МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

УДК 519.7 DOI 10.17223/20710410/55/2

КЛЕПТОГРАФИЧЕСКИЕ (АЛГОРИТМИЧЕСКИЕ) ЗАКЛАДКИ В ГЕНЕРАТОРЕ КЛЮЧЕЙ RSA

А. В. Маркелова

ООО «НТЦ Альфа-Проект», г. Москва, Россия

E-mail: a@safe-crypto.ru

Рассмотрены основные виды алгоритмических закладок. Представлен способ построения асимметричных клептографических закладок в генераторе ключей RSA, позволяющий владельцу ключа закладки (разработчику или авторизованной спецслужбе) получать доступ к пользовательскому ключу, сгенерированному инфицированным алгоритмом. Сформулированы теоремы, иллюстрирующие работоспособность описанных алгоритмов, оценена вычислительная сложность этих алгоритмов. Продемонстрирована стойкость построенных закладок к некоторым классам атак даже при условии, что противник знает используемые методы и имеет доступ к исходному коду ключевого генератора.

Ключевые слова: *RSA, клептография, алгоритмическая закладка, лазейка, клептографическая закладка, бэкдор.*

KLEPTOGRAPHIC (ALGORITHMIC) BACKDOORS IN THE RSA KEY GENERATOR

A. V. Markelova

Science and Technology Center “AlphaProject”, Moscow, Russia

A cryptographic (algorithmic) backdoor is the ability of the backdoor key owner to gain an unauthorized access to user’s secret information embedded in the cryptoalgorithm. There are two independent classifications of backdoors: by the level of cryptographic strength (weak, symmetric, asymmetric backdoor) and by the method of implementing undeclared capabilities (based on covert channel or on implicit weakening of the cryptoalgorithm). We present examples of each type of backdoor and discuss a method for constructing an asymmetric backdoor based on an implicit weakening of the algorithm in the RSA key generator. Let it be required to generate a public module of the RSA key $n = pq$, $|n| = L$. We will generate such prime numbers that $|p| = |q| = L/2$. Let D be the backdoor parameter, $|D| = K$; ID is the identifier of the generator instance; i is the key generation counter; $\psi_s(a, ID, i)$ is a one-way (trapdoor) function with the trapdoor s on the first argument. Let $(a, D) = 1$ and

$$r'(a, D, r_0) = \begin{cases} \min\{r : r \geqslant r_0; (rD + a) \text{ is prime}\}, & \text{if } r{<}2^{L/2-K} \text{ and } rD + a{<}2^{L/2}; \\ 0, & \text{otherwise.} \end{cases}$$

Let's choose the function $R(x, y, z, i)$ and define $r_{ID}^{(i)}(a, D) = r'(a, D, R(a, D, ID, i))$. For any random $a_p \in \mathbb{Z}_D^*$ and $r'_0 \in \mathbb{Z}$, $(2^{L/2-1})/D < r'_0 < 2^{L/2-K}$, the following values are uniquely determined:

$$p = r_{ID}^{(i)}(a_p, D)D + a_p,$$
$$q = r'(\psi_s(a_p, ID, i)a_p^{-1} \bmod D, D, r'_0)D + \psi_s(a_p, ID, i)a_p^{-1} \bmod D.$$

At the same time, if $r_{ID}^{(i)}(\cdot) \neq 0$ and $r'(\cdot) \neq 0$, then the numbers p and q are prime, $|p| = |q| = L/2$, $|n| = |pq| \in \{L-1, L\}$. If the numbers p and q are generated in this way, then, provided that the secret s is known, the public module $n = pq$ can be factorized in polynomial time of the key length. Really, $p = r_{ID}^{(i)}(\psi_s^{-1}(n \bmod D, ID, i), D)D + \psi_s^{-1}(n \bmod D, ID, i)$. This approach allows to develop a cryptographically strong key generator, even if the adversary knows the methods used and has access to the source code of the key generator. This allows us to use a backdoor generator even in open source systems. Cryptographic strength depends on the choice of algorithm parameters: in particular, on the level of cryptographic strength of the function $\psi_s(a, ID, i)$.

