ж таблицю, або знати алгоритм, за якою вона генерується.

Шифр багатоалфавітної заміни - група методів шифрування підстановкою, в яких для заміни символів вихідного тексту використовується не один, а кілька алфавітів за певним правилом. Таким чином, при шифруванні получаётся досить складна послідовність, яку вже не так просто розкрити, як один одноалфавітний шифр.

Окремим випадком багатоалфавітної підстановки є гамірування - метод шифрування, заснований на "накладанні" гамма-послідовності на відкритий текст. Зазвичай це підсумовування в якому-небудь кінцевому полі (підсумовування по модулю довжини алфавіту).

Найважливішим ефектом, що досягається при використанні багатоалфавітного шифру, є маскування частоти появи тих чи інших букв в тексті, на підставі якої зазвичай дуже легко розкриваються одноалфавітні шифри.

Лекція 3.
ПРИНЦИПИ ПОБУДОВИ БЛОКОВИХ ШИФРІВ ІЗ ЗАКРИТИМ КЛЮЧЕМ

Анотація: В цій лекції розглядаються принципи побудови сучасних блокових алгоритмів: операції, використовувані в блокових алгоритмах симетричного шифрування; структура блочного алгоритму; вимоги до блокового алгоритму шифрування.

Поняття композиційного шифру

Комбінація кількох поспіль застосованих простих шифрів, (наприклад, перестановки або підстановки) дає в результаті більш складне перетворення, зване комбінованим (композиційним) шифром. Цей шифр має сильніші криптографічні можливості, ніж окрема перестановка або підстановка.

Повернемося до прикладу з "Найпростіші методи шифрування із закритим ключем", в якому проводиться шифрування методом перестановки з фіксованим періодом. Нехай період перестановки d = 6, а ключ К дорівнює 436215. Це означає, що в кожному блоці з шести символів четвертий символ стає на перше місце, третій - на другому, шостий - на третьому і т.д. Зашифруємо за допомогою обраного ключа слово СИГНАЛ:

$$СИГНАЛ \xrightarrow{K=436215} НГЛИСА$$

Будемо припускати, що противнику відомий метод шифрування, але невідомий ключ. Якщо противник перехопить повідомлення НГЛИСА, йому знадобиться, як вказувалося в "Найпростіші методи шифрування із закритим ключем", не більше 720 спроб (при використанні методу повного перебору). Для того щоб вивчити 720 варіантів насправді потрібно не так вже й багато часу. Припустимо, що на вивчення кожного варіанту у супротивника йде 1 секунда. Тоді на все 720 спроб потрібно всього 12 хвилин. Таким чином, не більше ніж за 12 хвилин роботи противник дізнається наш ключ і зможе надалі розшифровувати всі повідомлення, закриті тим же ключем. Якщо ж аналіз проводиться з використанням комп'ютера, для дешифрування НГЛИСА і пошуку ключа потрібно набагато менше часу.

Яким чином можна ускладнити завдання криптоаналізу нашого шифру? Можна збільшувати розмір періоду перестановки, тобто блоку, в якому переставляються символи, наприклад, до тисячі знаків. Однак, по-перше, перебір сотень і тисяч знаків на сучасних комп'ютерах проводиться за частки хвилини, а по-друге, при цьому до тисячі символів зросте і розмір ключа. Такий ключ вже досить важко запам'ятати і використовувати. Спробуємо піти іншим шляхом і застосуємо перед перестановкою в блоці з 6 символів просту заміну за методом Цезаря, описану в "Основні поняття криптографії". Позначимо ключ в методі Цезаря k1 (1 <= k1 <= 31), а ключ при перестановці - k2. Тоді загальний ключ K = (k1, k2). Таким чином, якщо К = (5, 436215), це означає, що спочатку символи, що шифруються замінюються за методом Цезаря з ключем 5, а потім у кожному блоці з шести символів проводиться перестановка з ключем 436215. Виконаємо в два етапи шифрування слова СИГНАЛ:

$$\textbf{1 этап (замена):} \quad СИГНАЛ \xrightarrow{k1=5} ЦОИТЕР$$

$$\textbf{2 этап (перестановка):} \quad ЦОИТЕР \xrightarrow{k2=436215} ТИРОЦЕ$$

Можно записати також і так:

$$СИГНАЛ \xrightarrow{K=(5,436215)} ТИРОЦЕ$$

Кількість можливих ключів в шифрі Цезаря є в нашому випадку 31, тому загальне число варіантів можливих ключів (простір ключів) в застосованому комбінованому шифрі є 31x720 = 22320. Таким чином, дійсно, отриманий комбінований шифр значно сильніший окремо виконаних заміни і перестановки.

Для утруднення криптоаналізу статистичними методами можна використовувати наш комбінований шифр двічі з одним і тим же ключем:

Цикл шифрования **1**

1 *этап* (*замена*): СИГНАЛ $\xrightarrow{k1=5}$ ЦОИТЕР

2 *этап* (*перестановка*): ЦОИТЕР $\xrightarrow{k2=436215}$ ТИРОЦЕ

Цикл шифрования **2**

1 *этап* (*замена*): ТИРОЦЕ $\xrightarrow{k1=5}$ ЧОХУЫЛ

2 *этап* (*перестановка*): ЧОХУЫЛ $\xrightarrow{k2=436215}$ УХЛОЧЫ

В результаті двох поспіль виконаних циклів шифрування слово СИГНАЛ перетворилося на УХЛОЧИ. При цьому простір ключів шифру не змінився, однак за рахунок двократного шифрування статистичні закономірності початкового тексту замаскувалися сильніше.

У реальних шифрах також використовується комбінація декількох простих операцій над ланцюжками або блоками знаків. Для підвищення криптостійкості ці операції виконуються циклічно кілька разів, утворюючи раунди або кроки. ***На стійкість шифру впливають такі фактори, як розмір блоку, розмір ключа, кількість раундів шифрування***. Сучасні шифри із закритим ключем обробляють тільки двійкові дані, тому в них крім звичайних заміни і перестановки застосовуються деякі інші специфічні для двійкових чисел операції.

Алгоритми симетричного шифрування можуть обробляти вихідний текст блоками або потоком. Залежно від цього розрізняють блокові алгоритми симетричного шифрування і потокові. Блок тексту розглядається як невід'ємне ціле число або як кілька незалежних невід'ємних цілих чисел. Довжина блоку завжди вибирається рівною мірою двійки, наприклад, 64, 128, 256 біт.

Операції, що використовуються в блокових алгоритмах симетричного шифрування

Розглянемо *операції*, використовувані в більшості *алгоритмів симетричного шифрування*. Будемо при цьому пам'ятати, що розглянуті операції застосовуються до двійкових даних. Будь-яка інформація, наприклад, зображення або текст, можуть бути представлені в двійковому вигляді. Завдяки цьому при шифруванні не доводиться замислюватися про сенс переданих повідомлень.

Одна з часто використовуваних операцій - *операція побітового додавання за модулем 2*, що позначається XOR або $\oplus$. Принципи виконання цієї операції детально розглянуті в "Найпростіші методи шифрування із закритим ключем". При додаванні за модулем 2 операнди обробляються порозрядно. У розряді результату ставиться одиниця, якщо у відповідних розрядах операндів присутній непарне число одиниць. Наприклад, складемо по модулю 2 два 16-розрядних числа:

```
Номер разряда:    15 14 13 12 11 10 9  8  7  6  5  4   3  2  1  0
Операнд 1:         1  0  1  0  1  0  1  0  1  0  1  0   1  0  1  0
Операнд 2:         0  1  1  1  0  0  1  1  0  0  1  1   0  0  1  1
Сумма по мод. 2:   1  1  0  1  1  0  0  1  1  0  0  1   1  0  0  1
```

Ця операція має дуже зручну властивість: віднімання за модулем два є те ж саме, що і додавання, тому один з операндів може бути отриманий шляхом додавання до суми іншого операнда.

Також в блокових алгоритмах шифрування широко використовується *операція складання по модулю 2^{32} або по модулю 2^{16}*. Ця операція являє собою звичайне додавання двійкових чисел без урахування перенесення в старший 32-й або 16-й розряд результату. Наприклад, складемо по модулю 2^{16} два 16-розрядних числа:

```
Номер разряда:      15 14 13 12 11 10 9  8   7  6  5  4   3  2  1  0
Операнд 1:           1  0  1  0  1  0  1  0   1  0  1  0   1  0  1  0
Операнд 2:           0  1  1  1  0  0  1  1   0  0  1  1   0  0  1  1
Сумма по мод.2^16: (1)0  0  0  1  1  1  0   1  1  1  0   1  1  1  0  1
```

Перенесення з 15-го розряду, позначений у прикладі як одиниця в дужках, далі не використовується і тому відкидається.

Циклічний зсув пересуває ланцюжок біт на деяке число розрядів вліво або вправо. Двійкове число при виконанні операції зсуву нагадує довгу гусеницю, виповзають з одного боку тунелю і заповзає з іншого. При циклічному зсуві вліво біти, що виходять зліва за розрядну сітку дописуються праворуч на звільнені місця. При циклічному зсуві вправо всі біти пересуваються ланцюжком вправо, а ті, яким не вистачає місця, переносяться в хвіст ланцюжка. Наприклад, виконаємо циклічний зсув двійкового числа вліво на 3 розряди. Для цього будемо 3 рази переписувати двійкові цифри, щоразу зміщуючи їх вліво на 1 розряд і переносячи знаки, що виходять з п'ятнадцятого розряду на місце нульового.

Циклический сдвиг влево на 3 разряда (←)

```
Номер разряда:     15 14 13 12 11 10 9  8  7  6  5  4  3  2  1  0
Исходное число:  ← [1] 1  1  1  1  1  1  1  0  1  0  1  0  1  0
                    ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓  ↓
Сдвиг на 1 разряд: 1  1  1  1  1  1  1  0  1  0  1  0  1  0 [1] ←
Сдвиг на 1 разряд: 1  1  1  1  1  1  1  0  1  0  1  0  1  0  1  1
Сдвиг на 1 разряд: 1  1  1  1  1  1  0  1  0  1  0  1  0  1  1  1
```

Аналогічно виконується і циклічний зсув вправо. Наприклад, при зсуві вправо на 3 розряди нульовий, перший і другий біти вихідного числа виходять з розрядної сітки і запам'ятовуються, всі інші біти переміщаються вправо на 3 позиції, потім запам’ятовані цифри записуються на тринадцяте, чотирнадцятого і п'ятнадцятого місця.

