ГУАП

КАФЕДРА № 51

ПРЕПОДАВАТЕЛЬ

ассистент		А.М.Буланов
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 6

КРИПТОГРАФИЧЕСКИЕ ПРОТОКОЛЫ

по курсу: КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	5611		П.П.Недошивин
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2019

Задание:

Реализовать схему разделения секрета на китайской теореме об остатках. Разработка двух независимых модулей: первый должен принимать на вход секрет и возвращать его проекции, второй должен брать проекции и возвращать секрет.

Описание алгоритма шифрования:

Разделение секрета — любой из способов распределения секрета среди группы участников, каждому из которых достаётся своя некая доля. Секрет может воссоздать только коалиция участников из первоначальной группы, причём входить в коалицию должно не менее некоторого изначально известного их числа.

Схемы разделения секрета применяются в случаях, когда существует значимая вероятность компрометации одного или нескольких хранителей секрета, но вероятность недобросовестного сговора значительной части участников считается пренебрежимо малой.

Существующие схемы имеют две составляющие: разделение и восстановление секрета. К разделению относится формирование частей секрета и распределение их между членами группы, что позволяет разделить ответственность за секрет между её участниками. Обратная схема должна обеспечить его восстановление при условии доступности его хранителей в некотором необходимом количестве.

В отличие от процедуры разбиения секрета, где $t = n$, в процедуре разделения секрета количество долей, которые нужны для восстановления секрета, может отличаться от того, на сколько долей секрет разделён. Такая схема носит названия **пороговой схемы** (t, n), где n — количество долей, на которые был разделён секрет, а t — количество долей, которые нужны для восстановления секрета.

<u>Китайская теорема об остатках</u>

Если натуральные числа $a_1, a_2, \dots, a_n$ попарно взаимно просты, то для любых целых $r_1, r_2, \dots, r_n$ таких, что $0 \le r_i < a_i$ при всех $i \in \{1,2, \dots, n\}$, найдётся число N, которое при делении на a_i даёт остаток r_i при всех $i \in \{1,2, \dots, n\}$. Более того, если найдутся два таких числа N_1 и N_2, то $N_1 \equiv N_2 \ (mod\ a_1 * a_2 * \dots * a_n)$.

<u>Алгоритм поиска решений</u>

Задача - восстановление числа x по набору его остатков от деления на некоторые заданные взаимно простые числа $a_1, a_2, \dots, a_n$.

Как пример рассмотрим систему:

$$\begin{cases} x \equiv 1 \ (mod\ 2), \\ x \equiv 2 \ (mod\ 3), \\ x \equiv 6 \ (mod\ 7). \end{cases}$$

Для решения системы выпишем отдельно решения первого, второго и третьего уравнений (достаточно выписать решения не превосходящие 2 × 3 × 7 = 42):

$$x_1 \in \{1,3,5,7,9,11, \dots ,39,41,43, \dots \},$$
$$x_2 \in \{2,5,8,11,14, \dots ,38,41,44, \dots \},$$
$$x_3 \in \{6, 13,20,27,34,41,48, \dots \}.$$

Очевидно, что множество решений системы будет пересечение представленных выше множеств. По утверждению теоремы решение существует и единственно с точностью до операции взятия по модулю 42. В нашем случае $x \in \{41,83,125, \dots \}$ или $x \equiv 41 \ (mod\ 42)$.

Схема Асмута-Блума

Схема Асмута — Блума — пороговая схема разделения секрета, построенная с использованием простых чисел. Позволяет разделить секрет (число) между n сторонами таким образом, что его смогут восстановить любые m участников.

Пусть M — некоторый секрет, который требуется разделить. Выбирается простое число p, большее M. Выбирается n взаимно простых друг с другом чисел $d_1, d_2, \dots, d_n$, таких что:

- $\forall i{:}\, d_i > p$
- $\forall i{:}\, d_i < d_{i+1}$
- $d_1 * d_2 * \dots * d_m > p * d_{n-m+2} * d_{n-m+3} * \dots * d_n$

Выбирается случайное число r и вычисляется

$$M' = M + rp$$

Вычисляются доли:

$$k_i = M' \; mod \; d_i$$

Участникам раздаются $\{p, d_i, k_i\}$.

Теперь, используя китайскую теорему об остатках, можно восстановить секрет M, имея m и более долей.

Описание реализации:

Структура secret_s содержит:

1. int p – простое число
2. std::vector<int> d – взаимно простые числа
3. std::vector<int> k – доли секрета

Класс encoder содержит приватные поля:

1. std::vector<int> primes – предварительно сгенерированные простые числа
2. int n – количество участников схемы
3. int m – пороговое число участников

Публичные методы:

1. encoder(int om, int on) – конструктор
2. secret_s encode(int o) – разделение секрета на n долей
3. int decode(secret_s sec) – получение секрета из m долей
4. ~encoder() - деструктор

Пример работы:

Max users: 12
Min users: 8
Secret: 4
p = 4073
d = [4079 4091 4093 4099 4111 4127 4129 4133 4139 4153 4157 4159]
k = [514 2614 3104 715 1092 1046 2248 639 580 3180 2491 124]
Correct