УДК 681.3

А. А. Демьянчук, Н. А. Молдовян, А. В. Рыжков

Выбор «идеальных» параметров в схеме двухшаговой аутентификации и коммутативном шифре

Предложены способы реализации алгоритмов коммутативного шифрования и двухшаговых протоколов с нулевым разглашением секрета со специальным выбором параметров криптосхем указанных типов. В случае коммутативных шифров достигается устранение слабых сообщений, которые могли бы быть использованы для выполнения атак на основе специально подобранных текстов, а в случае протоколов с нулевым разглашением достигается их упрощение.

Шифрование, коммутативное шифрование, протокол с нулевым разглашением, защита информации

Интересные для практических применений алгоритмы коммутативного шифрования (АКШ) основаны на вычислениях в конечных циклических группах достаточно большого порядка Ω [1], [2]. При этом предполагается, что шифруемые исходные сообщения m представляют собой битовые последовательности некоторой заданной длины. Секретный ключ формируется в виде двух чисел e и d, которые удовлетворяют условию $d = e^{-1} \bmod \Omega$. Шифрование сообщения m осуществляется кодированием его элементом группы, обозначаемым M, и вычислением криптограммы в виде элемента C группы по формуле $C = M^e$ (в случае мультипликативной группы) или по формуле $C = eM$ (в случае аддитивной группы). Расшифровка криптограммы C выполняется по формуле $M = C^d$ или $M = dC$ соответственно с последующим декодированием элемента группы M в сообщение m. В ряде важных для практики случаев АКШ используются в криптографических протоколах, в ходе которых возникают условия, когда потенциальный нарушитель знает исходное шифруемое значение и получаемую из него криптограмму и пытается вычислить секретный ключ. Стойкость АКШ обеспечивается вычислительной сложностью задачи дискретного логарифмирования (ЗДЛ) в используемой циклической конечной группе.

Для АКШ актуальным вопросом является обеспечение достаточно малой вероятности существования сообщений, имеющих малое значение порядка или имеющих порядок, не содержащий ни одного большого простого делителя [2]. В алгоритмах коммутативного шифрования, основанных на трудности ЗДЛ в простом конечном поле $GF(p)$, эта проблема решается соответствующим выбором простого числа p. Для практического использования таких алгоритмов коммутативного шифрования этого достаточно. Однако теоретически идеальным случаем синтеза алгоритмов коммутативного шифрования над ЗДЛ является получение значения большого простого порядка для всех возможных значений шифруемых сообщений. При построении АКШ с использованием вычислений в поле $GF(p)$ принципиально невозможно получение большого значения порядка для всех возможных значений шифруемых сообщений. Это связано с тем, что в таких полях всегда присутствует элемент, имеющий порядок, равный двум.

В настоящей статье описывается задача синтеза «идеального» коммутативного шифра, основанного на трудности ЗДЛ. Для этого используются вычисления в конечном двоичном поле и вычисления на идеальной эллиптической кривой (ЭК). Описаны двухшаговые протоколы с нулевым разглашением секрета, основанные на применении схем открытого согласования ключей, и показано, что используемые «идеальные» параметры для АКШ при применении в указанных протоколах снижают вычислительную сложность шага, связанного с формированием ответа на запрос.

Под идеальным коммутативным шифром понимается такой АКШ, для которого все возможные сообщения имеют большой простой порядок, равный порядку используемой циклической группы. Использование групп с простым значением порядка связано с тем, что наличие сравнительно малых простых делителей порядка позволяет подобрать сообщения, по криптограмме которых можно получить существенную информацию о ключе, применяя для решения ЗДЛ способ, известный как метод больших и малых шагов [3].

Двоичные многочлены, заданные над полем $GF(2)$ и имеющие степень, не превышающую значение $s - 1$, образуют поле двоичных многочленов $GF(2^s)$ при задании операции умножения многочленов по модулю неприводимого двоичного многочлена $\eta(x)$ степени s. Известно, что мультипликативная группа любого конечного поля является циклической. В случае поля двоичных многочленов $GF(2^s)$ порядок мультипликативной группы равен $q = 2^s - 1$. Если выбрать такое значение s, что число q окажется простым, то все двоичные многочлены поля $GF(2^s)$ будут иметь порядок q.

Множество всех возможных значений шифруемых сообщений представляются ненулевыми битовыми цепочками длины s. При этом все сообщения будут иметь одинаковое значение порядка, равное $q = 2^s - 1$. При соответствующем выборе значения s порядок q будет достаточно большим и получение какой-либо информации о секретном ключе будет вычислительно невыполнимым при известных криптограммах, полученных шифрованием любых возможных исходных сообщений.