Циклический сдвиг вправо на 3 разряда (→)

```
Номер разряда:     15 14 13 12 11 10 9  8  7  6  5  4  3  2  1  0
Исходное число:    1  1  1  1  1  1  1  1  1  0  1  0  1 [0  1  0] →
                                                          ↘
Сдвиг на 3 разряда: → [0  1  0] 1  1  1  1  1  1  1  1  1  0  1  0  1
```

При виконанні *табличної підстановки* група бітів відображається в іншу групу бітів. При цій операції один блок двійкових даних замінюється за певним правилом або таблиці іншим блоком. Наприклад, можна замінювати кожну групу з трьох довічних цифр іншою групою з трьох цифр по наступній таблиці:

Вхід	Вихід
000	011
001	101
010	000
011	111
100	010
101	110
110	001
111	100

Якщо кожне значення, записане в шпальтах "Вхід" і "Вихід" записати не в двійковому, а в десятковому вигляді, то ту ж саму таблицю замін можна буде записати більш коротко, наприклад, так:

```
0->3, 1->5, 2->0, 3->7, 4->2, 5->6, 6->1, 7->4
```

Перша цифра в такого запису представляє значення на вході, а друга - на виході. Якщо значення входів впорядковані за зростанням у звичайному порядку, то можна взагалі не писати першу цифру, а записати тільки відповідні значення виходів:

```
3, 5, 0, 7, 2, 6, 1, 4.
```

Тобто в якості заміни для значення 3-бітового блоку вибирається елемент з таблиці замін з порядковим номером, рівним значенню замінного блоку.

Якщо необхідно замінювати групи з чотирьох двійкових цифр, то таблиця замін повинна містити вже 16 значень. У загальному випадку для *n*-бітових блоків таблиця замін повинна містити 2^n елементів.

Табличну підстановку в літературі іноді називають заміною з використанням S-блоків або S-box. (Буква S взята від англійського слова substitution - підстановка).

Структура блокового алгоритму симетричного шифрування

Таким чином, в алгоритмах симетричного шифрування часто використовуються операції додавання за модулем 2, додавання за модулем 2^{16} або 2^{32}, циклічного зсуву, заміни та перестановки.

Ці операції циклічно повторюються в алгоритмі N раз, утворюючи так звані раунди або кроки. Вихідними даними для кожного раунду є вихід попереднього раунду і ключ, який отримано за певним алгоритмом із загального ключа шифрування K. Ключ раунду наз. підключем K_i. В результаті блоковий алгоритм шифрування може бути представлений наступним чином (рис. 3.1).

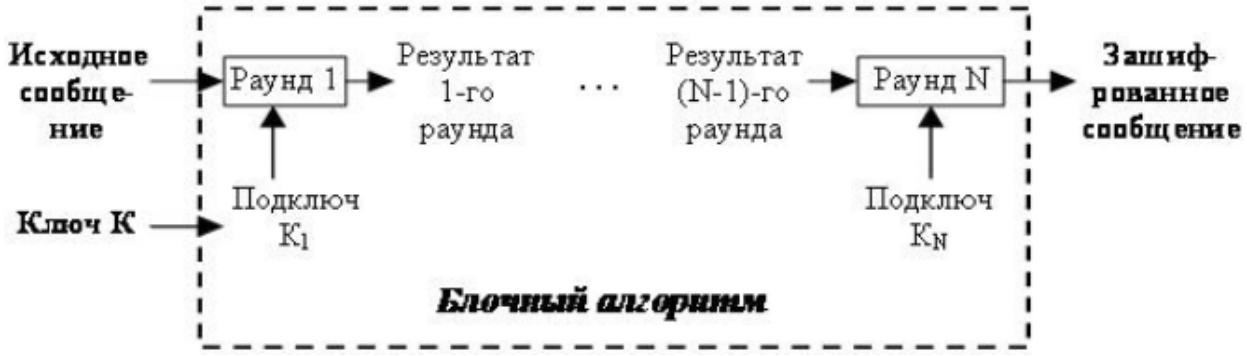


Рис. 3.1. Структура блочного алгоритму симетричного шифрування.

Блокові алгоритми шифрування застосовуються до двійкових даних. У загальному випадку процедура блочного шифрування перетворює *n*-бітний блок відкритого тексту в *k*-бітний блок зашифрованого тексту. Число блоків довжини *n* рівне 2^n. Для того щоб перетворення було оборотним, кожен з таких блоків повинен перетворюватися в свій унікальний блок зашифрованого тексту. Довжина блоку завжди вибирається рівною мірою двійки, наприклад, 64, 128, 256 біт.

Вимоги до блокового алгоритму шифрування

До сучасних алгоритмів блокового шифрування пред'являють досить жорсткі вимоги, пов'язані з областю застосування, можливістю реалізації на різних обчислювальних платформах та іншими факторами. Розглянемо основні з вимог.

1. Алгоритм повинен забезпечувати високий рівень стійкості, і ця стійкість не повинна ґрунтуватися на збереженні в секреті самого алгоритму.
2. Незначна зміна вихідного повідомлення повинна приводити до істотної зміни зашифрованого повідомлення, навіть при використанні одного і того ж ключа.
3. Алгоритм повинен успішно протистояти атакам по обраному тексту, тобто таким, щоб не можна було впізнати ключ, навіть знаючи досить багато пар (зашифроване повідомлення, незашифроване повідомлення), отриманих при шифруванні з використанням даного ключа.
4. Алгоритм шифрування повинен мати можливість бути реалізованим на різних платформах, які висувають різні вимоги. Для найбільш швидких додатків використовується спеціальна апаратура. Незважаючи на це, програмні реалізації застосовуються також досить часто. Тому алгоритм повинен допускати ефективну програмну реалізацію на універсальних мікропроцесорах. Алгоритм повинен також працювати на мікроконтролерах та інших процесорах середнього розміру.
5. Алгоритм повинен використовувати прості операції, які ефективні на мікропроцесорах, тобто виключаюче АБО, додавання, табличні підстановки, множення по модулю. Не повинно використовуватися зрушень змінної довжини, побітових перестановок або умовних переходів.
6. Алгоритм повинен ефективно реалізовуватися на спеціалізованій апаратурі, призначеній для виконання операцій шифрування і розшифрування, тобто реалізація алгоритму у вигляді електронних пристроїв повинна бути економічною.
7. Алгоритм шифрування повинен бути застосовний в багатьох додатках. Алгоритм повинен бути ефективний при шифруванні файлів даних або великого потоку даних, при створенні певної кількості випадкових бітів, а також повинна бути можливість його використання для формування односторонньої хеш-функції.
8. Алгоритм повинен бути простим для написання коду, щоб мінімізувати ймовірність програмних помилок. Також це дає можливість аналізу і зменшує закритість алгоритму.

9. Алгоритм повинен допускати будь-який випадковий рядок бітів потрібної довжини в якості можливого ключа (*це називається мати плоский простір ключів*). Не повинно бути "слабких" ключів, що полегшують криптоаналіз.
10. Алгоритм повинен легко модифікуватися для різних рівнів безпеки і задовольняти як мінімальним, так і максимальним вимогам.

Деяке уточнення необхідно зробити щодо пункту 1, що вимагає високу криптостойкость алгоритму шифрування. Зазвичай під "*високою криптостійкістю*" розуміють, що шифр повинен бути стійким по відношенню до атаки по обраному тексту. Це автоматично передбачає його стійкість по відношенню до атак по шифротексту і по відомому тексту. Однак відомо, що при атаці по обраному тексту шифр завжди може бути зламаний шляхом перебору ключів. Тому вимога стійкості шифру можна уточнити таким чином: "*Шифр стійкий (при атаці по обраному тексту), якщо для нього не існує алгоритму злому, істотно більш швидкого, ніж прямий перебір ключів*". Цікаво, що станом на сьогоднішній день ні для одного використовуваного шифру не доведена сувора відповідність з цим визначенням стійкості.

Короткі підсумки

У симетричних блокових шифрах використовується комбінація декількох простих операцій над ланцюжками або блоками біт: додавання за модулем *2*, додавання за модулем 2^{16} або 2^{32}, циклічного зсуву, заміни та перестановки. Для підвищення криптостійкості ці операції виконуються циклічно кілька разів, утворюючи раунди або кроки. На стійкість шифру впливають такі фактори, як розмір блоку, розмір ключа, кількість раундів шифрування.

Алгоритми симетричного шифрування розрізняються способом, яким обробляється вихідний текст. Можливо шифрування блоками або шифрування потоком. Блок тексту розглядається як невід'ємне ціле число або як кілька незалежних невід'ємних цілих чисел. Довжина блоку завжди вибирається рівною мірою двійки, наприклад, 64, 128, 256 біт.

Блоковий алгоритм симетричного шифрування може мати в своїй основі мережу (схему) Фейштеля. У цьому випадку вхідний блок ділиться на кілька частин рівної довжини (гілки). Гілки обробляються окремо, після чого здійснюється циклічний зсув всіх гілок вліво.

Питання для самоперевірки

1. Який шифр називають комбінованим або композиційним шифром?
2. Які фактори впливають на стійкість блокового алгоритму шифрування?
3. Які найпростіші операції застосовуються в блокових алгоритмах шифрування?
4. У чому відмінність блокових алгоритмів шифрування від поточних?
5. Що розуміється під "раундом" алгоритму шифрування?
6. Які вимоги до блокового алгоритму шифрування?
7. Чому блоковий алгоритм шифрування повинен мати просту і зрозумілу структуру?
8. Що розуміється під вимогою "високої криптостійкості" алгоритму шифрування?

Вправи для самоперевірки

1. Складіть по модулю 2:
- двійкові числа 10101100 та 11001010;
- десяткові числа 15 і 10;
- шістнадцяткові числа 0B5 і 37.

Примітка: десяткові і шістнадцяткові числа необхідно спочатку перевести в двійковий вигляд.

2. Складіть по модулю 2^8:
- двійкові числа 10101100 та 11001010;

- десяткові числа 155 і 100;
- шістнадцяткові числа 0B5 і 37.
Примітка: десяткові числа необхідно спочатку перевести в двійковий вигляд.

3. Виконайте операцію циклічного зсуву:
- вліво на 5 розрядів для двійкового числа 10101100;
- вправо на 4 розряди для шістнадцяткового числа 9E;
- вправо на 2 розряди для шістнадцяткового числа 55.
Примітка: шістнадцяткові числа необхідно спочатку перевести в двійковий вигляд.

4. Нехай кожні три біти вхідного повідомлення замінюються по наступній таблиці замін:

Вхід	Вихід
000	011
001	101
010	000
011	111
100	010
101	110
110	001
111	100

Виконайте розбиття вихідного повідомлення на блоки по три біти і зробіть поблочну заміну для наступних повідомлень, поданих в цифровому вигляді:
- 1010 1100 1100 (2)
- 2356 (10)
- 0B57 (16)
Примітка: десяткові і шістнадцяткові числа необхідно спочатку перевести в двійковий вигляд.

