РЕСУРСОЕМКОСТЬ ПОРОГОВЫХ СХЕМ РАЗДЕЛЕНИЯ СЕКРЕТА ПРИ РАБОТЕ С 256-БИТНЫМИ КЛЮЧАМИ

А. В. Зарубин

ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской обл.

Введение

На протяжении всей истории человечества всегда была актуальна проблема защиты информации. Каждый человек стремится к сохранности важных ему данных. Например, каждый владелец банковской карты заинтересован в защите своих денежных средств. Однако есть те, кто желает получить деньги других любой ценой. Уже сейчас мошенники используют считывающие устройства в банкоматах, взломы баз данных банков и т. д. Естественно, сами банки применяют многоступенчатые системы защиты от мошенников. И проблема защиты информации не останавливается на банковской системе. Гораздо серьезнее может представляться атака хакеров на сервера министерства обороны. И это далеко не полный список опасностей, которым подвергаются данные в наши дни. Учитывая то, как быстро развивается компьютерная инфраструктура, возрастают вычислительные мощности процессоров, сохранение, передача и защита информации становится одной из самых актуальных задач современности.

В основе многих криптосистем защиты информации лежат программно-аппаратные комплексы (ПАК) и аппаратные модули защиты конфиденциальной информации(Hardware Security Module-HSM). HSM модули основываются на схемах разделения секрета. По таким схемам секретный ключ (разделяемый секрет) путем математических преобразований делится на N частей секрета и выдается N участникам структуры доступа ПАК. Для восстановления исходного секретного ключа необходимо «собрать вместе» K долей секрета (где $K < N$). Такие схемы называются пороговыми схемами разделения секрета.

За три десятилетия существования, задача разделения секрета превратилась в активно развивающуюся область современной криптографии. Такие системы очень широко применяются в нашей современной жизни, начиная от банковских карт и заканчивая авторизацией на различных сайтах.

В данной работе рассматриваются и анализируются пороговые схемы разделения секрета.

Виды пороговых схем разделения секрета

Схема Шамира

Идея схемы заключается в том, что двух точек достаточно для задания прямой, трех – для задания параболы, четырёх – для кубической параболы, и так далее. Чтобы задать многочлен степени k требуется $k+1$ точек.

Для того чтобы после разделения секрет могли восстановить только k участников, его «прячут» в формулу многочлена степени $(k+1)$ над конечным полем G. Для однозначного восстановления этого многочлена необходимо знать его значения в k точках, причем, используя меньшее число точек, однозначно восстановить исходный многочлен не получится. Количество же различных точек многочлена не ограничено. Кратко данный алгоритм можно описать следующим образом. Пусть дано конечное поле G. Зафиксируем n различных ненулевых несекретных элементов данного поля. Каждый из этих элементов приписывается определённому члену группы. Далее выбирается произвольный набор из t элементов поля G, из которых составляется многочлен $f(x)$ над полем G степени $t-1$, $1 < t \le n$. После получения многочлена высчитываем его значение в несекретных точках и сообщаем полученные результаты соответствующим членам группы [1].

Важным достоинством схемы Шамира является то, что она легко масштабируема. Чтобы увеличить число пользователей в группе, необходимо лишь добавить соответствующее число несекретных элементов к уже существующим, при этом должно выполняться условие при $r_i \neq r_i$ при $i \neq j$.

Анализ сложности математического алгоритма:

– этап разделения секрета на доли: $O(N * K)$;

– этап восстановления секрета: $O(K^2)$.

Анализ ресурсоемкости вычислений:

– этап разделения секрета на доли:

При расчете значений полинома необходимо просчитывать $\boldsymbol{x^{k-1}}$ где x принимает значения от 1 до N. Максимальное значение данной операции достигается при $K = N > 64$. Возьмем $K = N = 128$ откуда получим $\mathbf{128^{127} = 2^{899} \approx 256^{112}}$ = 112 байт;

При расчете значения каждого монома в полиноме мы получим операцию $S^* \boldsymbol{x^{k-1}}$, где значение S выбирается произвольно из, $\mathbf{2^{32} = 256^4}$, откуда максимальная длина значения монома будет 116 байт.

Из вышеперечисленных вычислений следует, что все отдельные операции по вычислению значений мономов полинома требуют операций приведения по модулю.

Значение вычисляемой доли принимает максимальную длину только при финальном сложении значения мономов с секретом.

Вывод: для разделения секрета на доли необходимо $N^* |M| + O(|M|)$ байт оперативной памяти для хранения долей секрета;

– этап восстановления секрета:

при расчете значений интерполяционного многочлена Лагранжа будем просчитывать отдельно операции над делимым и над делителем. В общем случае при $K = N = 128$ получаем произведение $\mathbf{x^{127} = 128^{127} = 2^{899} \approx 256^{112}}$ = 112 байт;

При подсчете суммы получаем (512+889)*128 = = 2 кбайт.