Коммутативное шифрование с использованием вычислений в поле $GF(2^s)$ осуществляется следующим образом. Генерируется случайный секретный ключ шифрования e ($e < q$). Затем вычисляется секретный ключ расшифровки $d = e^{-1} \bmod q$. Шифрование s-битовых сообщений $\mu(x)$ выполняется по формуле

$$c(x) = \left(\mu(x)\right)^e \bmod \eta(x).$$

Расшифровка криптограммы, представляющей собой двоичный многочлен $c(x)$, осуществляется по формуле

$$\mu(x) = \left(c(x)\right)^d \bmod \eta(x).$$

Рассматривая потенциальные атаки на основе известного или специально подобранного исходного текста, легко заметить, что возникает ЗДЛ в конечном поле многочленов, причем порядок основания логарифма не зависит от сообщения и всегда равен q. Другими словами, сложность возникающей ЗДЛ не зависит от выбираемого сообщения. Это означает, что не существует «слабых» входных сообщений.

Дискретное логарифмирование в поле двоичных многочленов представляет собой вычислительно трудную задачу, если значение s удовлетворяет условию $s \geq 1024$ [2]. Возникает вопрос существования значений $s \geq 1024$, таких, что число $q = 2^s - 1$ является простым. Простые числа вида $2^s - 1$ известны как простые числа Мерсенна. Для следующих значений степени $s \geq 1024$ имеем простые значения q: 1279, 2203, 2281, 3217, 4253, 4423, 9689, 9941, 11 213, 19 937, 21 701, 23 209, …*. Найдено еще много других значении $s \geq 44\ 497$*, задающих простые числа Мерсенна, однако они в настоящее время представляют меньший интерес для использования в предложенном алгоритме коммутативного шифрования, поскольку при их применении уменьшается производительность процедуры коммутативного шифрования.

В криптографии используются ЭК, которые представляет собой конечные множества пар элементов (x, y) некоторого конечного поля F, удовлетворяющих уравнению третьей степени (уравнению ЭК) некоторого достаточно простого вида.

Важнейшим параметром ЭК является ее порядок $\#E$ – число точек на ЭК. Известны общие методы вычисления порядка кривой [5] по заданному полю $GF(p)$ и коэффициентам a и b. В частных случаях формирования ЭК ее порядок может быть вычислен по значениям p, a и b, используя известные достаточно простые формулы. Варианты ЭК, пригодные для построения криптосхем, с указанием значения их порядка приведены в стандарте [6].

Для некоторых ЭК порядок равен простому числу. Такие ЭК называются идеальными. Использование идеальных ЭК для построения коммутативных шифров – еще один способ построения идеальных АКШ.

Удобные и стойкие протоколы строгой аутентификации удаленных пользователей могут быть построены на основе использования схем открытого согласования ключей. Рассмотрим вариант реализации с использованием схемы Диффи–Хелл-

* http://oeis.org/A000043

мана, в которой системными параметрами являются большое простое число p и примитивный элемент α по модулю p. В этой схеме каждый пользователь выбирает случайный секретный ключ x и вычисляет открытый ключ y по формуле $y = \alpha^x \bmod p$.

Открытый ключ делается общеизвестным, и любой желающий имеет принципиальную возможность однозначно вычислить значение секретного ключа x, хотя эта возможность практически нереализуема, если число p имеет размер не менее 1024 бит, а число $p - 1$ содержит простой делитель q размером не менее 160 бит.

Протокол с нулевым разглашением секрета – это такой протокол, многократное выполнение которого не уменьшает сложность вычисления секретного ключа по открытому ключу. Протокол аутентификации удаленных пользователей с нулевым разглашением использует значение α, имеющее по модулю p порядок, равный q, и включает следующие 2 шага:

1. Проверяющий генерирует случайное число k и вычисляет значение своего разового открытого ключа $U = \alpha^k \bmod p$ и значение $Z = y^k \bmod p$, где y – открытый ключ доказывающего (пользователя, подлинность которого проверяется), после чего передает доказывающему значение U в качестве своего запроса, на который он ожидает ответ доказывающего.

2. Доказывающий вычисляет значение ω, которое является порядком числа U по модулю p. Если $\omega = q$, то он вычисляет значение $Z = U^x \bmod p$ и направляет Z проверяющему в качестве своего ответа на полученный запрос. Если ω не равно q, то доказывающий направляет ответ «Некорректный запрос».

Если проверяющий получил правильное значение Z, т. е. то значение, которое он вычислил до направления своего запроса доказывающему, то им делается вывод о подлинности доказывающего. Проверка на втором шаге порядка числа U является принципиальным моментом. Если эта проверка не делается, то становится возможна атака со стороны проверяющего, результатом которой будет вычисление части или всего секретного ключа доказывающего. Действительно, это можно сделать посылая в качестве запроса значения U, порядок которых содержит только небольшие простые делители числа $p - 1$.