Keywords: *RSA, kleptography, algorithmic backdoor, trapdoor, kleptographic backdoor, backdoor.*

Введение

Алгоритмическая закладка — это заложенная в криптоалгоритм возможность получения неавторизованного доступа владельца ключа закладки к секретной пользовательской информации.

Клептографическая закладка — это преднамеренная модификация криптографического алгоритма (разработанного изначально без алгоритмической закладки) с целью получения неавторизованного доступа владельца ключа закладки к секретной пользовательской информации при сохранении достаточного уровня стойкости к атакам противника и полиномиальной неотличимости работы модифицированного алгоритма от работы исходного алгоритма.

В данной теме, к сожалению, не выработано единой терминологии: например, ряд авторов называет закладки такого вида лазейками («trapdoor» [1, 2]), когда требуется подчеркнуть их отличие от «обычных» закладок (бэкдоров, «backdoor» [3]), вносящих изменения в программное обеспечение или аппаратную платформу. Реализацию алгоритма с клептографической закладкой можно также назвать инфицированным алгоритмом [4].

Важно отметить, что подобные механизмы могут быть частью основной структуры алгоритма (как в случае DUAL_EC [5]) — в этом случае будем называть их алгоритмическими закладками — или они могут быть привнесены в существующие криптоалгоритмы (например, методами [1, 2, 6–8]. Закладки второго вида будем называть клептографическими, чтобы подчеркнуть их отличие от алгоритмических закладок общего вида — в данном случае мы пользуемся понятийным аппаратом Адама Янга и Моти Юнга, которые в 1996 г. ввели термин «клептография», означающий раздел криптографии, посвящённый изучению закладок (бэкдоров, лазеек) в криптоалгоритмах [1]. Можно считать, что клептографические закладки являются частным случаем алгоритмических.

В настоящей работе под «противником» будем понимать того нарушителя/злоумышленника, который не является участником (разработчиком или пользователем)

информационной системы, то есть в общем случае не знает ни ключа закладки, ни ключа какого-либо пользователя, ни особенностей реализации.

Заметим, что алгоритмические/клептографические закладки можно рассматривать как один из способов доступа авторизованных спецслужб к пользовательской информации. В такой трактовке алгоритмическая закладка является одним из инструментов системы оперативно-розысных мероприятий. Таким образом, владелец ключа закладки может быть как нарушителем (если это недобросовестный разработчик прикладного программного обеспечения), так и честным участником системы — например, спецслужбой, чьей задачей как раз является использование данного ключа для проведения оперативно-розысных мероприятий.

Западные спецслужбы уже несколько десятилетий насаждают внедрение алгоритмических закладок в криптоалгоритмы и информационные системы, причём в последнее время это фактически является их официальной позицией. Изначально идеи базировались на депонировании ключа, как в случае с Clipper Chip [9], затем они трансформировались в концепцию «ответственного шифрования» («responsible encryption» [10]), а в последнее время спецслужбы всё чаще используют термин «исключительный доступ» («exceptional access» [11]).

Подобный ребрендинг представляет собой игру понятиями и не меняет сути явления. Во всех случаях речь идёт о том, что пользовательский секретный ключ или данные, защищённые им, становятся доступны спецслужбам.

Таким образом, данная тема давно вышла за рамки чисто теоретических исследований. Алгоритмические закладки проникают даже в международные криптографические стандарты. Самым ярким примером стал алгоритм генерации псевдослучайных чисел DUAL_EC, наличие закладки в котором на данный момент считается доказанным [5]. DUAL_EC вошёл в стандарт NIST [12] в 2006 г. и просуществовал до 2015 г., будучи внедрённым в ряд криптографических продуктов [13].

При невозможности стандартизировать инфицированную версию алгоритма её использование в программных реализациях может быть объяснено мерами по обфускации кода или оптимизацией вычислений, что позволяет автору закладки скрыть факт её внедрения.

В открытых источниках описано немало случаев, когда западные спецслужбы внедряли те или иные закладки в пользовательское программное обеспечение и криптоалгоритмы [3, 9, 14, 15], но это относится к вопросам недекларированных возможностей и выходит за рамки нашего исследования.

