

ВЕЛИКОТЪРНОВСКИ УНИВЕРСИТЕТ

“СВ. СВ. КИРИЛ И МЕТОДИЙ”

ФАКУЛТЕТ “ПЕДАГОГИЧЕСКИ”

катедра “Информационни технологии”

КУРСОВ ПРОЕКТ

Защита на данните

**Тема:**

**“Симетрични криптографски системи”**

Разработил: Научен ръководител: Михаил Николов Леисков Доц.д-р Т.Трифонов

Фак. № МПИ-10

Специалност:

“Информационни системи”

ВЕЛИКО ТЪРНОВО

- 2011 –

**1. Въведение**

#### В съвременния свят информацията е една от най-ценните стоки. Развитието на телекомуникациите и системите за пренос и съхраняване на данни улесниха достъпът до различни по обем и характер данни, както на отделни лица така и на големи групи от потребители. Упростените процедури, прилагани широко в различните мрежи, доведоха до значителни усложнения, свързани с гарантиране сигурността на данните при отсъствие на мерки за тяхната защита.

#### 

#### Някои от най-често срещаните прояви на посегателство върху информацията са:

#### 

#### Неоторизиран достъп.

#### Неоторизирано изменение на информацията.

#### Неоторизирано използване на сервизни функции.

#### Отклоняване на транзакции и отказ от обслужване.

Криптографията осигурява различни методи за превърщането на читимата информация в нечитими данни, които могат сигурно и без загуби на да бъдат предавани от една точка до друга и с помоща на ключ тези данни да бъдат отново трансформирани в информация която да бъде разбираема за получателя.

Корените на криптографията започват от преди хиляди години. Тогава криптирането ставало елементарно но все пак било достаточно за предаване на поверителна информация. Един такав шифър е шифъра на Цезар, кадето всяка буква от азбуката се замества с друга буква, която се намира на фиксиран брой позиции напред. Методът е кръстен на Юлий Цезар, който го използвал за комуникация с неговите генерали. Този шифър в денешно време не е широко ползван тъй като не предоставя возможност за сигурна комуникация, но се позлват други който се базират на него.

По време на втората световна война, са изобратени машини с които криптираните съобщения стават още по сложни и трудни за разбиване. Тогава били изобратени и по-усовършени алгоритми за разбиване на криптираните съобщения.

Развитието на компютри и електрониката след втората световна война предоставили нови възможности за разработване на много по-сложени шифри. Освен това компютрите позволяват криптиране на всякакви данни които могат да бъдат представени в двоичен формат, докато класическите шифъри позволяват криптиране само на текстови данни. Съвремените компютърни алгоритми за криптиране и декриптиране са много ефективни и бързи и изискват малко ресурси като памет и процесорно време за да обработат дадена част информация. Докато разбиването на алгоритама изисква много повече усилия и време от всеки класически шифър, така че на практика ефективното разбиване е невъзможно.

**2. Симетрични криптографски системи**

Симетричните криптографски системи(Едноключови крипто системи) са класическите системи за защита на информацията.  За шифроване и разшифроване на информацията във съобщението при тях се използва един и същ секретен ключ. Съхраняването му в тайна осигурява надеждността на защитата на информацията. Подателят обработва информацията, използвайки ключа по определен алгоритъм, данните се предават по комуникационния канал и след това се възстановяват при получателя. Противно на разпространеното мнение, съществуват ключове, които позволяват криптираните данни да не могат да бъдат възстановени по никакъв друг начин освен с валиден ключ.

Такива ключове отговарят на следните изисквания:

* Дължината на ключа е равна или по-голяма от тази на съобщението
* Ключът представлява случайна последователност (псевдо-случайните

последователности не са неуязвими);

* Ключът се използва само веднъж



Фиг.1

Симетричните криптографски системи(Фиг.1) често се използват като съставни части на най-разнообразни криптографски примитиви, а не само за криптиране. Криптирането на съобщението не гарантира че това съобщение няма да се промени докато то се криптира. Затова много често към криптираното съобщение се добавя код, който гарантира, че получателя ще разбере ако съобщението бъде променено. Кодовете за автентикация на съобщенията могат да бъдат произведени от симетричните шифри. Друго приложение на симетричните шифри е при изграждане на хеш функции, кадето от какви и да било входни данни с различна дължина се получават изходни променени данни с фиксирана дължина. При хеш функциите винаги за едни входни данни се получават същи изходни данни. Ако при входните данни е променен само един бит, на изхода се получава изцяло различен хеш код. При хеш кодовете се гарантира,че те ще бъдат винаги различни за различни входни данни и няма да има колизия.

**3. Предимства и недостатъци**

**Предимства:**

* Голяма скорост на криптиране и декриптиране както при хардуерна, таки при софтуерна реализация.
* Лесна реализация
* Използват се като съставни части на най-разнообразни криптографски примитиви.

**Недостатъци:**

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

**4. Методи за симетрично криптиране.**

Симетричните криптографски системи се подразделят според методите на криптиране: *поточни*, *блокови* и *комбинирани*.

**4.1 Поточни методи за симетрично криптиране.**

При поточните криптографски системи съобщението се криптира байт по

байт. Всеки байт се комбинира с друг байт с ползване на **or** или **xor** операция. Всяка буква от текстовото соопщени се комбинира с друга буква или цифра последователно и преобразуването на последователните цифри варира по време на криптирането. На практика цифрите са единични битове или байтове. Поточните системи обикновено криптират с по-висока скорост от блоковите системи и са по леки за хардуерна реализация. Потоковите шифъри модат да бъдат лесно разбити ако не се ползват правилно. Алгоритама за криптиране има два входа. На едния вход се подава бита или байта, който трябва да се криптира, а на другия се подава ключовия бит или байт с, когото трябва да се извърши **or** или **xor** операцията.