Возможны различные варианты реализации описанного протокола с использованием задачи дискретного логарифмирования в скрытой циклической подгруппе конечной некоммутативной группы [7], [8] или на основе трудности задачи дискретного логарифмирования на ЭК. В последнем случае при применении идеальных ЭК обеспечивается существенное уменьшение вычислительной сложности второго шага протокола – для проверки корректности запроса достаточно проверить то, что запрос представляет собой точку ЭК $U = (x_U, y_U)$. Для последнего достаточно вычислить значение $w = \left(x_C^3 + ax_C + b\right) \bmod p$ и индекс Лежандра $\lambda = w/p$. Если $\lambda = 1$, то запрос корректен, в противном случае нет.

Наиболее простая реализация протокола связана с использованием вычислений в поле двоичных многочленов $GF(2^s)$, порядок мультипликативной группы которых q равен простому числу вида $2^s - 1$ (числа Мерсена). Числа такого вида являются простыми только при определенных значениях s. Практический интерес для реализации протокола с нулевым разглашением представляют, например, значения $s = 1279$ и 2203.

Примерами неприводимых двоичных многочленов таких степеней являются трехчлены $\eta_1(x) = x^{1279} + x^{216} + 1$ и $\eta_2(x) = x^{1279} + x^{418} + 1$ [9], а также пятичлен $\eta_3(x) = x^{2203} + x^{14} + x^6 + x^5 + 1$. При выборе указанных трехчленов $\eta_1(x)$ и $\eta_2(x)$ в качестве модуля операция модульного умножения двоичных многочленов в поле $GF(2^{1279})$ может быть выполнена с помощью одной операции арифметического умножения многочленов, двух операций арифметического сдвига и шести операций сложения. Выполнять операцию деления многочленов не требуется, что существенно снижает сложность вычислений в поле $GF(2^{1279})$ и повышает быстродействие протокола аутентификации. В поле $GF(2^{2203})$ операция умножения может быть задана по модулю неприводимого пятичлена $\eta_3(x)$. В этом случае умножение в поле $GF(2^{2203})$ можно выполнить также без осуществления операции деления многочленов, так как оно реализуется с помощью одной операции арифметического умножения многочленов, шести операций арифметического сдвига и десяти операций сложения.

В двухшаговом протоколе аутентификации на основе трудности ЗДЛ в поле $GF(2^{1279})$ открытый ключ формируется по формуле $y(x) = (\alpha(x))^k \bmod \eta_1(x)$, где k – секретный ключ; $\alpha(x)$ – некоторый специфицированный многочлен, а протокол описывается следующими шагами:

1. Проверяющий генерирует случайное число t и вычисляет запрос в виде многочлена $u(x) = (\alpha(x))^t \bmod \eta_1(x)$ и многочлен $z(x) = (y(x))^t \bmod \eta_1(x)$, после чего пересылает доказывающему двоичный многочлен $u(x)$ в качестве своего запроса.

2. Доказывающий вычисляет двоичный многочлен $z'(x) = (u(x))^k \bmod \eta_1(x)$ и направляет $z'(x)$ проверяющему в качестве ответа.

Если $z'(x) = z(x)$, то проверяющий делает вывод о подлинности доказывающего. В этом варианте протокола нет необходимости выполнять проверку корректности запроса, поскольку любая ненулевая битовая строка, интерпретируемая как элемент поля $GF(2^{1279})$, имеет порядок, равный простому числу $2^{1279} - 1$, т. е. никакие запросы со стороны проверяющего не могут быть им использованы для получения хотя бы одного бита информации о ключе.

Предложенный идеальный алгоритм коммутативного шифрования, основанный на трудности ЗДЛ в двоичных полях $GF(2^s)$, представляет практический интерес благодаря тому, что операция умножения в двоичном поле многочленом выполняется достаточно эффективно аппаратными и программными средствами при выборе неприводимых двоичных многочленов с малым числом ненулевых коэффициентов. В алгоритме реализуется идеальная ситуация, характеризующаяся тем, что все возможные входные сообщения имеют одно и то же значение простого порядка большого размера, однако существует ограниченное число значений степени расширения двоичного поля $GF(2^s)$, при которых мультипликативная группа поля имеет простой порядок. В настоящее время представляют интерес для практической реализации разработанного алгоритма следующие 6 значений s: 1279, 2203, 2281, 3217, 4253, 4423, поскольку при их использовании достигается приемлемый компромисс между стойкостью и производительностью. Несмотря на ограниченный выбор значений s, большое разнообразие конкретных вариантов предложенного алгоритма коммутативного шифрования достигается возможностью выбора большого числа различных неприводимых двоичных многочленов $\eta(x)$.