Алгоритмические (и в частности, клептографические) закладки отличаются от недекларированных возможностей программного обеспечения тем, что они используют математическую структуру заражаемых алгоритмов и протоколов: выходные данные криптоалгоритма видоизменяются таким образом, что для стороннего наблюдателя результат неотличим от «честного» алгоритма, а владелец ключа закладки может вычислить какую-либо секретную пользовательскую информацию.

В п. 1 представлена классификация алгоритмических закладок по двум ключевым признакам: уровню стойкости и способу реализации недекларированных возможностей. В п. 2 приведены примеры каждого класса для генератора ключей RSA.

В п. 3 рассмотрен авторский метод построения асимметричной закладки на основе неявного ослабления алгоритма. Такие закладки являются стойкими к атакам противника (в том числе в предположении доступа противника к исходному коду генератора). Пункты 4–6 иллюстрируют этод метод конкретными вариантами встраивания лазеек в генератор ключей RSA.

Общая идея такой асимметричной закладки предложена автором в [16]. В настоящей работе даётся более детальное описание инфицированного алгоритма, приводится математическое обоснование его работоспособности и надёжности (теоремы 1–7), а также демонстрируется его место в обобщённой классификации закладок.

1. Виды клептографических закладок

Клептографическая закладка является частным случаем алгоритмической закладки общего вида и характеризуется тем, что она модифицирует криптоалгоритм, который изначально был спроектирован без алгоритмической закладки.

Инфицированный криптоалгоритм должен обладать, как минимум, следующими свойствами:

— *идентичностью инициализации*: инфицированный и исходный алгоритмы работают на одних тех же входных данных (начальных условиях);
— *структурной идентичностью результата*: выходные данные инфицированного алгоритма имеют ту же структуру, что и у исходного алгоритма;
— *функциональной идентичностью результата*: выходные данные удовлетворяют тем же математическим соотношениям, которым должны удовлетворять выходные данные исходного алгоритма.

Дополнительные (опциональные) свойства инфицированного криптоалгоритма:

— *статистическая идентичность результата*: выходные данные инфицированного алгоритма статистически неотличимы от выходных данных исходного алгоритма;
— *неотличимость среднего времени работы* инфицированного криптоалгоритма от времени работы исходного криптоалгоритма.

В информационной системе с алгоритмической (клептографической) закладкой традиционно рассматривают три основные роли участников [4]: *разработчик*, *пользователь* и *противник* (*злоумышленник*).

В принятой терминологии [17, 18] пользователи могут быть внутренними нарушителями, а противник всегда является внешним нарушителем. То есть роль противника (злоумышленника) тождественна определяемой российской нормативно-правовой документацией роли внешнего нарушителя. Данное утверждение подтверждается и тем, что [18] определяет термины «внешний нарушитель» и «противник» как синонимы.

Далее, в соответствии с рекомендациями [18], будем использовать термин «противник», когда речь идёт о внешнем нарушителе. Под «внутренним нарушителем» будем подразумевать участников системы, выполняющих атаки на криптосистему со встроенной алгоритмической закладкой.

Принимая во внимание тенденции последних лет, ролевую модель можно видоизменить, добавив нового участника — *спецслужбу* — и ограничив уровень знания разработчика (производителя программного обеспечения или аппаратуры со встроенной реализацией криптоалгоритма) [19]: спецслужба, будучи автором закладки и владельцем её ключа, описывает производителю инфицированный алгоритм и, возможно, даёт какие-либо открытые данные, используемые этим алгоритмом. Для того чтобы не возникало смысловой путанницы между разработчиком (автором) алгоритма с закладкой и разработчиком соответствующего программного обеспечения, будем далее называть последнего *производителем*.

Разумеется, подобное расширение ролевой модели имеет смысл, если производитель не является автором закладки, а встраивает её в реализацию по указанию спецслужбы.