Лекція 4
КРИПТОГРАФІЧНІ ХЕШ-ФУНКЦІЇ

Анотація: В цій лекції сформульовано поняття хеш-функції, а також наведено короткий огляд алгоритмів формування хеш-функцій. Крім того, розглянуто можливість використання блокових алгоритмів шифрування для формування хеш-функції.

Ключові слова: функція, значення, біт, хешування, ймовірність, операції, байт, довжина, вихід, алгоритм симетричного шифрування, алгоритм, шифрування, CBC, имитовставка, цілісність, ключ, довжина блоку, MD5, SHA-1, SHA, MD4, MD, message digest, автор, вхідні дані, інверсія, кон'юнкція, додавання, secure, algorithm, NIST, довільне, SHA-256, SHA-512, вектор

Поняття хеш-функції

Хеш-функцією (hash function) називається математична чи інша функція, яка для рядка довільної довжини обчислює деяке ціле значення або деякий інший рядок фіксованої довжини. Математично це можна записати так:

h=H(M),

де *M* - вихідне повідомлення (праобраз), а *h* - результат, який називається значенням хеш-функції (а також хеш-кодом або дайджестом повідомлення (від англ. message digest)).

Зміст хеш-функції полягає у визначенні характерної ознаки праобразу - значення хеш-функції. Це значення зазвичай має певний фіксований розмір, наприклад, 64 або 128 біт. Хеш-код може бути надалі проаналізований для вирішення якої-небудь задачі. Так, наприклад, хешування може застосовуватися для порівняння даних: якщо у двох масивів даних хеш-коди різні, масиви гарантовано відрізняються; якщо однакові - масиви, швидше за все, однакові. У загальному випадку однозначної відповідності між вихідними даними і хеш-кодом немає через те, що кількість значень хеш-функцій завжди менше, ніж варіантів вхідних даних. Отже, існує безліч вхідних повідомлень, що дають однакові хеш-коди (такі ситуації називаються колізіями). Імовірність виникнення колізій відіграє важливу роль в оцінці якості хеш-функцій.

Хеш-функції широко застосовуються в сучасній криптографії.

Найпростіша хеш-функція може бути складена з використанням операції "сума по модулю 2" таким чином: отримуємо вхідний рядок, складаємо всі байти по модулю 2 і байт-результат повертаємо в якості значення хеш-фукнції. Довжина значення хеш-функції складе в цьому випадку 8 біт незалежно від розміру вхідного повідомлення.

Наприклад, нехай вихідне повідомлення, перекладене в цифровий вигляд, було наступним (у шістнадцятковому форматі):

3E 54 A0 1F B4

Переведемо повідомлення в двійковий вигляд, запишемо байти один під одним і складемо біти в кожному стовпчику по модулю 2:

0011 1110
0101 0100
1010 0000
0001 1111
1011 0100

0110 0101

Результат ($0110\ 0101_{(2)}$ или $65_{(16)}$) и будет значением хеш-функции.

Однак таку хеш-функцію не можна використовувати для криптографічних цілей, наприклад для формування електронного підпису, так як досить легко змінити зміст підписаного повідомлення, не змінюючи значення контрольної суми.

Тому розглянута хеш-функція не годиться для криптографічних застосувань. У криптографії хеш-функція вважається хорошою, якщо важко створити два праобрази з однаковим значенням хеш-функції, а також, якщо вихід функції немає явної залежності від входу.

Сформулюємо основні вимоги, що пред'являються до криптографічних хеш-функцій:

- хеш-функція повинна бути застосовна до повідомлення будь-якого розміру;
- обчислення значення функції повинно виконуватися досить швидко;
- при відомому значенні хеш-функції повинно бути важко (практично неможливо) знайти відповідний праобраз *M*;
- при відомому повідомленні *M* повинно бути важко знайти інше повідомлення *M'* з таким же значенням хеш-функції, як у вихідного повідомлення;
- повинно бути важко знайти яку-небудь пару випадкових різних повідомлень з однаковим значенням хеш-функції.

Створити хеш-функцію, яка задовольняє всім перерахованим вимогам - завдання непросте. Необхідно також пам'ятати, що на вхід функції надходять дані довільного розміру, а хеш-результат не повинен виходити однаковим для даних різного розміру.

В даний час на практиці в якості хеш-функцій застосовуються функції, що обробляють вхідне повідомлення блок за блоком і обчислюють хеш-значення h_i для кожного блоку M_i вхідного повідомлення:

$$h_i=H(M_i,h_{i-1}),$$

де h_{i-1} - результат, отриманий при обчисленні хеш-функції для попереднього блоку вхідних даних.

В результаті вихід хеш-функції h_n є функцією від усіх n блоків вхідного повідомлення

Використання блокових алгоритмів шифрування для формування хеш-функції

В якості хеш-функції можна використовувати *блоковий алгоритм симетричного шифрування*. Якщо використовуваний блоковий алгоритм криптографічно стійкий, то і хеш-функція на його основі буде надійною.

Найпростішим способом використання блокового алгоритму для отримання хеш-коду є шифрування повідомлення в режимі зчеплення блоків по шифротексту. У цьому випадку повідомлення представляється у вигляді послідовності блоків, довжина яких дорівнює довжині щоб вийшов блок потрібної довжини. Хеш-значенням буде останній зашифрований блок тексту. За умови використання надійного блокового алгоритму шифрування, отримане хеш-значення буде характеризуватися такими властивостями:

- практично неможливо без знання ключа шифрування обчислення хеш-значення для заданого відкритого масиву інформації;
- практично неможливий без знання ключа шифрування підбір відкритих даних під задане значення хеш-функції.

Сформоване таким чином хеш-значення зазвичай називають *імітовставкою або аутентифікатором* і використовується для перевірки цілісності повідомлення. Таким чином, ***імітовставка*** *- це контрольна комбінація, що залежить від відкритих даних і секретної ключової інформації*. Метою використання імітовставки є виявлення всіх випадкових або навмисних змін в масиві інформації. Значення, отримане хеш-функцією при обробці вхідного

повідомлення, приєднується до повідомлення в той момент, коли відомо, що повідомлення правильне. Одержувач перевіряє цілісність повідомлення шляхом обчислення імітовставки отриманого повідомлення і порівняння його з отриманим хеш-кодом, який повинен бути переданий безпечним способом. Одним з таких безпечних способів може бути шифрування імітовставки закритим ключем відправника, тобто створення підпису. Можливо також шифрування отриманого хеш-коду алгоритмом симетричного шифрування, якщо відправник і одержувач мають загальний ключ симетричного шифрування.

Зазначений процес отримання та використання імітовставки описаний у вітчизняному стандарті ГОСТ 28147-89. Стандарт пропонує використовувати молодші 32 біти блоку, отриманого на виході операції шифрування всього повідомлення в режимі зчеплення блоків шифру для контролю цілісності переданого повідомлення. Таким же чином для формування імітовставки можна використовувати будь-який блоковий алгоритм симетричного шифрування.

Іншим можливим способом застосування блокового шифру для вироблення хеш-коду є наступний. Початкове повідомлення обробляється послідовно блоками. Останній блок при необхідності доповнюється нулями, іноді в останній блок приписують довжину повідомлення у вигляді двійкового числа. На кожному етапі шифруємо хеш-значення, отримане на попередньому етапі, взявши в якості ключа поточний блок повідомлення. Останнє отримане зашифроване значення буде остаточним хеш-результатом.

Таким чином, якщо звичайну схему шифрування повідомлення M за допомогою блочного шифру f на ключі K ми записували як $E = f(M, K)$, то схему отримання хеш-коду h за описаним вище алгоритмом можна представити як

$$h_i = f(h_{i-1}, M)$$

В якості початкового хеш-коду h_0 беруть деяку константу. Шифрування проводиться в режимі простої заміни. При використанні зазначеного способу розмір блоку збігається з довжиною ключа і *розміром* хеш-значення буде *довжина блоку*.

Можливий також **інший спосіб** використання блокового шифру в режимі простої заміни: елементи повідомлення шифруються хеш-значеннями, отриманими на попередньому етапі:

$$h_i = f(M, h_{i-1})$$

Насправді можливі ще кілька схем використання блочного шифру для формування хеш-функції. Нехай M_i - блок вихідного повідомлення, h_i - значення хеш-функції на i-тому етапі, f - блоковий алгоритм шифрування, використовуваний в режимі простої заміни, $\oplus$ - операція додавання по модулю 2. Тоді можливі, наприклад, такі схеми формування хеш- функції:

$$h_i = f(M_i, h_{i-1}) \oplus M_i,$$
$$h_i = f(M_i, h_{i-1}) \oplus h_{i-1} \oplus M_i,$$
$$h_i = f(h_{i-1}, M_i) \oplus h_i,$$
$$h_i = f(h_{i-1} \oplus M_i, M_i) \oplus h_i$$

У всіх цих *схемах використання блокового алгоритму шифрування* довжина хеш-значення, що формується, дорівнює довжині блоку при шифруванні.

Узагальнена схема симетричного блочного алгоритму шифрування зображена на рис. 2.

Рис.2. Обобщенная схема формирования хэш-функции

A, B, C можуть приймати значення $M_i, H_{i-1}, (M_i \oplus H_{i-1})$ або бути константою, де M_i - *i*-тий блок вхідного потоку, $\oplus$ - додавання по *mod2,* H_i - результат *i*-тої ітерації.

Таким чином, ми отримуємо 64 варіанти побудови функції стиснення. Більшість з них є або тривіальними, або небезпечними. Нижче зображено чотири найбільш безпечні схеми при всіх видах атак.

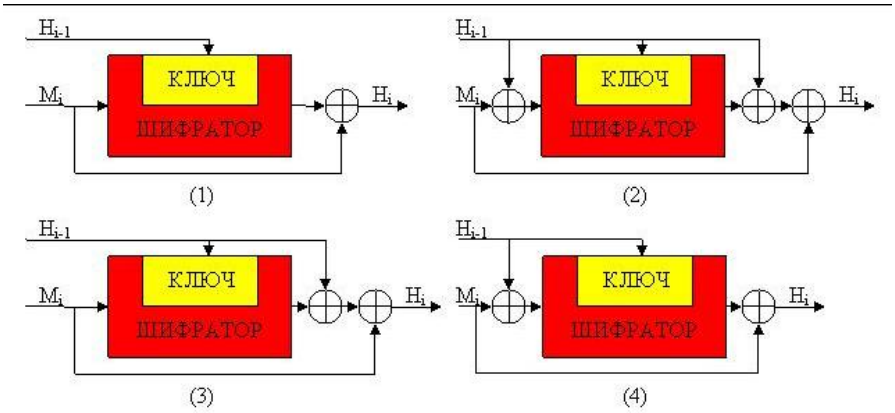


Рис.3. Четыре схемы безопасного хэширования

Основним недоліком хеш-функцій, спроектованих на основі блокових алгоритмів, є відносно низька швидкодія. Необхідну криптостійкость можна забезпечити і за меншу кількість операцій над вхідними даними. Існують більш швидкі алгоритми хешування, спроектовані самостійно, з нуля, виходячи з вимог криптостійкості.