Другим способом реализации идеального коммутативного шифра является его построение с использованием вычислений на ЭК простого порядка. Данный способ также представляет большое разнообразие вариантов построения идеальных АКШ.

В предложенном протоколе аутентификации удаленных пользователей с нулевым разглашением важна проверка корректности запроса проверяющего. Для упрощения этой проверки можно использовать идеальные ЭК, а полностью исключить ее удается при построении двухшагового протокола аутентификации с использованием вычислений в двоичных полях, порядок мультипликативной группы которых является достаточно большим простым числом Мерсенна.

СПИСОК ЛИТЕРАТУРЫ

1. Пат. США № 4424414 / M. E. Hellman, S. C. Pohlig. Exponentiation cryptographic apparatus and method. 1984.

2. Повышение производительности процедур коммутативного шифрования / П. А. Молдовяну, Д. Н. Молдовян, Е. В. Морозова, С. В. Пилькевич // Вопр. защиты информации. 2009. №. 4. С. 24–31.

3. Menezes A. J., Vanstone S. A. Handbook of Applied Cryptography. CRC Press, 1996.

4. Dewaghe L. Remarks on the Schoof-Elkies-Atkin algorithm //Mathematics of Computation of the American Mathematical Society. 1998. Т. 67, № 223. С. 1247–1252.

5. FIPS P. U. B. 186-2. Digital Signature Standard (DSS) //National Institute of Standards and Technology (NIST). 2000.

6. Молдовян Д. Н. Примитивы криптосистем с открытым ключом: конечные некоммутативные группы с четырехмерных векторов // Информационно-управляющие системы. 2010. №. 5. С. 43–50.

7. Moldovyan D. N. Non-commutative finite groups as primitive of public key cryptosystems // Quasigroups Relat. Syst. 2010. Т. 18, № 2. С. 165–176.

8. Болотов А. А., Гашков С. Б., Фролов А. Б. Элементарное введение в эллиптическую криптографию. Протоколы криптографии на эллиптических кривых. М.: КомКнига, 2006.

A. A. Demyanchuk, N. A. Moldovyan, A. V. Ryzhkov

CHOOSING IDEAL PARAMETERS FOR ZERO-KNOWLEDGE AUTHENTICATION PROTOCOLS AND COMMUTATIVE CIPHERS

It has been proposed a method for commutative encryption and zero-knowledge two-step protocols with a special choice of parameters for this cryptoschemes. Advantage of the method is eliminating weak messages in case of commutative encryption and simplification of zero-knowledge protocols.

Encryption, commutative encryption, zero-knowledge protocol, information security

УДК: 20.53.19, 28.23.13

М. С. Куприянов, И. И. Холод, З. А. Каршиев

Метод построения параллельных алгоритмов интеллектуального анализа данных из потоконезависимых функциональных блоков

Описывается метод построения параллельных алгоритмов интеллектуального анализа данных, опирающийся на представление алгоритма в виде иерархии вложенных функциональных потокобезопасных блоков.

Интеллектуальный анализ данных, параллельные алгоритмы, интеллектуальный анализ распределенных данных

В соответствии с формальной моделью [1] алгоритм интеллектуального анализа данных (ИАД) можно декомпозировать на отдельные потокобезопасные функциональные блоки. Каждый блок должен получать на вход обрабатываемые данные, настройки алгоритма и созданную на предыдущих этапах модель знаний. При этом реализуется следующий принцип: данные и настройки не подлежат изменению – меняться может только модель знаний. Изменение модели знаний можно рассматривать как создание внутри блока новой модели знаний на основе переданных в него параметров: набора данных, настройки алгоритма и исходной модели знаний. Таким образом, каждый блок является «монолитным» с точки зрения основного алгоритма. По сути, он представляет собой мини-алгоритм.

Функциональные блоки являются потоко безопасными, если в процессе выполнения своих вычислений не модифицируют значения глобальных переменных. Для достижения этого требования при разбиении алгоритма на блоки необходимо соблюдать следующие правила:

- каждый блок может представлять собой или некоторую операцию, выполняемую над моделью, или некоторый структурный элемент, характерный для алгоритма ИАД;
- блок, реализующий некоторую операцию, должен изменять модель знаний, поданную на вход таким образом, чтобы она оставалась целостной (т. е. должны выполняться все ограничения, накладываемые на модель);
- внутри функционального блока не должно происходить обращений к внешним переменным, вся работа должна выполняться на основании набора данных, настроек и исходной модели знаний, переданных в функциональный блок.

Функциональный блок может содержать как исполняемый код (неразделимую операцию), так и последовательность других функциональных блоков. В общем случае блок может содержать не одну последовательность.

В результате алгоритм ИАД можно представить как иерархию вложенных функциональных потокобезопасных блоков (рис. 1). Сам алгоритм, являющийся корневым функциональным блоком