Фиг.2

За декодиране на съобщението получателя трябва да знае каква е редицата от битове или байтове, които са ползвани за криптиране. Най-често ключовите симболи се генерират последователно от друг алгоритъм , за който е дадено входящо число, генерирано от поредица от различни числа и винаги за същто входящо число се генерира съща поредица от числа. Трябва да се внимава да не се ползва същото начално число повече пъти за генериране на ключовия поток. За декриптиране на съобщението се извършва същата операция както за криптирането. Потоковите методи често се използват поради тяхната скорост и простотата за хардуерна имплементация, когато информацията идва в потоци с не позната дължина. На пример за защита на безжичната (wireless) връзка. Ако е трябвало да се ползват блокови методи за защита на безжичната вързка, дизайнерите на хардуера и софтуера ще трябва да избират между ефективното предаване на данните или сложноста за изпълнение, тей като блоковите методи не могат да работат с блокове които са по кратки от дефинирания размер на блока. На пример ако системата

работи с блок от 128 бита и се получени само 32 бита, данните не могат да бидат декриптирани, докато не се получи блок с размер от 128 бита. За да се избегне тоя случай, трябва да се дефинира най-малка единица за работа или да се допълва всеки блок с нулеви байти.

**4.1 Блокови методи за симетрично криптиране**

За разлика от поточните системи, блоковите криптиращи системи действат на големи блокове от цифри с фиксирани не променими трансформации. При блоковите системи се взимат няколко байта и те се криптират като едно цяло. Най-често са ползват блокове с 64 бита. На входа на алгоритама за блоково симетрично криптиране, се подава блок с дължина 128 бита от текстовото съобщение, а на изхода се получава блок с дължина 128 бита трансформирано криптирано съобщение. Точната трансформация се контролира с помоща на втори вход, където се подава тайния ключ с който трябва да се извърши криптирането. Алгоритама за разшифриране е подобен на алгоритама за криптиране. На входа се подава 128 бита кодирано съобщение на втория вход се подава ключа, а на изхода след трансформацията се получава 128 битов блок от текстовото съобщение. Кодирането на съобщения, много по-дълги от 128 бита, се разбиват на блокове и се кодират отделно. Алгоритама AES (Advanced Encryption Standard) ползва блокове с 128 бита за криптиране и е по-сигурен и по-ефективен систем.

Блоковите методи имат повече модове по-които да работят. Някои такива модове са: ECB (Electronic codebook), CBC (Cipher-block chaining), PCBC (Propagating cipher-block chaining), CFB (Cipher feedback), OFB (Output feedback), CTR (Counter). Всички тези модове се нуждаят от начален вектор с цел да се създаде първия блок и да се продължи процесът така че за криптиране на всеки блок да се създава друг вектор, който се получава от входния вектор, криптиращия блок или входното съобщение. Тук няма нужда вектора да бъде таен, достаточно е този вектор да не се използва никога заедно със същия ключ.



Фиг.3

**5. Сигурност**

Алгоритмите на симетричните криптогравски системи подлежат на атаки няколко видове атаки които могат да се применат за намиране на ключа или за

декриптиране на соопщението.

За намиране на ключа може да се използва метода на грубата сила (brute

force), при което се прави опит да се декриптира соопщението с всичките

възможни ключове които могат да се генерират. Метода на грубата сила не е един

от най успешните методи за разбиване на алгоритама. Тои изисква много време за

намиране на точния ключ и то време нараства с нарастването на ключа. Но метода

на грубата сила може да разбие алгоритама и да намери точния ключ ако

потребителя използва къса ключова дума.

Друг начин за разбиване на алгоритама е с криптоанализ. Рядко се търсат

ключовете за да се дознае съдържанието на криптиранато соопщение. Много по

често се ползват методи които ползват комплексни математически изчисления и

компютърна мощност. При този потход се декриптират соопщенията без да е

неопходимо да се знае ключа. Със криптоанализ може да се намери ключа или да

се декриптира дадено соопщение.

За да се намери ключа със криптоанализ нужно е едно некриптирано и едно

криптирано соопщение. След намирането на ключа, този ключ се ползва за

криптиране и декриптиране на други соопщения.

Има и други методи за декриптиране на соопщенията и намиране на ключа с

криптоанализ. Некои от тях се прилагат на голям брой блокове от криптирани

соопщения които не са различават много и слет това се върши споредба на

резултатите. Този потход се нарича диференциялен криптоанализ.

**6. Заключение**

Симетричните криптографски системи са бързи и лесни за имплементация

като хардуерно така и софтуерно. Има различни методи които го подобряват

криптирането на данните. Симетричните криптографски системи намират голяма

примена в техниката. Те се ползват при криптирането при пренос на данните, за

генериране на хаш ключове които гарантират не промененост на данните като при

самия пренос така и при самия процес на криптиране и декриптиране.

Недостатъците се че за всяка двоика потребители трябва да се генерира отделен

ключ и те ключове трябва да се пазат в тайна през цялото време. Също така се

изисква често смяна на ключа което поражда проблеми с разпространението на

ключовете. Те не са удобни за използване в механизмите за електронен подпис

защото изискват много голями ключове за проверяващата трансформация.

Литература

**1.** w ww.google.com

**2.** en.wikipedia.org/wiki/Cryptography

**3.** en.wikipedia.org/wiki/Symmetric-key\_algorithm

**4.** en.wikipedia.org/wiki/Stream\_cipher

**5.** en.wikipedia.org/wiki/Cryptographic\_hash\_function

**6.** en.wikipedia.org/wiki/One-way\_compression\_function

7. en.wikipedia.org/wiki/Block\_cipher\_modes\_of\_operation\_\_