Практическая работа №7

ИССЛЕДОВАНИЕ МЕТОДОВ УПРАВЛЕНИЯ КРИПТОГРАФИЧЕСКИМИ КЛЮЧАМИ

Цель работы. Изучить принципы управления ключами при криптографической защите информации.

Краткие сведения из теории

Как бы ни была сложна и надежна сама криптосистема, она основана на использовании ключей. Если для обеспечения конфиденциального обмена информацией между двумя пользователями процесс обмена ключами тривиален, то в системе, где количество пользователей составляет десятки и сотни управление ключами, – это серьезная проблема.

Под ключевой информацией понимается совокупность всех действующих в системе ключей. Если не обеспечено достаточно надежное управление ключевой информацией, то, завладев ею, злоумышленник получает неограниченный доступ ко всей информации.

Управление ключами – информационный процесс, включающий в себя три элемента:

– генерацию ключей;

– накопление ключей;

– распределение ключей.

Рассмотрим, как они должны быть реализованы для того, чтобы обеспечить безопасность ключевой информации.

Генерация ключей. В реальных системах используются специальные аппаратные и программные методы генерации случайных ключей. Как правило используют датчики случайных чисел. Однако степень случайности их генерации должна быть достаточно высокой. Идеальными генераторами являются устройства на основе “натуральных” случайных процессов. Например, генерация ключей на основе белого радиошума. Другим случайным математическим объектом являются десятичные знаки иррациональных чисел, например π или e, которые вычисляются с помощью стандартных математических методов.

В системах со средними требованиями защищенности вполне приемлемы программные генераторы ключей, которые вычисляют случайные числа как сложную функцию от текущего времени и (или) числа, введенного пользователем.

Накопление ключей. Под накоплением ключей понимается организация их хранения, учета и удаления.

Поскольку ключ является самым привлекательным для злоумышленника объектом, открывающим ему путь к конфиденциальной информации, то вопросам накопления ключей следует уделять особое внимание.

Секретные ключи никогда не должны записываться в явном виде на носителе, который может быть считан или скопирован.

В достаточно сложной системе один пользователь может работать с большим объемом ключевой информации, и иногда даже возникает необходимость организации минибаз данных по ключевой информации. Такие базы данных отвечают за принятие, хранение, учет и удаление используемых ключей.

Каждая информация об используемых ключах должна храниться в зашифрованном виде. Ключи, зашифровывающие ключевую информацию называются м а с т е р - к л ю ч а м и . Желательно, чтобы мастер-ключи каждый пользователь знал наизусть и не хранил их вообще на каких-либо материальных носителях.

Очень важным условием безопасности информации является периодическое обновление ключевой информации в системе. При этом переназначаться должны как обычные ключи, так и мастер-ключи. В особо ответственных системах обновление ключевой информации необходимо производить ежедневно.

Вопрос обновления ключевой информации связан и с третьим элементом управления ключами – распределением ключей.

Распределение ключей. Распределение ключей – самый ответственный процесс в управлении ключами. К нему предъявляются два требования:

– оперативность и точность распределения;

– скрытность распределяемых ключей.

В последнее время заметен сдвиг в сторону использования криптосистем с открытым ключом, в которых проблема распределения ключей отпадает. Тем не менее распределение ключевой информации в системе требует новых эффективных решений.

Распределение ключей между пользователями реализуются двумя разными подходами:

1 *Путем создания одного или нескольких центров распределения ключей.* Недостаток такого подхода состоит в том, что в центре распределения известно, кому и какие ключи назначены, и это позволяет читать все сообщения, циркулирующие в системе. Возможные злоупотребления существенно влияют на защиту.

2 *Прямой обмен ключами* между пользователями системы. В этом случае проблема состоит в том, чтобы надежно удостоверить подлинность субъектов.

В обоих случаях должна быть гарантирована подлинность сеанса связи. Это можно обеспечить двумя способами:

1 *Механизм запроса-ответа*, который состоит в следующем. Если пользователь А желает быть уверенным, что сообщения, которые он получает от пользователя В, не являются ложными, он включает в посылаемое для В сообщение непредсказуемый элемент (запрос). При ответе пользователь В должен выполнить некоторую операцию над этим элементом (например, добавить 1). Это невозможно осуществить заранее, так как не известно, какое случайное число придет в запросе. После получения ответа с результатами действий пользователь А может быть уверен, что сеанс является подлинным. Недостатком этого метода является возможность установления, хотя и сложной, закономерности между запросом и ответом.

2 *Механизм отметки времени.* Он подразумевает фиксацию времени для каждого сообщения. В этом случае каждый пользователь системы может знать, насколько “старым” является пришедшее сообщение.

В обоих случаях следует использовать шифрование, чтобы быть уверенным, что ответ послан не злоумышленником и штемпель отметки времени не изменен.