Огляд алгоритмів формування хеш-функцій

В даний час запропоновані і практично використовуються різні спеціальні алгоритми для обчислення хеш-функції. Найбільш відомими алгоритмами є MD5, SHA-1, SHA-2 і інші версії SHA, а також алгоритм, викладений у ГОСТ Р 34.11-94.

Алгоритм MD5 з'явився на початку 90-х років XX століття в результаті удосконалення алгоритму формування хеш-функції MD4. Символи в назві "MD" означають Message Digest - короткий виклад повідомлення. Автор алгоритмів MD4 і MD5 - Р. Ривест (R.Rivest). В результаті використання MD5 для довільного повідомлення формується 128-бітове хеш-значення. Вхідні дані обробляються блоками по 512 біт. В алгоритмі використовуються елементарні логічні операції (*інверсія, кон'юнкція, додавання по модулю 2, циклічні зсуви та ін.*), а також звичайне арифметичне додавання. Комплексне повторення цих елементарних функцій алгоритму забезпечує те, що результат після обробки добре перемішаний. Тому малоймовірно, щоб два повідомлення, обрані випадково, мали однаковий хеш-код. *Алгоритм MD5 має наступну властивість: кожен біт отриманого хеш-значення є функцією від кожного біта входу*. Вважається, що MD5 є найбільш сильною хеш-функцією для 128-бітного хеш-значення.

Алгоритм SHA (Secure Hash Algorithm - Безпечний хеш-алгоритм) був розроблений національним інститутом стандартів і технології (NIST) США і опублікований в якості американського федерального інформаційного стандарту в 1993 році. SHA-1, як і MD5, заснований на алгоритмі MD4. SHA-1 формує 160-бітове хеш-значення на основі обробки вихідного повідомлення блоками по 512 біт. В алгоритмі SHA-1 також використовуються

прості логічні і арифметичні операції. Найбільш важливою відмінністю SHA-1 від MD5 є те, що хеш-код SHA-1 на 32 біта довший, ніж хеш-код MD5. Якщо припустити, що обидва алгоритми однакові за складністю для криптоаналізу, то SHA-1 є більш стійким алгоритмом. Використовуючи атаку методом грубої сили (лобову атаку), важче створити довільне повідомлення, що має даний хеш-код, а також важче створити два повідомлення, що мають однаковий хеш-код.

У 2001 році національний інститут стандартів і технології США прийняв в якості стандарту три хеш-функції з більшою довжиною хеш-коду, ніж у SHA-1. Часто ці хеш-функції називають SHA-2 або SHA-256, SHA-384 і SHA-512 (у назві вказується довжина створюваного алгоритмами хеш-коду). Ці алгоритми відрізняються не тільки довжиною створюваного хеш-коду, але і використовуваними внутрішніми функціями і довжиною оброблюваного блоку (у SHA-256 довжина блоку - 512, а у SHA-384 і SHA-512 довжина блоку - 1024 біта). Поступові удосконалення алгоритму SHA ведуть до збільшення його криптостійкості. Незважаючи на відмінності розглянутих алгоритмів один від одного, всі вони є подальшим розвитком SHA-1 і MD4 і мають схожу структуру.

У Росії прийнятий **ГОСТ Р34.11-94**, який є стандартом для хеш-функцій. Його структура досить сильно відрізняється від структури алгоритмів SHA-1,2 або MD5, в основі яких лежить алгоритм MD4. Довжина хеш-коду, створюваного алгоритмом ГОСТ Р 34.11-94, дорівнює 256 бітам. Алгоритм послідовно обробляє вихідне повідомлення блоками по 256 біт справа наліво. Параметром алгоритму є стартовий вектор хешування - довільне фіксоване значення довжиною також 256 біт. В алгоритмі ГОСТ Р 34.11-94 використовуються операції перестановки, зсуву, арифметичного додавання, додавання за модулем 2. В якості допоміжної функції в ГОСТ 34.11-94 використовується алгоритм по ГОСТ 28147-89 в режимі простої заміни.

Ключові терміни

Hash function - хеш-функція.

Хеш-функція - математична чи інша функція, яка для рядка довільної довжини обчислює деяке ціле значення або деякий інший рядок фіксованої довжини.

Хеш-код - результат роботи хеш-функції, деяка характерна "ознака" вхідного масиву даних.

Питання для самоперевірки

1. Що в криптографії називається хеш-функцією?
2. Для яких цілей використовуються хеш-функції?
3. Перелічіть основні вимоги, пропоновані до хеш-функцій.
4. Назвіть приклади криптографічних хеш-функцій.
5. Яким чином можна використовувати блоковий алгоритм шифрування для формування хеш-функції?

Вправи для самоперевірки

Нехай хеш-функція y = x (x_1,x_2, ... x_n) визначається як результат виконання побітової операції "сума по модулю 2" для всіх байтів повідомлення, представленого в двійковому вигляді. Довжина хеш-коду рівна 8 біт. Для кожного з шести повідомлень, записаних в лівому стовпчику, знайдіть відповідний результат обчислення хеш-функції з правого стовпчика. Всі повідомлення і значення хеш-функції представлені в шістнадцятковому форматі.

Повідомлення	Значення хеш-функції
• 0A3 69 2C • 82 0F B5 • 0DA 14 90 • 32 01 BF • 9E A6 23 • 10 BE 57	• 38 • 1B • 0F9 • 8C • 0E6 • 5E

Лекція 5
ВВЕДЕННЯ В КРИПТОГРАФІЮ З ВІДКРИТИМ КЛЮЧЕМ

Анотація: В цій лекції читач познайомиться з найбільш важливим досягненням криптографів XX століття - асиметричною криптографією і дізнається, які математичні функції називаються односторонніми і як вони використовуються для шифрування, формування секретних ключів та цифрового підпису на електронних документах.

Передумови створення методів шифрування з відкритим ключем і основні визначення

При використанні шифрування із закритим ключем виникають дві досить серйозні проблеми. Перша проблема полягає у виготовленні секретних ключів і доставці їх учасникам інформаційного обміну. При великій кількості і територіальної розподіленості учасників інформаційного обміну, що використовують канали зв'язку загального призначення, наприклад, звичайну або електронну пошту, часто буває складно гарантувати безпеку доставки такого ключа і його справжність.

Другою проблемою є забезпечення автентичності партнерів при електронному спілкуванні. Розвиток ділового листування та електронної комерції вимагає наявності методів, при використанні яких неможливо було б підмінити кого-небудь з учасників обміну. Одержувач кореспонденції повинен мати можливість упевнитися в достовірності документа, а творець електронного послання повинен бути в змозі довести своє авторство одержувачу або третій стороні. Отже, електронні документи повинні мати аналог звичайного підпису.

Багато криптографів працювали над вирішенням цих проблем, в результаті чого в другій половині сімдесятих років XX століття були розроблені принципово нові підходи, що дозволяють вирішити перераховані вище (і деякі інші) завдання. Основою послужило відкриття так званих асиметричних криптоалгоритмів, або методів, в яких процедури прямого і зворотного криптоперетворень виконуються на різних ключах і не мають між собою очевидних і легко простежуються зв'язків, які дозволили б по одному ключу визначити інший. Асиметричні алгоритми набагато більше засновані на властивостях математичних функцій, ніж алгоритми симетричного шифрування, які використовують в основному тільки операції перестановки і заміни. Великий внесок у ці дослідження внесли американські вчені У. Діффі (W. Diffie), Е. Хеллман (M. Hellman), Р. Меркль (R. Merkle). Вони першими запропонували способи вирішення обох завдань, які радикально відрізняються від усіх попередніх підходів до шифрування.

Асиметричні алгоритми шифрування називаються також алгоритмами з відкритим ключем. На відміну від алгоритмів симетричного шифрування (алгоритмів шифрування із закритим ключем), в яких для шифрування і розшифрування використовується один і той же ключ, в асиметричних алгоритмах один ключ використовується для шифрування, а інший, відмінний від першого, - для розшифрування. Алгоритми називаються асиметричними, так як ключі шифрування і розшифрування різні, отже, відсутня симетрія основних криптографічних процесів. Один з двох ключів є відкритим (public key) і може бути оголошений всім, а другий - закритим (private key) і повинен триматися в секреті. Який з ключів, відкритий або закритий, використовується для шифрування, а який для розшифрування, визначається призначенням криптографічного системи.

В даний час асиметричні алгоритми широко застосовуються на практиці, наприклад, для забезпечення інформаційної безпеки телекомунікаційних мереж, в тому числі мереж, які мають

складну топологію; для забезпечення інформаційної безпеки в глобальній мережі Internet; в різних банківських і платіжних системах (в тому числі використовують інтелектуальні картки).

Алгоритми шифрування з відкритим ключем можна використовувати для вирішення, як мінімум, трьох завдань:

1. Для шифрування переданих і збережених даних для їх захисту від несанкціонованого доступу.
2. Для формування цифрового підпису під електронними документами.
3. Для розподілу секретних ключів, які використовуються потім при шифруванні документів симетричними методами.

Односторонні функції

Всі алгоритми шифрування з відкритим ключем засновані на використанні так званих односторонніх функцій. **Односторонньою функцією** (*one-way function*) називається математична функція, яку відносно легко вирахувати, але важко знайти за значенням функції відповідне значення аргументу. Тобто, знаючи x легко обчислити $f(x)$, але за відомим $f(x)$ важко знайти підходяще значення x. Під словом "важко вирахувати" розуміють, що для цього буде потрібно не один рік розрахунків з використанням ЕОМ. Односторонні функції застосовуються в криптографії також як хеш-функції. Використовувати односторонні функції для шифрування повідомлень з метою їх захисту не має сенсу, так як назад розшифрувати зашифроване повідомлення вже не вийде. Для цілей шифрування використовуються спеціальні односторонні функції - **односторонні функції з люком** (або з секретом) - це особливий вид односторонніх функцій, що мають деякий секрет (люк), що дозволяє відносно швидко обчислити зворотне значення функції.

Для односторонньої функції з люком f справедливі наступні твердження:
- знаючи x, легко обчислити $f(x)$,
- за відомим значенням $f(x)$ важко знайти x,
- знаючи додатково деяку секретну інформацію, можна легко обчислити x.

