1. Симетричні криптосистеми (з секретним ключем – secret key systems) – в основі даних криптосистем закладено збереження в таємниці ключа шифрування. Процеси шифрування і дешифрування використовують один і той самий ключ. Секретність ключа є постулатом. Основна проблема при застосуванні симетричних криптосистем для зв'язку полягає в складності передачі секретного ключа обом сторонам. Проте дані системи мають високу швидкодію. Розкриття ключа зловмисником загрожує розкриттям тільки тієї інформації, що була зашифрована на цьому ключі. Найбільш вживаний американський стандарт шифрування DES, AES і ГОСТ 28147-89 – всі ці алгоритми є представниками симетричних криптосистем.

|  |  |
| --- | --- |
| **Тип** | **Опис** |
| DES (Data Encryption  Standard) | Відомий алгоритм шифрування, стандарт шифрування даних урядом США.  Шифрується блок з 64 біт, використовується 64-бітовий ключ (потрібний тільки 56 битий), 16 проходів  Може працювати в 4 режимах:   * Електронна кодова книга - звичайний DES, використовує два різні алгоритми. * Ланцюговий режим (CBC-Cipher Block Chaining), в якому шифрування дешифрування блоку даних має залежність від результатів шифрування/дешифрування попередніх блоків даних. * Зворотний зв'язок по виходу (OFB-Output Feedback), використовується як генератор випадкових чисел. * Зворотний зв'язок по шифратору (CFB-Cipher Feedback), використовується для отримання код автентифікації повідомлень. |
| 3-DES або  потрійний DES | 64-бітовий блоковий шифратор, використовує DES 3 рази з трьома різними 56-бітовими ключами.  Достатньо стійкий до всіх атак |
| Каскадний 3-DES | Стандартний потрійний DES, до якого доданий механізм зворотного зв'язку, такий як CBC, OFB або CFB  Надто стійкий до всіх атак. |
| FEAL (швидкий  алгоритм шифрування) | Блоковий шифратор. Використовується як альтернатива DES  Був розкритий, проте після цього отримав нові версії. |
| IDEA (міжнародний  алгоритм шифрування) | 64-бітовий блоковий шифратор, 128-бітовий ключ, 8 проходів  Запропонований недавно; ще не пройшов повну перевірку на надійність, але вважається кращим, чим DES |
| Skipjack | Розроблене АНБ в ході проектів уряду США "Clipper" і "Capstone".  Був секретним до недавнього часу. Має гарну стійкість.  64-бітовий блоковий шифратор, 80-бітові ключі використовуються в режимах ECB, CFB, OFB або CBC, 32 проходи |
| RC2 | 64-бітовий блоковий шифратор, ключ перемінного розміру  Швидше ніж DES в 2 рази  Як і DES може використовуватися в режимі потрійного шифрування.  Конфіденційний, його власник - RSA Data Security |
| RC4 | Потоковий шифр, що має змінного розміру ключ.  Швидше за DES майже в 10 разів.  Конфіденційний, його власник - RSA Data Security |
| RC5 | Має розмір блоку 32, 64 або 128 битий, ключ з довжиною від 0 до 2048 битий, від 0 до 255 проходів  Швидкий блоковий шифр  Конфіденційний, його власник - RSA Data Security |
| CAST | 64-бітовий блоковий шифратор, має ключі від 40 до 64 біт, 8 проходів.  Невідомо способів розкрити його іншим чином ніж за допомогою прямого перебору. |
| Blowfish. | 64-бітовий блоковий шифратор, ключ змінного розміру до 448 битий, 16 проходів, на кожному проході виконуються перестановки, що залежать від ключа, і підстановки, що залежать від ключа і даних.  Швидше чим DES  Спроектований для 32-бітових машин |
| Пристрій з  одноразовими ключами | Шифратор, який не можна розкрити.  Ключем (який має ту ж довжину, що і шифровані дані) є 'n' біт взяті з масиву біт створених випадковим чином, які зберігаються в даному пристрої. Відправника і отримувач мають однакові пристрої. Після використання біти стираються, і потім використовуються нові біти. |
| Потокові шифри | Швидкі алгоритми симетричного шифрування, що зазвичай оперують бітами (а не їх блоками).  Є аналогом пристрою з одноразовими ключами, і хоча не є такими ж безпечними, як вони, але принаймні практичні. |

2. Асиметричні криптосистеми (системи відкритого шифрування, з відкритим ключем) – суть цих криптосистем є в тому, що для шифрування і дешифрования використовуються різні перетворення.

Суть їх полягає в тому, що з кожним адресатом інформаційної системи генеруються два ключі, зв'язані між собою за певним правилом. Один ключ є закритим, а інший відкритим. Відкритий ключ публікується і є доступним кожному, хто має бажання надіслати повідомлення адресатові. Секретний ключ має зберігатися в таємниці.