Вывод: для восстановления секрета необходимо больше 112 кбайт оперативной памяти.

Схема Блэкли

Две непараллельные прямые на плоскости пересекаются в одной точке. Любые две некомпланарные плоскости пересекаются по одной прямой, а три некомпланарные плоскости в пространстве пересекаются тоже в одной точке. Вообще n – мерных гиперплоскостей всегда пересекаются в одной точке. Одна из координат этой точки будет секретом. Если закодировать секрет как несколько координат точки, то уже по одной доле секрета (одной гиперплоскости) можно будет получить какую-то информацию о секрете, то есть о взаимозависимости координат точки пересечения.

Схема Блэкли в трёх измерениях: каждая доля секрета - это плоскость, а секрет – это одна из координат точки пересечения плоскостей. Двух плоскостей недостаточно для определения точки пересечения [2].

Схема Блэкли менее эффективна, чем схема Шамира: в схеме Шамира каждая доля такого же размера как и секрет, а в схеме Блэкли каждая доля в t раз больше. Существуют улучшения схемы Блэкли, позволяющие повысить её эффективность.

Анализ сложности математического алгоритма:

– этап разделения секрета на доли: $O(K * N)$;

– этап восстановления секрета: $O(K^3)$.

Анализ ресурсоемкости вычислений:

– этап разделения секрета на доли:

секретная точка Q составляется как вектор из Координат максимальной длины равной $|M|$, откуда мы получаем объем памяти для хранения данной точки равный $K * |M|$ = (в максимальном случае) = = 128 * 64 = 8192 байта, откуда получаем, что всякая точка (уравнение гиперплоскости) будет занимать 8192 байта.

Вывод: для разделения секрета на доли необходимо $N^* K *$ 64 = 1 Мбайт оперативной памяти;

– этап восстановления секрета:

Для восстановления секрета необходимо решить систему линейных алгебраических уравнений. Единственным допущением является то, что найти необходимо всего 1 неизвестное, а именно первое, которое и является секретом. Наилучший вариант для целочисленной арифметики является метод Крамера. Для вычисления значения 1 неизвестной необходимо просчитать 2 определителя матриц размерности. Вычисления производить в поле, откуда размером промежуточных значений можно пренебречь (не превышают длину удвоенного модуля).

Вывод: для восстановления секрета необходимо $K * N^*$ 64 = 1 Мбайт + 128 байт оперативной памяти.

Схема Карнин – Грин – Хеллмана

В 1983 году Карнин, Грин и Хеллман предложили свою схему разделения секрета, которая основывалась на невозможности решить систему с m неизвестными, имея менее m уравнений.

В рамках данной схемы выбираются $n + 1$ m-мерных векторов $\boldsymbol{V_0, V_1, \ldots, V_n}$ так, чтобы любая матрица размером m×m, составленная из этих векторов, имела ранг m. Пусть вектор U имеет размерность m.

Секретом в схеме является матричное произведение $\boldsymbol{U^T \cdot V_0}$. Долями секрета являются произведения $\boldsymbol{U^T \cdot V_i, 1 \le I \le n}$.

Имея любые m долей, можно составить систему линейных уравнений размерности m×m, неизвестными в которой являются коэффициенты U. Решив данную систему, можно найти U, а имея U, можно найти секрет. При этом система уравнений не имеет решения в случае, если долей меньше, чем m [3].

Анализ сложности математического алгоритма:

– этап разделения секрета на доли: $O(N)$;

– этап восстановления секрета: $O(K^3)$.

Анализ ресурсоемкости вычислений:

– этап разделения секрета на доли:

Так как секрет представляется в виде матричного произведения 2-х векторов, то объем оперативной памяти для хранения координат вектора не превышает модуля и равен 64 байта.

Вывод: для разделения секрета на доли необходимо $(N + 1) \cdot 64$ = 8256 байт оперативной памяти;

– этап восстановления секрета:

для восстановления секрета необходимо решить систему линейных алгебраических уравнений. Наилучший вариант для целочисленной арифметики является метод Крамера. Для вычисления значения необходимо просчитать определителей матриц размерности. Вычисления производить в поле, откуда размером промежуточных значений можно пренебречь (не превышают длину удвоенного модуля).

Вывод: для восстановления секрета необходимо $N \cdot 64$ = 8192 байт оперативной памяти.

Схема Асмута – Блума

В 1983 году Асмут и Блум предложили схему разделения секрета основанную на китайской теореме об остатках. Для некоторого произвольного числа вычисляются остатки от деления на последовательность чисел, которые раздаются сторонам. Благодаря ограничениям на последовательность чисел, восстановить секрет может только определённое число сторон [4].