При использовании отметок времени встает проблема допустимого временно́го интервала задержки для подтверждения подлинности сеанса. Ведь сообщение с отметкой времени в принципе не может быть передано мгновенно. Кроме этого, компьютерные часы получателя и отправителя не могут быть абсолютно синхронизированы.

Для обмена ключами можно использовать криптосистемы с открытым ключом, используя тот же алгоритм RSA.

Но весьма эффективным оказался алгоритм Диффи-Хелмана, позволяющий двум пользователям без посредников обменяться ключом, который может быть использован затем для симметричного шифрования.

Алгоритм Диффи-Хеллмана. Диффи и Хелман предложили для создания криптографических систем с открытым ключом функцию дикретного возведения в степень.

Необратимость преобразования в этом случае обеспечивается тем, что достаточно легко вычислить показательную функцию в конечном поле Галуа, состоящим из p элементов (p – либо простое число, либо простое в любой степени). Вычисление же логарифмов в таких полях – значительно более трудоемкая операция.

Для обмена информацией первый пользователь выбирает случайное число x_1, равновероятное из целых чисел от 1 до $p - 1$. Это число он держит в секрете, а другому пользователю посылает число $y_1 = \alpha^{x_1} \bmod p$, где α – фиксированный элемент поля Галуа $GF(p)$, который вместе с p заранее распространяется между пользователями.

Аналогично поступает и второй пользователь, генерируя x_2 и вычислив y_2, отправляя его первому пользователю. В результате этого они оба могут вычислить общий секретный ключ $k_{12} = \alpha^{x_1 x_2} \bmod p$.

Для того, чтобы вычислить k_{12}, первый пользователь возводит y_2 в степень x_1 и находит остаток от деления на p. То же делает и второй пользователь, только используя y_1 и x_2. Таким образом, у обоих пользователей оказывается общий ключ k_{12}, который можно использовать для шифрования информации обычными алгоритмами. В отличие от алгоритма RSA, данный алгоритм не позволяет шифровать собственно информацию.

Не зная x_1 и x_2, злоумышленник может попытаться вычислить k_{12}, зная только перехваченные y_1 и y_2. Эквивалентность этой проблемы проблеме вычисления дискретного логарифма есть главный и открытый вопрос в системах с открытым ключом. Простого решения до настоящего времени не найдено. Так, если для прямого преобразования 1000-битных простых чисел требуется 2000 операций, то для обратного преобразования (вычисления логарифма в поле Галуа) – потребуется около 10^{30} операций.

Для примера выберем $p = 43$. α можно определить как первый простой множителель числа $(p - 1)$. Первым простым множителем числа $(43 - 1) = 42$ является 3. Между двумя пользователями распределяется пара чисел (43, 3). Пользователь *A* придумывает свой секретный ключ, допустим, число 8 и посылает пользователю *B* число $y_1 = 3^8 \pmod{43} = 25$. Пользователь B придумывает свой секретный ключ, допустим, число 37 и посылает пользователю *A* число $y_2 = 3^{37} \pmod{43} = 20$. Оба пользователя вычисляют общий секретный ключ k_{12}: пользователь *A* – $k_{12} = 20^8 \pmod{43} = 9$; пользователь *B* – $k_{12} = 25^{37} \pmod{43} = 9$.

Как видно, при всей простоте алгоритма Диффи-Хелмана, вторым его недостатком по сравнению с системой RSA является отсутствие гарантированной нижней оценки трудоемкости раскрытия ключа.

Кроме того, хотя описанный алгоритм позволяет обойти проблему скрытой передачи ключа, необходимость аутентификации остается. Без дополнительных средств, один из пользователей не может быть уверен, что он обменялся ключами именно с тем пользователем, который ему нужен.

Порядок выполнения работы

1 Изучить краткие сведения из теории.

2 Для решения задачи из пп. 3 по первой цифре шифра из таблицы 1 необходимо выбрать параметры шифрования.

Таблица 1 – **Параметры алгоритмов шифрования**

Параметр	Первая цифра шифра									
	1	2	3	4	5	6	7	8	9	0
p	45	39	41	37	49	57	61	33	48	53

x_1	13	31	15	5	9	31	42	5	30	14
x_2	27	9	21	22	13	12	10	23	16	22

3 Создать общий ключ для двух пользователей с использованием алгоритма Диффи-Хелмана с параметрами p, x_1, x_2.

Содержание отчета

1 Цель работы.
2 Исходные данные.
3 Результаты расчетов.
4 Вывод по работе.

Контрольные вопросы

1 Что входит в управление ключами?
2 Генерация ключей.
3 Накопление ключей.
4 Распределение ключей.
5 Алгоритм Диффи-Хеллмана.