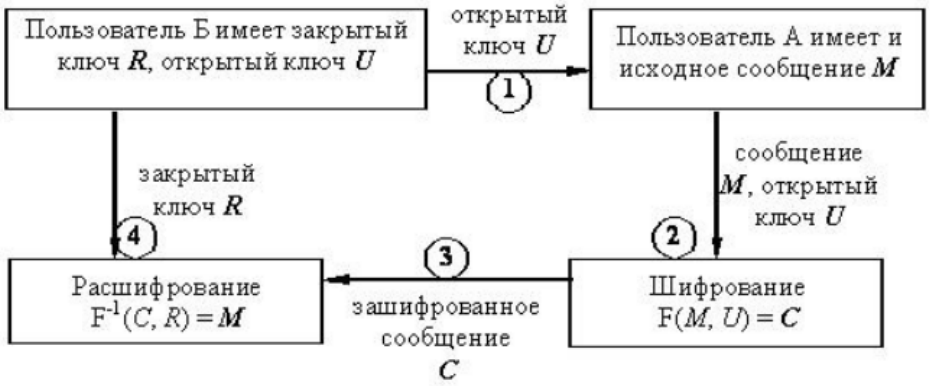
Використання асиметричних алгоритмів для шифрування

У 70-х роках XX століття Діффі і Хеллман запропонували принцип шифрування, заснований на використанні двох різних ключів, хоча і пов'язаних між собою, але сформованих так, що обчислити по одному з них (відкритому) інший (закритий) практично неможливо. Цей принцип може бути використаний для вирішення проблеми повідомлення ключів шифрування / розшифрування користувачам, а точніше - для усунення цієї проблеми. Згідно Діффі і Хеллману попередньо розподілені закриті ключі взагалі не повинні використовуватися для шифрування даних (бо секрет, який відомий більш ніж одній людині, - вже не секрет). Закритий ключ повинен бути відомий лише одній особі - його власнику. Такий принцип використання асиметричних алгоритмів отримав назву *відкритого шифрування або шифруванням з відкритим ключем*.

Згідно з цим принципом, будь-який бажаючий може зашифрувати повідомлення відкритим ключем. Розшифрувати повідомлення зможе тільки власник закритого ключа. Нехай, наприклад, користувачі **А** і **Б**, що мають можливість обмінюватися електронними повідомленнями, використовують схему відкритого шифрування. Припустимо, користувач **А** повинен передати секретне повідомлення користувачу **Б** так, щоб ніхто інший не зміг його прочитати. Для цього необхідно виконати наступні дії:

1. Користувач **Б** посилає користувачеві **А** свій відкритий ключ ***U*** по будь-якому каналу зв'язку, наприклад, по електронній пошті.
2. Користувач **А** шифрує своє повідомлення ***M*** отриманим відкритим ключем ***U*** і отримує зашифроване повідомлення ***C***.
3. Зашифроване повідомлення ***C*** пересилається користувачеві **Б**.
4. Користувач **Б** розшифровує отримане повідомлення ***C*** своїм закритим ключем ***R***.

Если операцию шифрования обозначить как F, а операцию расшифрования как F^{-1}, то схему протокола обмена информацией между пользователями можно изобразить, как на рис. 9.1.



Використання відкритого шифрування знімає проблему розподілу ключів. Раніше користувачі перед обміном зашифрованими даними повинні були якимось чином по закритому каналу зв'язку погоджувати використовуваний секретний ключ. Для цього вони могли зустрітися особисто або використовувати кур'єра. Якщо один з користувачів вважав потрібний змінити ключ, він повинен був передати на новий ключ своєму абоненту. Криптографія з відкритими ключами все спрощує. Користувачі системи зв'язку можуть абсолютно вільно обмінюватися відкритими ключами і зашифрованими ними повідомленнями. Якщо користувач надійно зберігає свій закритий ключ, ніхто не зможе прочитати передані повідомлення.

Для спрощення процедури обміну в мережі передачі повідомлень зазвичай використовується база даних, в якій зберігаються відкриті ключі всіх користувачів. При необхідності будь-який користувач системи може запросити з бази відкритий ключ іншої людини і використовувати отриманий ключ для шифрування повідомлень.

Цифровий підпис на основі алгоритмів з відкритим ключем

Як і всі люди, абоненти мережі передачі даних можуть не довіряти один одному або вести себе нечесно. Вони можуть підробляти чужі повідомлення, заперечувати своє авторство або видавати себе за іншу особу. Особливо актуальними стають ці проблеми у зв'язку з розвитком електронної комерції та можливістю оплати послуг через Інтернет. Тому в багатьох системах зв'язку одержувач кореспонденції повинен мати можливість упевнитися в достовірності документа, а творець електронного послання повинен бути в змозі довести своє авторство одержувачу або третій стороні.

Отже, електронні документи повинні мати аналог звичайної фізичної підпису. При цьому підпис повинна володіти такими властивостями:

- підпис відтворюється тільки однією особою, а справжність його може бути засвідчена багатьма;
- підпис пов'язують тільки з одним повідомленням і він не може бути перенесений на інший документ;
- після того, як документ підписаний, його неможливо змінити;

- від поставленого підпису неможливо відмовитися, тобто особа, що підписала документ, не зможе потім стверджувати, що не ставила підпис.

Асиметричні алгоритми шифрування можуть бути використані для формування **цифрового (електронного) підпису** (digital signature) - унікального числового доповнення до переданої інформації, що дозволяє перевірити її авторство. Електронний (цифровий) підпис (ЕЦП) являє собою послідовність біт фіксованої довжини, яка обчислюється певним чином за допомогою змісту підписаної інформації та секретного ключа.

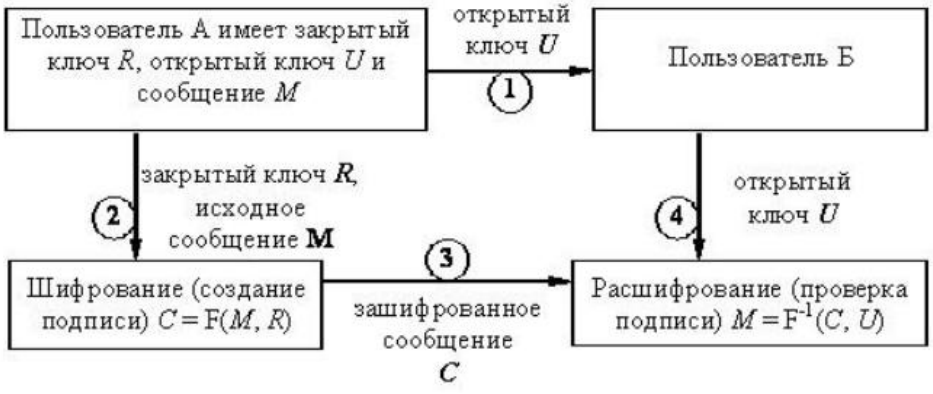
При формуванні цифрового підпису спеціальним чином шифрується або все повідомлення цілком, або результат обчислення хеш-функції від повідомлення. Останній спосіб використовується частіше, оскільки повідомлення, що підписується, може мати різний розмір, іноді досить великий, а хеш-код завжди має постійну не дуже велику довжину.

Розглянемо докладніше обидва варіанти формування ЕЦП.

Найпростіший спосіб ґрунтується, так само як і при відкритому шифруванні, на використанні пари зв'язаних між собою ключів (відкритого і закритого). Однак ролі закритого і відкритого ключів змінюються - *ключ **підписання** стає **секретним**, а ключ **перевірки** - **відкритим***. Якщо при цьому зберігається властивість, що по відкритому ключу не можна практично знайти закритий ключ, то як підпис може виступати саме повідомлення, зашифроване секретним ключем. Таким чином підписати повідомлення може тільки власник закритого ключа, але кожен, хто має його відкритий ключ, може перевірити підпис.

Нехай, наприклад, користувач **А** хоче відправити користувачеві **Б** підписане повідомлення. Процедура створення та перевірки підпису складається з наступних кроків:

1. Користувач **А** посилає користувачеві **Б** свій відкритий ключ ***U*** по будь-якому каналу зв'язку, наприклад, по електронній пошті.
2. Користувач **А** шифрує повідомлення ***M*** своїм закритим ключем ***R*** і отримує зашифроване повідомлення ***C***.
3. Зашифроване повідомлення пересилається користувачеві **Б**.
4. Користувач **Б** розшифровує отримане повідомлення ***C***, використовуючи відкритий ключ користувача **А**. Якщо повідомлення розшифрувалося, значить, воно підписане користувачем **А**.



До тих пір, поки користувач **А** надійно зберігає свій закритий ключ, його підписи достовірні. Крім того, неможливо змінити повідомлення, не маючи доступу до закритого ключа абонента **А**; тим самим забезпечується аутентифікація і цілісність даних.

Фізичне представлення пари ключів залежить від конкретної системи, що підтримує використання ЕЦП. Найчастіше ключ записується у файл, який, на додаток до самого ключа, може містити, наприклад, інформацію про користувача - власника ключа, про термін дії ключа,

а також якийсь набір даних, необхідних для роботи конкретної системи. Дані про власника ключа дозволяють реалізувати іншу важливу функцію ЕЦП - встановлення авторства, оскільки при перевірці підпису відразу ж стає ясно, хто підписав те або інше повідомлення. Зазвичай програмні продукти, які здійснюють перевірку ЕЦП, настроюються так, щоб результат виконання з'являвся на екрані в зручному для сприйняття вигляді із зазначенням користувача, що поставив підпис.

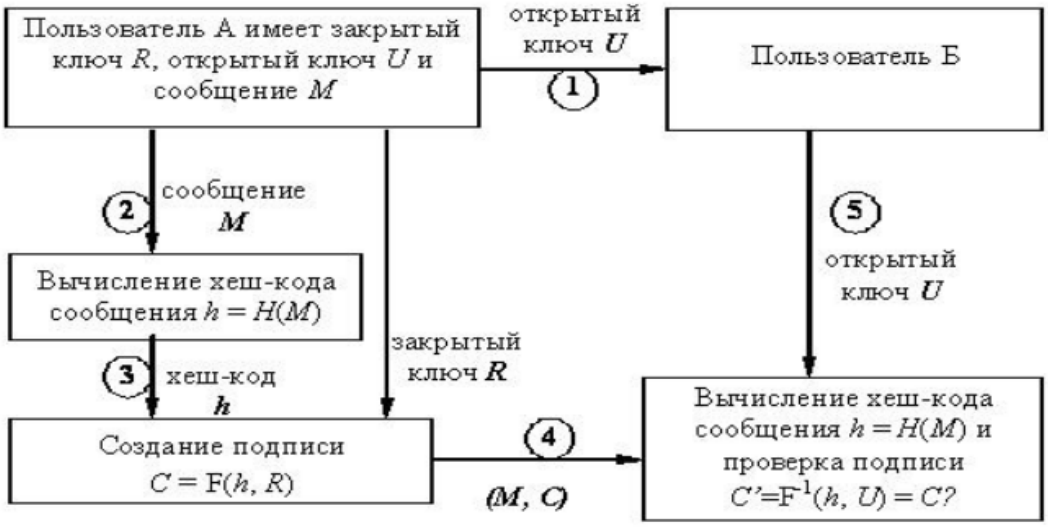
На рис. 9.2 представлена схема формування так званого **цифрового підпису з відновленням документа**. Цифрові підписи з відновленням документа містять в собі документ, що підписується: у процесі перевірки підпису автоматично визначається і тіло документа. Якщо при розшифровуванні повідомлення відновилося правильно, значить, підпис був вірним. Цифровий підпис з відновленням документа може бути реалізований, наприклад, за допомогою одного з найпопулярніших алгоритмів формування ЕЦП - RSA.

