

Задача

Вариант 2.

Реализовать протокол идентификации Guillou-Quisquater.

Требования к работе:

- Разработка двух независимых модулей-участников протокола.
- Реализация должна позволять попытки ложной аутентификации.
- Количество раундов протокола должно быть параметром схемы.

Тестируемый алгоритм

Guillou-Quisquater представляет собой интерактивный протокол, который позволяет доказать, что доказываемое утверждение верно, и доказывающий знает это доказательство, в то же время не предоставляя никакой информации о самом доказательстве данного утверждения. Данный криптографический протокол обладает тремя свойствами:

1. *Полнота*: если утверждение действительно верно, то доказывающий убедит в этом проверяющего с любой наперед заданной точностью.
2. *Корректность*: если утверждение неверно, то любой, даже «нечестный», доказывающий не сможет убедить проверяющего.
3. *Нулевое разглашение*: если утверждение верно, то любой, даже «нечестный», проверяющий не узнает ничего кроме самого факта, что утверждение верно.

Описание алгоритма

Протокол Guillou-Quisquater требует только один раунд обмена сообщениями и состоит из трёх этапов. Схематично их можно изобразить следующим образом:

$A \Rightarrow B$: доказательство

$A \Leftarrow B$: вызов

$A \Rightarrow B$: ответ

Сначала A выбирает из заранее определённого непустого множества некоторый элемент, который становится её секретом — закрытым ключом. По этому элементу вычисляется, а затем публикуется открытый ключ. Знание секрета определяет множество вопросов, на которые A всегда сможет дать правильные ответы. Затем A выбирает случайный элемент из множества, по определённым правилам вычисляет доказательство и затем отправляет его B . После этого B выбирает из всего множества вопросов один и просит A ответить на него (вызов). В зависимости от вопроса, A посылает B ответ. Полученной информации B достаточно, чтобы проверить

действительно ли А владеет секретом.

Схема идентификации:

Сторона А отправляет стороне В свои атрибуты J. Стороне А необходимо убедить сторону В, что это именно ее атрибуты. Для этого сторона А доказывает свое знание секрета x стороне В, не раскрывая при этом ни одного бита самого секрета x. Для этого сторонам потребуется всего 1 раунд.

Алгоритм создания открытого и закрытого ключей:

1. Центр доверия Т выбирает два различных случайных простых числа p и q, после чего вычисляет их произведение $n = p * q$
2. Т выбирает целое число e ($1 < e < \varphi(n)$), взаимно простое со значением функции $\varphi(n)$.
3. Т вычисляет $s = e^{-1} \bmod \varphi(n)$ и секрет $x = J^{-s} \bmod n$
4. Т вычисляет $y = x^e \bmod n$
5. Тройка {n, e, y} публикуется в качестве открытого ключа.
6. x играет роль закрытого ключа и передается стороне А

Обмен сообщениями:

1. А выбирает случайное целое r, находящееся в диапазоне от 1 до n-1. А вычисляет $a = r^e \bmod n$ и отправляет его В.
2. В выбирает случайное целое c, находящееся в диапазоне от 0 до e-1. В посылает c стороне А.
3. А вычисляет $z = rx^c \bmod n$ и отправляет его В.
4. В проверяет: если $z^e = ay^c \bmod n$, то подлинность доказана.

Обмен сообщениями можно повторять заданное количество раундов, чтобы избежать случайного попадания в секретный ключ.

Пример использования алгоритма

Атрибуты J: «sign»

Количество раундов: 3

```
n = 72937285652457759353394074128781524406917004981809048652236461249651826263529
e = 9966880339632639732929882668407260529485772302494114847343276357645876584160780044913572041064998899940454373235699449475851371515163716776307853306576703
x = 65294678052833334900965774626072231492416574928736207838781967483470469384380
y = 38856733131838352453684321016089655040695972800910604003550427550236688208597
B -> True
A <- good true
B -> True
A <- good true
B -> True
A <- good true
A <- very good true
```

Изменим секретный ключ x и проверим:

```
n = 72937285652457759353394074128781524406917004981809048652236461249651826263529
e = 9966880339632639732929882668407260529485772302494114847343276357645876584160780044913572041064998899940454373235699449475851371515163716776307853306576703
x = 65294678052833334900965774626072231492416574928736207838781967483470469384380
y = 38856733131838352453684321016089655040695972800910604003550427550236688208597
B -> False
A <- bad false
```

Вывод

В данной лабораторной работе был реализован протокол идентификации Guillou-Quisquater, который позволяет доказать подлинность сообщения без передачи секретного ключа. Ключ зависит от атрибутов, задаваемых пользователем (данные банковской карты, паспорт), которые потом обрабатываются хеш-функцией. Поэтому при попытке взлома протокол достаточно быстро и просто вычислит попытку атаки, а дополнительные раунды помогут избежать случайных прохождений теста.