Вихідний текст зашифровується відкритим ключем адресата і передається одержувачу. З допомогою відкритого ключа не можливо в принципі розшифрувати той же зашифрований текст. Для розшифрування повідомлення необхідне використанням закритого ключа, що відомий тільки самому адресатові.

Таким чином, пропадає проблема передачі секретного ключа (як у симетричних систем). Проте, не дивлячись на всі свої переваги, ці криптосистеми достатньо трудомісткі і повільні. Стійкість асиметричних криптосистем базується, в основному, на алгоритмічній важкості вирішити за прийнятний час яку-небудь задачу. Якщо зловмисникові зможе побудувати такий алгоритм, то дискредитована буде вся система і всі повідомлення, зашифровані за допомогою її. Саме це є головною небезпекою асиметричних криптосистем на відміну від симетричних.

Таким чином для гарантування надійного захисту інформації, до режимів з відкритим ключем надаються дві важливих і очевидних вимоги:

1. Перетворення вхідного тексту повинне бути необоротним і виключати його відновлення з допомогою відкритого ключа.

2. Повинне бути неможливим визначення закритого ключа на основі відкритого.

Алгоритми шифрування з відкритим ключем стали достатньо популярними в сучасних інформаційних системах. Приклади – системи відкритого шифрування RSA, система Рабіна, Ель Гамаля, система на основі еліптичних рівнянь та інші.

|  |  |  |
| --- | --- | --- |
| **Тип** | | **Опис** |
| RSA | Популярний алгоритм асиметричного шифрування. Його стійкість має залежність від складності факторизації цілих великих чисел. | |
| ECC (криптосистема  на основі  еліптичних кривих) | Використовує алгебраїчні системи , яка описуються в термінах точок еліптичних кривих, для реалізації асиметричного алгоритму шифрування.  Є конкурентом по відношенню до інших асиметричних алгоритмів шифрування, оскільки при еквівалентній стійкості використовує ключі меншої довжини і має велику продуктивність.  Є набагато ефективнішим, ніж інші алгоритми з відкритими ключами. Має продуктивність набагато вище, ніж у такі алгоритми як RSA, Діффі-Хеллман і DSA. | |
| Ель-гамаль. | Варіант Діффі-Хеллмана. Може використовуватися як для електронного підпису, так і для шифрування, | |

3. Хеш-функція, або геш-функція — функція, що перетворює вхідні дані будь-якого (як правило великого) розміру в дані фіксованого розміру.

Хешування (гешування) — перетворення вхідного масиву даних довільної довжини у вихідний бітовий рядок фіксованої довжини. Такі перетворення також називаються хеш-функціями, або функціями згортання, а їхні результати називають хешем, хеш-кодом, хеш-сумою, або дайджестом повідомлення (message digest).

Хеш-функція використовується зокрема у структурах даних — хеш-таблицях, широко вживаних у програмному забезпеченні для швидкого пошуку даних. Хеш-функції використовуються для оптимізації таблиць та баз даних за рахунок того, що у однакових записів однакові значення хеш-функції. Такий підхід пошуку дублікатів ефективний у файлах великого розміру. Прикладом цього буде знаходження подібних ділянок у послідовностях ДНК. Криптографічна геш-функція дозволяє легко перевірити, що деякі вхідні дані зіставляються із заданим значенням хешу, але, якщо вхідні дані невідомі, то навмисно важко відновити вхідне значення (або еквівалентну альтернативу), знаючи збережене значення хеш-функції. Це використовується для забезпечення цілісності переданих даних, і є будівельним блоком для HMACs, які забезпечують аутентифікацію повідомлень.

Гешування застосовується для побудови асоціативних масивів, пошуку дублікатів в серіях наборів даних, побудови унікальних ідентифікаторів для наборів даних, контрольного підсумовування з метою виявлення випадкових або навмисних помилок при зберіганні або передачі, для зберігання паролів в системах захисту (у цьому випадку доступ до області пам'яті, де знаходяться паролі, не дозволяє відновити сам пароль), при виробленні електронного підпису (на практиці часто підписується не саме повідомлення, а його геш-образ).

У загальному випадку однозначної відповідності між вихідними даними і геш-кодом немає в силу того, що кількість значень геш-функцій менша, ніж число варіантів значень вхідного масиву. Існує безліч масивів з різним вмістом, що дають однакові геш-коди — так звані колізії. Імовірність виникнення колізій відіграє важливу роль в оцінці якості геш-функцій.

Розроблено багато алгоритмів гешування з різними властивостями (розрядність, обчислювальна складність, криптостійкість тощо). Вибір тієї чи іншої геш-функції визначається специфікою розв'язуваної задачі. Найпростішими прикладами геш-функцій можуть служити контрольна сума або CRC.