У разі використання цифрового підпису з відновленням документа все повідомлення цілком підписується, тобто шифрується. В даний час на практиці так зазвичай не робиться. Алгоритми шифрування з відкритим ключем досить повільні, крім того, для підтвердження цілісності повідомлення потрібно багато пам'яті. До того ж практично всі застосовувані алгоритми обчислення ЕЦП використовують для розрахунку повідомлення, заздалегідь заданої стандартної довжини. Наприклад, у російському алгоритмі формування цифрового підпису ГОСТ Р34.10-94 цей розмір визначений рівним 32 байтам. Тому для економії часу та обчислювальних ресурсів, а також для зручності роботи асиметричний алгоритм зазвичай використовується разом з якою-небудь односпрямованою хеш-функцією. У цьому випадку спочатку з допомогою хеш-функції з повідомлення довільної довжини обчислюється хеш-код потрібного розміру, а потім для обчислення ЕЦП проводиться шифрування отриманого хеш-коду від повідомлення.

ЕЦП, обчислені за хеш-кодом документа, називають **приєднаними цифровими підписами**. Такі цифрові підписи являють собою деякий числовий код, який необхідно пристиковувати до підписуваних документів. Саме повідомлення при цьому не шифрується і передається у відкритому вигляді разом з цифровим підписом відправника.

Якщо користувач **А** хоче відправити користувачеві **Б** повідомлення ***M***, доповнене приєднаним цифровим підписом, то процедура створення та перевірки підпису повинна складатися з наступних кроків:

1. Користувач **А** посилає користувачеві **Б** свій відкритий ключ ***U*** по будь-якому каналу зв'язку, наприклад, по електронній пошті.
2. Користувач **А** за допомогою деякої надійної хеш-функції ***H*** обчислює хеш-код свого повідомлення ***h = H (M)***.
3. Потім користувач **А** шифрує хеш-код повідомлення ***h*** своїм закритим ключем ***R*** і отримує цифровий підпис ***C***.
4. Початкове повідомлення ***M*** разом з цифровим підписом ***C*** пересилаються користувачеві **Б**.
5. Користувач **Б** обчислює хеш-код ***h*** отриманого повідомлення ***M***, а потім перевіряє цифровий підпис ***C***, використовуючи відкритий ключ користувача **А.**



Описаний процес створення підпису не забезпечує конфіденційність. Тобто повідомлення, надіслане таким способом, неможливо змінити, але можна прочитати. Навіть якщо не використовувати хеш-функцію, а шифрувати всі повідомлення цілком, конфіденційність не забезпечується, оскільки будь-хто може розшифрувати повідомлення, використовуючи відкритий ключ відправника.

Вимоги до алгоритмів шифрування з відкритим ключем

Розглянувши основні способи застосування алгоритмів шифрування з відкритим ключем, вивчимо вимоги, яким повинен, задовольняти алгоритм шифрування з відкритим ключем. Ці вимоги такі:

1. Обчислювально легко створювати пару (відкритий ключ, закритий ключ).
2. Обчислювально легко зашифрувати повідомлення відкритим ключем.
3. Обчислювально легко розшифрувати повідомлення, використовуючи закритий ключ.
4. Обчислювально неможливо, знаючи відкритий ключ, визначити відповідний закритий ключ.
5. Обчислювально неможливо, знаючи тільки відкритий ключ і зашифроване повідомлення, відновити вихідне повідомлення.

Ключові терміни

Алгоритм шифрування з відкритим ключем (або асиметричні криптоалгоритми) - криптографічний алгоритм, в якому для шифрування і розшифрування використовуються різні ключі.

Закритий ключ - ключ, використовуваний в асиметричних криптографічних алгоритмах, який повинен зберігатися в секреті.

Одностороння функція - математична функція, яку відносно легко вирахувати, але важко знайти за значенням функції відповідне значення аргументу. Тобто, знаючи *x* легко обчислити *f(x)*, але за відомим *f(x)* важко знайти підходяще значення *x*.

Одностороння функція з люком (або з секретом) - це особливий вид односторонніх функцій, що мають деякий секрет (люк), що дозволяє відносно швидко обчислити зворотне значення функції.

Відкритий ключ - ключ, використовуваний в асиметричних криптографічних алгоритмах, який може не зберігатися в секреті.

Приєднані цифрові підписи - підписи, обчислені за хеш-кодом документа. Такі цифрові підписи являють собою деякий числовий код, який необхідно пристиковувати до підписуваного документу. Саме повідомлення при цьому не шифрується і передається у відкритому вигляді разом з цифровим підписом відправника.

Цифровий (електронний) підпис (digital signature) - унікальний числовий додаток до переданої інформації, що дозволяє перевірити її авторство. Електронний (цифровий) підпис (ЕЦП) являє собою послідовність біт фіксованої довжини, яка обчислюється певним чином за допомогою змісту підписаної інформації та секретного ключа.

Цифрові підписи з відновленням документа - підписи, які містять в собі підписаний документ: у процесі перевірки підпису автоматично обчислюється і тіло документа. Якщо при розшифровуванні повідомлення відновилося правильно, значить, підпис був вірним.

Короткі підсумки

Асиметричні алгоритми шифрування (або алгоритми з відкритим ключем) - криптографічні алгоритми, в яких один ключ використовується для шифрування, а інший, відмінний від першого, - для розшифрування. Алгоритми називаються асиметричними, так як ключі шифрування і розшифрування різні, отже, відсутня симетрія основних криптографічних процесів. Один з двох ключів є відкритим (public key) і може бути оголошений всім, а другий - закритим (private key) і повинен триматися в секреті. Який з ключів, відкритий або закритий, використовується для шифрування, а який для розшифрування, визначається призначенням криптографічної системи.

Всі алгоритми шифрування з відкритим ключем засновані на використанні односторонніх функцій. Односторонньої функцією називається математична функція, яку відносно легко вирахувати, але важко знайти за значенням функції відповідне значення аргументу. Використовувати односторонні функції для шифрування повідомлень з метою їх захисту не має сенсу, так як назад розшифрувати зашифроване повідомлення вже не вийде. Для цілей шифрування використовуються односторонні функції з люком (або з секретом) - особливий вид односторонніх функцій, що мають деякий секрет (люк), що дозволяє відносно швидко обчислити зворотне значення функції.

Алгоритми шифрування з відкритим ключем можна використовувати для вирішення наступних завдань:

1. Для шифрування переданих і збережених даних для їх захисту від несанкціонованого доступу.
2. Для формування цифрового підпису під електронними документами.
3. Для розподілу секретних ключів, використовуваних потім при шифруванні документів симетричними методами.

Цифровий (електронний) підпис - унікальний числовий додаток до переданої інформації, що дозволяє перевірити її авторство. Електронний (цифровий) підпис (ЕЦП) являє собою послідовність біт фіксованої довжини, яка обчислюється певним чином за допомогою змісту підписаної інформації та секретного ключа. Розрізняють приєднані цифрові підписи і цифрові підписи з відновленням документа. Приєднані цифрові підписи - підписи, обчислені за хеш-кодом документа. Такі цифрові підписи являють собою деякий числовий код, який необхідно

пристиковувати до підписаного документу. Саме повідомлення при цьому не шифрується і передається у відкритому вигляді разом з цифровим підписом відправника. Цифрові підписи з відновленням документа - підписи, які як би містять в собі підписаний документ: у процесі перевірки підпису автоматично обчислюється і тіло документа. Якщо при розшифровуванні повідомлення відновилося правильно, значить, підпис був вірним.

Питання для самоперевірки

1. Чим асиметричні алгоритми шифрування відрізняються від симетричних?
2. Для вирішення яких завдань можуть на практиці застосовуватися алгоритми шифрування з відкритим ключем?
3. Які математичні функції називаються односторонніми? Для чого вони можуть застосовуватися в криптографії?
4. Що таке цифровий підпис?
5. Який алгоритм формування цифрового підпису при використанні алгоритмів шифрування з відкритим ключем?
6. Які вимоги пред'являються до асиметричних алгоритмах?

Лекція 6.
КРИПТОГРАФІЧНІ АЛГОРИТМИ З ВІДКРИТИМ КЛЮЧЕМ ТА ЇХ ВИКОРИСТАННЯ

Анотація: В цій лекції викладені найбільш відомі криптографічні алгоритми з відкритим ключем: RSA, алгоритм Діффі-Хеллмана, алгоритм Ель-Гамаля. Опис кожного з алгоритмів супроводжується докладним прикладом. Також в цій лекції сформульовані принципи роботи криптографічних систем на еліптичних кривих.

Алгоритм RSA. Основні відомості

Алгоритм шифрування з відкритим ключем RSA був запропонований одним з перших в кінці 70-х років XX століття. Його назва складається з перших літер прізвищ авторів: Р.Райвеста (R.Rivest), А.Шаміра (A.Shamir) і Л.Адлемана (L.Adleman). Алгоритм RSA є, напевно, найбільш популярним і широко застосовуваним асиметричним алгоритмом в криптографічних системах.

Алгоритм заснований на використанні того факту, що завдання розкладання великого числа на прості співмножники є важким. Криптографічна система RSA базується на наступних двох фактах з теорії чисел:

- задача перевірки числа на простоту є порівняно легкою;
- задача розкладання чисел виду $n = pq$ (p і q - прості числа) на множники є дуже важкою, якщо ми знаємо тільки n, а p і q - великі числа (це так звана задача факторизації).

Алгоритм RSA являє собою блоковий алгоритм шифрування, де зашифровані і незашифровані дані повинні бути представлені у вигляді цілих чисел між 0 і $n-1$ для деякого n.

Шифрування

Отже, розглянемо сам алгоритм. Нехай абонент **А** хоче передати зашифроване повідомлення абоненту **Б**. У цьому випадку абонент **Б** повинен підготувати пару (відкритий ключ; закритий ключ) і відправити свій відкритий ключ користувачеві **А**.