Схема Асмута – Блума является доработанной схемой Миньотта. В отличие от схемы Миньотта, её можно построить в таком виде, чтобы она была совершенной.

Анализ сложности математического алгоритма:

– этап разделения секрета на доли: *O (N)*;

– этап восстановления секрета: *O (K2)*.

Анализ ресурсоемкости вычислений:

– этап разделения секрета на доли:

Каждое простое число *di* занимает объем оперативной памяти больший размерности модуля, но меньший удвоенного модуля ($|di| \approx 100$ байт);

Для проверки условий на нахождение *di* необходимо $2 \cdot K \cdot |di| = 25600$ байт = 25 кбайт оперативной памяти;

Объем оперативной памяти для $M' |M'| = |di| \cdot K =$ = 8192 байта = 8 кбайт;

объем оперативной памяти для хранения 1 доли секрета = $|P| + |di| + |ki| \approx 192$ байта.

Вывод: для разделения секрета на доли необходимо $192 \cdot 128 + 8192 + 25600 = 58368$ байт = 57 кбайт оперативной памяти.

– этап восстановления секрета:

Для восстановления секрета необходимо 24576 байт для хранения параметров и около $5 \cdot |P| = 320$ байт для вычислений.

Результаты исследования схем разделения секрета

Опираясь на сделанные расчеты сложности и ресурсоемкости можно определить наиболее оптимальную пороговую схему разделения секрета.

Схема Шамира является системой с самым низким объемом занимаемой оперативной памяти, так же при увеличении количества участников, на которые делится секрет система не становится более сложной для вычислений, так как для этого достаточно добавить соответствующее число несекретных элементов к уже существующим.

Наиболее неэффективной пороговой схемой разделения секрета является схема Блэкли. Основной ее недостаток в том, что в ней каждая доля секрета в несколько раз больше самого секрета, вследствие чего, она имеет самый большой объем занимаемой оперативной памяти.

Схема Карнин – Грин – Хелламана отличается тем, что восстановить секрет может только полная группа участников, что существенно повышает ее надежность. Однако при повышении количества участников увеличивается количество уравнений, необходимых для восстановления секрета, что пагубно сказывается на ее сложности, а в следствии и на быстродействии.

Схема Асмута – Блума основывается на китайской теореме об остатках, вследствие чего имеет хорошее быстродействие и простой расчет при большом количестве участников, однако, является не самой оптимальной схемой в плане объема занимаемой оперативной памяти.

Принимая во внимание примерно равную производительность каждой системы, и, используя результаты проведенных исследований, можно определить и сопоставить параметры ресурсоемкости, составив таблицу. Данная таблица сформирована с учетом того, что возможное количество участников в пороговой схеме разделения секрета должно быть не меньше 64 ($N \geq 64$), так как это является одним из требований по минимальному количеству участников, определяющих надежность схемы. При этом необходимое пороговое количество участников для восстановления секрета не должно превышать общее количество участников ($K \leq N$).Секретом являлся 256-битный ключ.

Результаты исследования схем разделения секрета

	Объем занимаемой оперативной памяти при разделении секрета	Объем занимаемой оперативной памяти при восстановление секрета
Схема Шамира	12 кбайт	> 112 кбайт
Схема Блэкли	1 Мбайт	1 Мбайт + 128 байт
Схема Карнин – Грин – Хеллмана	≈ 8 Кбайт	8 кбайт
Схема Асмута – Блума	57 кбайт	24 кбайта-хранение параметров 320 байт-вычисление

Заключение

В данной работе проведены исследования основных пороговых схем разделения секрета на основе определения параметра ресурсоемкости каждой схемы.

Основываясь на результатах исследования определено, что схема разделения секрета Шамира является наименее ресурсоемкой схемой. Данную схему, как одну из самых быстродействующих и менее требовательную к ресурсоемкости программно-аппаратных обеспечения, можно успешно применять в пороговых криптосистемах ПАК.

Литература

1. Shamir A. How to share a secret // Com. Of the ACM. – 1979. – Vol. 22, №11. – P.612-613.

2. Blakley G. R. Safeguarding cryptographic keys // Proc. Of AFIPSNasiona1 ComputerConference. -1979. – 48. – P.313-317.

3. C. Asmuth, J. Bloom. A modular approach to key safeguarding // Information Theory, IEEE Transactions on. – 1983. – B. 2. – T. 29.

4. Carnin E. D., Greene J. W., Hellman M. E. On Secret Sharing Systems // IEEE Trans. Inform. Theory. – 1983. – V.29. – №1. – P.231-241.