Першим етапом є генерація відкритого та закритого ключів. Для цього спочатку вибираються два великих простих числа P і Q. Потім обчислюється похідна N:

$$N = PQ.$$

Після цього визначається допоміжне число f:

$$f = (P - 1)(Q - 1).$$

Потім випадковим чином вибирається число $d < f$ і взаємно просте з f.

Далі необхідно знайти число e, таке, що

$$ed \bmod f = 1.$$

Числа d і N будуть відкритим ключем користувача, а значення e - закритим ключем.

Таким чином, на цьому етапі у користувача повинна бути інформація, зазначена в наступній таблиці:

	Відкритий ключ	Закритий ключ
Користувач	N, d	e

Так як користувач **Б** хоче отримати зашифроване повідомлення від користувача **А**, значить користувач **Б** повинен відправити свій відкритий ключ (***d, N***) користувачеві **А**. Числа **P** і **Q** більше не потрібні, однак їх не можна нікому повідомляти; найкраще їх взагалі забути.

На цьому етап підготовки ключів закінчений і можна використовувати основний протокол RSA для шифрування даних.

Другий етап - шифрування даних. Якщо абонент **А** хоче передати деякі дані абоненту **Б**, він повинен представити своє повідомлення в цифровому вигляді і розбити його на блоки $m_1, m_2, m_3, ...$, де $m_i < N$. Зашифроване повідомлення складатиметься з блоків c_i.

Абонент **А** шифрує кожен блок свого повідомлення за формулою

$$c_i = m_i^d \bmod N,$$

використовуючи відкриті параметри користувача **Б**, і пересилає зашифроване повідомлення $C = (c_1, c_2, c_3, ...)$ по відкритій лінії.

Абонент **Б**, який отримав зашифроване повідомлення, розшифровує всі блоки отриманого повідомлення за формулою

$$m_i = c^e \bmod N$$

Всі розшифровані блоки будуть точно такими ж, як і вихідні від користувача **А**.

Зловмисник, що перехоплює всі повідомлення і знає всю відкриту інформацію, не зможе знайти вихідне повідомлення при великих значеннях **P** і **Q**.

Приклад обчислень за алгоритмом

Нехай користувач **А** хоче передати користувачеві **Б** повідомлення. У цьому випадку спочатку користувач **Б** має підготувати відкритий і закритий ключі. Нехай їм обрані, наприклад, наступні параметри:

P=3, Q=11, N=3x11=33.

Тоді $f = (P - 1)(Q - 1) = (3-1)(11-1) = 20$.

Потім користувач **Б** обирає будь-яке число ***d***, яке не має спільних дільників з ***f*** (це необхідно для того, щоб зашифроване повідомлення можна було потім однозначно відновити). Нехай $d = 13$. Це число буде одним з компонентів відкритого ключа.

Далі необхідно знайти число ***e***, яке можна буде використати в якості закритого ключа для розшифрування повідомлення. Значення ***e*** має задовольняти співвідношення $ed \bmod f = 1$.

Для малих значень ***f*** число ***e*** можна знайти підбором. У загальному випадку для пошуку ***e*** можна використовувати узагальнений алгоритм Евкліда. У нашому випадку підходить $e = 17$. (Перевіряємо: $13 * 17 \bmod 20 = 221 \bmod 20 = 1$.)

Тепер користувач **Б** повинен запам'ятати свій закритий ключ **17**, відправити відкритий ключ $(13, 33)$ користувачу **А** і знищити числа $P = 3$ і $Q = 11$.

Користувач **А**, який отримав відкритий ключ $(13, 33)$, побачивши, що $N = 33$, розбиває вихідне повідомлення на три блоки, причому значення кожного менше ***N***. Наприклад, нехай є три блоки $m_1 = 8, m_2 = 27, m_3 = 5$. Потім користувач **А** шифрує кожен блок:

$$c_1 = 8^{13} \bmod 33 = 17$$

$c_2 = 27^{13} \bmod 33 = 15$

$c_3 = 5^{13} \bmod 33 = 26$

Зашифроване повідомлення, що складається з трьох блоків ***(17, 15, 26)***, передається користувачеві **Б**, який, використовуючи свій закритий ключ $e = 17$ і $N = 33$, розшифровує повідомлення:

$m_1 = 17^{17} \bmod 33 = 8$

$m_2 = 15^{17} \bmod 33 = 27$

$m_3 = 26^{17} \bmod 33 = 5$

Таким чином, абонент **Б** розшифрував повідомлення від абонента **А.**

Питання практичного використання алгоритму RSA

Протягом багатьох років алгоритм RSA активно використовується як у вигляді самостійних криптографічних продуктів, так і в якості вбудованих засобів в популярних додатках. Відкрите шифрування на базі алгоритму RSA застосовується в популярному пакеті шифрування PGP, операційній системі Windows, різних Інтернет-браузерах, банківських комп'ютерних системах. Крім того, різні міжнародні стандарти шифрування з відкритим ключем і формування цифрового підпису використовують RSA в якості основного алгоритму.

Для забезпечення високої надійності шифрування необхідно, щоб виступаюче в якості модуля число ***N*** було дуже великим - кілька сотень або тисяч біт. Тільки в цьому випадку буде практично неможливо за відкритими параметрам визначити закритий ключ. Так, відомо, що в кінці 1995 року вдалося практично реалізувати розкриття шифру RSA для 500-значного модуля. Для цього за допомогою мережі Інтернет було задіяно більше тисячі комп'ютерів.

Самі автори RSA рекомендували використовувати такі розміри модуля ***N: 768 біт*** - для приватних осіб; ***1024 біт*** - для комерційної інформації; ***2048 біт*** - для особливо секретної інформації. З моменту отримання їх рекомендацій пройшов якийсь час, тому сучасні користувачі повинні робити поправки у бік збільшення розміру ключів. Однак, чим більше розмір ключів, тим повільніше працює система. Тому збільшувати розмір ключа без необхідності не має сенсу.

З розміром ключів пов'язаний і інший аспект реалізації RSA - *обчислювальний*. При використанні алгоритму обчислення необхідні як при створенні ключів, так і при шифруванні / розшифруванні, при цьому, чим більше розмір ключів, тим важче проводити розрахунки. Для роботи з величезними числами доводиться використовувати апарат довгої арифметики. Числа, що складаються з багатьох сотень біт, не вміщаються в регістри більшості мікропроцесорів і їх доводиться обробляти по частинах. При цьому як шифрування, так і розшифрування включають піднесення великого цілого числа в цілу степінь ***по модулю N***. При прямих розрахунках проміжні значення були б неймовірними. Щоб спростити процес обчислень використовують спеціальні алгоритми для роботи з великими числами, засновані на властивостях модульної арифметики, а також оптимізацію при піднесенні до степеня.

Алгоритм RSA реалізується як програмним, так і апаратним шляхом. Багато світових фірм випускають спеціалізовані мікросхеми, що виробляють шифрування алгоритмом RSA. Програмні реалізації значно повільніші, ніж апаратні. До переваг програмного шифрування RSA належить можливість гнучкого налаштування параметрів, можливість інтеграції в різні програмні пакети. В цілому, і програмна, і апаратна реалізації RSA вимагають для виконання приблизно в тисячі разів більшого часу в порівнянні з симетричними алгоритмами, наприклад ГОСТ 28147-89.

Алгоритм RSA може використовуватися для формування електронного цифрового підпису, а також і для обміну ключами. Можливість застосування алгоритму RSA для отримання електронного підпису пов'язана з тим, що секретний і відкритий ключі в цій системі

рівноправні. Кожен з ключів, ***d*** або ***e***, можуть використовуватися як для шифрування, так і для розшифрування. Ця властивість виконується не у всіх криптосистемах з відкритим ключем.

Алгоритм Ель-Гамаля. Основні відомості

Асиметричний алгоритм, запропонований в 1985 році Ель-Гамаль (T. ElGamal), універсальний. Він може бути використаний для вирішення всіх трьох основних завдань: для шифрування даних, для формування цифрового підпису та для узгодження спільного ключа. Крім того, можливі модифікації алгоритму для схем перевірки пароля, докази ідентичності повідомлення й інші варіанти. Безпека цього алгоритму заснована на труднощах обчислення дискретних логарифмів. Цей алгоритм формує загальний секретний ключ для абонентів, що передають один одному повідомлення, і потім повідомлення шифрується шляхом множення його на цей ключ.

І у випадку шифрування, і в разі формування цифрового підпису кожному користувачеві необхідно згенерувати пару ключів. Для цього вибирається деяке велике просте число **P** і число **A**, такі, що різні степені **A** являють собою різні числа по модулю **P**. Числа **P** і **A** можуть передаватися у відкритому вигляді і бути спільними для всіх абонентів мережі.

Потім кожен абонент групи вибирає своє секретне число X_i, $1 < X_i < P-1$, і обчислює відповідне йому відкрите число

$$Y_i : Y_i = A^{X_i} \bmod P$$

Таким чином, кожен користувач може згенерувати закритий ключ X_i і відкритий ключ Y_i.

Інформація про установки системи зведена в наступну таблицю

	Общие параметры	Открытый ключ	Закрытый ключ
Пользователь 1		Y_1	X_1
…	***P, A***	…	…
Пользователь *i*		Y_i	X_i

Шифрування

Тепер розглянемо, яким чином проводиться шифрування даних. Повідомлення, призначене для шифрування, має бути представлено у вигляді одного числа або набору чисел, кожне з яких менше **P**. Нехай користувач **1** хоче передати користувачеві **2** повідомлення ***m***. У цьому випадку послідовність дій така.

1. Перший користувач вибирає випадкове число ***k***, взаємно просте з ***P-1***, і обчислює числа

$$r = A^k \bmod P, \qquad e = m \times Y_2^k \bmod P$$

де Y_2 - відкритий ключ користувача **2**. Число ***k*** тримається в секреті.

2. Пара чисел ***(r, e)***, які є шифротекстом, передається другому користувачеві.
3. Другий користувач, отримавши ***(r, e***), для розшифрування повідомлення обчислює

$$m = e \times r^{P-1-X_2} \bmod P$$

де X_2 - закритий ключ користувача **2**. У результаті він отримує вихідне повідомлення ***m.***

Якщо зловмисник дізнається або перехопить ***P, A,*** Y_2***, r, e***, то він не зможе по них розкрити ***m***. Це пов'язано з тим, що противник не знає параметр ***k***, обраний першим користувачем для шифрування повідомлення ***m***. Обчислити яким-небудь чином число ***k*** практично неможливо, так як це завдання дискретного логарифмування. Отже, зловмисник не може обчислити і значення ***m***, так як ***m*** було помножено на невідоме йому число. Противник також не може

відтворити дії законного одержувача повідомлення (другого абонента), так як йому не відомий закритий ключ X_2 (обчислення X_2 на підставі Y_2 - також задача дискретного логарифмування).

За аналогічним алгоритмом може проводитися і узгодження ключа, використовуваного для симетричного шифрування великих обсягів даних. Більше того, алгоритм Ель-Гамаля на практиці доцільно використовувати саме для узгодження спільного ключа сесії, а не прямого шифрування великих повідомлень. Це пов'язано з тим, що в алгоритмі використовуються операції зведення в ступінь і множення за великим модулем. Так само як і в алгоритмах RSA, операції проводяться над великими, що складаються з декількох сотень або тисяч біт, числами. Тому шифрування великих повідомлень проводиться вкрай повільно.

Приклад шифрування

Нехай два абонента, що обмінюються через Інтернет зашифрованими повідомленнями, мають такі загальні параметри:

P = 11, A = 7.

Крім того, користувачі 1 і 2 мають пари закритих і відкритих ключів, обчислювані також, як в п. 5.3.3:

Користувач 1: закритий ключ X1 = 3, відкритий ключ Y1 = 7^3 mod 11 = 2,
Користувач 2: закритий ключ X2 = 9, відкритий ключ Y2 = 7^9mod 11 = 8.

Перший абонент бажає передати другому повідомлення. Для цього перший абонент запитує з центру розподілу ключів відкритий ключ другого абонента Y2 = 8. Тепер він може зашифрувати своє повідомлення, яке в числовому вигляді нехай має значення m = 9.

Перший абонент вибирає випадково число k, наприклад k = 7. Число k повинно бути взаємно простим з P-1. Значення k = 7 не має спільних дільників з P-1 = 10, значить, воно нам підходить. Перший абонент шифрує своє повідомлення за формулами:

$r = A^k \bmod P = 7^7 \bmod 11 = 6$
$e = m * Y_2^k \bmod P = 9 * 8^7 \bmod 11 = 7$

Пара чисел (6, 7) буде являти собою шифротекст і передається другому користувачеві. Другий користувач, отримавши (6,7) і використовуючи свій закритий ключ X2 = 9 для розшифрування повідомлення, обчислює

$$m = e \times r^{P-1-X_2} mod P = 7 \times 6^{11-1-9} mod 11 = 7 \times 6^1 mod 11 = 9$$

У результаті він дійсно отримує вихідне повідомлення m.

Алгоритм Діффі-Хеллмана

Основні відомості

Перша публікація даного алгоритму з'явилася в 70-х роках XX століття в статті Діффі і Хеллмана, в якій вводилися основні поняття криптографії з відкритим ключем. Алгоритм Діффі-Хеллмана не застосовується для шифрування повідомлень або формування електронного підпису. Його призначення - в розподілі ключів. Він дозволяє двом або більше користувачам обмінятися без посередників ключем, який може бути використаний потім для симетричного шифрування. Це була перша криптосистема, яка дозволяла захищати інформацію без використання секретних ключів, що передаються по захищених каналах. Схема відкритого

розподілу ключів, запропонована Діффі і Хеллманом, здійснила справжню революцію в світі шифрування, так як знімала основну проблему класичної криптографії - проблему розподілу ключів.

Алгоритм заснований на труднощах обчислень дискретних логарифмів. Спробуємо розібратися, що це таке. У цьому алгоритмі, як і в багатьох інших алгоритмах з відкритим ключем, обчислення проводяться по модулю деякого великого простого числа P. Спочатку спеціальним чином підбирається деяке натуральне число *A*, менше *P*. Якщо ми хочемо зашифрувати значення *X*, то обчислюємо

$Y=A^X \mathrm{mod} P.$

Причому, маючи *X*, обчислити *Y* легко. Зворотнє завдання обчислення *X* з *Y* є досить складним. Експонента *X* якраз і називається дискретним логарифмом *Y*. Таким чином, знаючи про складність обчислення дискретного логарифма, число *Y* можна відкрито передавати по будь-якому каналу зв'язку, так як при великому модулі *P* початкове значення *X* підібрати буде практично неможливо. На цьому математичному факті заснований алгоритм Діффі-Хеллмана для формування ключа.

Формування загального ключа

Нехай два користувача, яких умовно назвемо користувач 1 і користувач 2, бажають сформувати загальний ключ для алгоритму симетричного шифрування. Спочатку вони повинні вибрати велике просте число P і деяке спеціальне число A, 1 <A <P-1, таке, що всі числа з інтервалу [1, 2, ..., P-1] можуть бути представлені як різні ступені A mod P. Ці числа мають бути відомі всім абонентам системи і можуть вибиратися відкрито. Це будуть так звані загальні параметри.

Потім перший користувач вибирає число X_1 ($X_1 < P$), яке бажано формувати за допомогою датчика випадкових чисел. Це буде закритий ключ першого користувача, і він повинен триматися в секреті. На основі закритого ключа користувач 1 обчислює число

$$Y_1 = A^{X_1} \, mod \, P$$

яке він посилає другого абоненту.

Аналогічно робить і другий користувач, генеруючи X_2 і обчислюючи

$$Y_2 = A^{X_2} \, mod \, P$$

Це значення користувач 2 відправляє першому користувачеві.

Після цього у користувачів повинна бути інформація, зазначена в таблиці нижче:

	Общие параметры	Открытый ключ	Закрытый ключ
Пользователь 1	P, A	Y_1	X_1
Пользователь 2		Y_2	X_2

З чисел Y_1 і Y_2, а також своїх закритих ключів кожен з абонентів може сформувати загальний секретний ключ *Z* для сеансу симетричного шифрування.

Ось як це має зробити перший користувач:

$$Z = (Y_2)^{X_1} \, mod \, P$$

Ніхто інший крім користувача 1 цього зробити не може, так як число X_1 секретне. Другий користувач може отримати те ж саме число *Z*, використовуючи свій закритий ключ і відкритий ключ свого абонента у такий спосіб:

$$Z = (Y_1)^{X_2} \, mod \, P$$

Якщо весь протокол формування загального секретного ключа виконаний вірно, значення *Z* у одного і другого абонента повинні вийти однаковими. Причому, що найважливіше,

противник, не знаючи секретних чисел X_1 і X_2, не зможе обчислити число Z. Не знаючи X_1 і X_2, зловмисник може спробувати вирахувати Z, використовуючи тільки передані відкрито P, A, Y_1 і Y_2. Безпека формування загального ключа в алгоритмі Діффі-Хеллмана випливає з того факту, що, хоча відносно легко обчислити експоненти по модулю простого числа, дуже важко обчислити дискретні логарифми. Для великих простих чисел розміром сотні і тисячі біт завдання вважається нерозв'язним, так як вимагає колосальних витрат обчислювальних ресурсів.

Користувачі 1 і 2 можуть використовувати значення Z в якості секретного ключа для шифрування і розшифрування даних. Таким же чином будь-яка пара абонентів може обчислити секретний ключ, відомий тільки їм.

Приклад обчислень за алгоритмом

Нехай два абонента, бажаючі обмінюватися через Інтернет зашифрованими повідомленнями, вирішили сформувати секретний ключ для чергового сеансу зв'язку. Нехай вони мають такі загальні параметри:

$P = 11,\ A = 7.$

Кожен абонент вибирає секретне число X і обчислює відповідне йому відкрите число Y. Нехай обрані

$X_1 = 3,\ X_2 = 9.$

Обчислюємо:

$Y_1 = 7^3 \bmod 11 = 2,$

$Y_2 = 7^9 \bmod 11 = 8.$

Потім користувачі обмінюються відкритими ключами Y_1 і Y_2. Після цього кожен з користувачів може обчислити загальний секретний ключ:

користувач 1: $Z = 8^3 \bmod 11 = 6.$

користувач 2: $Z = 2^9 \bmod 11 = 6.$

Тепер вони мають загальний ключ *6*, який не передавався по каналу зв'язку.

Питання практичного використання алгоритму Діффі-Хеллмана

Для того, щоб алгоритм Діффі-Хеллмана працював правильно, тобто обидва користувачі, які беруть участь в протоколі, отримували одне й те саме число Z, необхідно правильним чином вибрати число A, що використовується в обчисленнях. Число A повинно володіти наступними властивостями:

всі числа виду $A \bmod P,\ A^2 \bmod P,\ A^3 \bmod P, \ldots, A^{P-1} \bmod P$

повинні бути різними і складатися з цілих додатніх значень в діапазоні від *1* до *P-1* з деякими перестановками. Тільки в цьому випадку для будь-якого цілого $Y < P$ і значення A можна знайти єдину експоненту X, таку, що

$Y = A^X \bmod P,$ де $0 <= X <= (P-1)$

При довільно заданому P завдання вибору параметра A може виявитися важким завданням, пов'язаним з розкладанням на прості множники числа *P-1*.

На практиці можна використовувати наступний підхід, рекомендований фахівцями. Просте число P вибирається таким, щоб виконувалося рівність $P = 2q + 1$, де q - також просте число. Тоді як A можна взяти будь-яке число, для якого справедливі нерівності:

$1 < A < P-1$ і $A^q \bmod P \neq 1$

На підбір відповідних параметрів A і P потрібен певний час, однак це звичайно не критично для системи зв'язку і не уповільнює її роботу. Ці параметри є загальними для цілої групи користувачів. Вони зазвичай вибираються один раз при створенні спільноти користувачів, що бажають використовувати протокол Діффі-Хеллмана, і не змінюються в

процесі роботи. А ось значення закритих ключів рекомендується кожен раз міняти і вибирати їх за допомогою генераторів псевдовипадкових чисел.

Слід зауважити, що даний алгоритм, як і всі алгоритми асиметричного шифрування, вразливий для атак типу "man-in-the-middle" ("людина в середині"). Якщо противник має можливість не тільки перехоплювати повідомлення, але і замінювати їх іншими, він може перехопити відкриті ключі учасників, створити свою пару відкритого і закритого ключа і послати кожному з учасників свій відкритий ключ. Після цього кожен учасник визначить ключ, який буде спільним з противником, а не з іншим учасником. Способи запобігання такої атаки і деяких інших розглянуті в кінці цієї лекції.

Питання для самоперевірки

1. Для яких цілей може застосовуватися алгоритм RSA?
2. Опишіть процес шифрування з використанням алгоритму RSA.
3. Для яких цілей може застосовуватися алгоритм Діффі-Хеллмана?
4. Опишіть послідовність дій при використанні алгоритму Діффі-Хеллмана.
5. Для яких цілей може застосовуватися алгоритм Ель-Гамаля ?.
6. Опишіть послідовність дій при використанні алгоритму Ель-Гамаля.
7. Які атаки можливі при використанні алгоритмів шифрування з відкритим ключем?