В [20] выделяют два вида противников:

— *различительный противник* (*distinguishing adversary*): его цель в том, чтобы отличить честную реализацию от инфицированной; различительная атака может либо выявлять отклонения реализации от эталонной (по времени работы, по статистическим характеристикам выхоных данных и т. п.), либо проверять наличие в реализации конкретного варианта клептографической закладки; в случаях, когда закладка официально встраивается в реализацию спецслужбой и документируется, различительная атака не имеет смысла;
— *противник-криптоаналитик* (*cryptanalyzing adversary*): его цель состоит в том, чтобы «сломать» безопасность данного устройства; это может включать нахождение секретного пользовательского ключа или данных, подделку подписи, вычисление ключа закладки и т. д.

По аналогии можем рассмотривать *различительного нарушителя* и *нарушителя-криптоаналитика*.

Нарушителем может быть практически любой участник системы, за исключением спецслужбы, если её действия по встраиванию закладок являются легальными в рамках информационной системы.

Производитель может рассматриваться как нарушитель, ставящий своей целью вычислить секретный ключ закладки и тем самым получить те же права, что и спецслужба. При этом различительная атака на реализацию со стороны производителя бессмысленна, поскольку он по определению знает о наличии закладки.

Пользователь может рассматриваться как нарушитель, причём его целью может быть как успешная различительная атака — то есть обнаружение факта инфицирования криптосистемы, так и получение ключа закладки для дальнейшего доступа к секретам других пользователей.

Резюмируя, выделим четыре основные роли в информационной системе с алгоритмической закладкой:

— *спецслужба* — владелец ключа закладки: обладает информацией о закладке, владеет ключом к закладке, не владеет секретным ключом пользователя, но может получить к нему полный или частичный доступ, используя ключ закладки;
— *производитель* — разработчик программного обеспечения и/или аппаратуры с реализацией инфицированного алгоритма: обладает информацией о закладке, не владеет ключом к закладке (если он не встроен в реализацию), не владеет секретным ключом пользователя;
— *пользователь*: не обладает (в общем случае) информацией о закладке, не владеет ключом закладки, владеет секретным ключом пользователя;
— *противник*: не обладает (в общем случае) информацией о закладке, не владеет ни ключом закладки, ни секретным ключом пользователя.

Алгоритмическая закладка (в частности, клептографическая), как и классическая программная закладка, реализует недекларируемую возможность: предоставляет доступ к секретным пользовательским данным при условии знания ключа закладки и особенностей её структуры. В зависимости от типа закладки доступ может быть предоставлен либо любому, кто знает о наличии закладки в реализации и о её структуре, либо только тому, кто дополнительно знает ключ закладки.

Таким образом, алгоритмические закладки можно классифицировать по уровню стойкости и по способу реализации недекларированных возможностей. По уровню стойкости закладки делятся на *слабые*, *симметричные* и *асимметричные*.

Слабые закладки являются бесключевыми. Единственный метод их защиты — сокрытие факта встраивания закладки и алгоритма её функционирования. Таким образом, если закладка обнаружена, то любой противник и/или внутренний нарушитель получает тот же доступ к пользовательским данным, что и спецслужба.

Симметричные и *асимметричные закладки* защищены с помощью соответственно симметричных и асимметричных ключей. В случае симметричной закладки ключ, необходимый для доступа к пользовательским данным, встроен в реализацию. Этот ключ участвует в защите скрытого канала или в механизме модификации криптоалгоритма. В случае асимметричной закладки ключ, встроенный в реализацию, не позволяет эффективно вычислить ключ доступа к закладке. Таким образом, асимметричные закладки могут быть использованы в реализациях с открытым исходным кодом — при этом доступ к закладке останется только у владельца ключа закладки (спецслужбы).

По способу реализации недекларированных возможностей можно выделить два вида закладок: *на основе скрытых каналов* и *на основе неявного ослабления криптоалгоритма*.

Скрытый канал — это непредусмотренный разработчиком коммуникационный канал, который может быть применён для нарушения политики безопасности [21]. Недекларированной возможностью в случае алгоритмической закладки на основе скрытого канала является передача автору закладки секретной пользовательской информации. При этом в качестве скрытого канала используется часть легально передаваемых по открытым каналам данных.

Для того чтобы скрытый канал реализовывал работу алгоритмической закладки, используется некоторая обратимая функция E. Пусть $D = E^{-1}$ — обратная функция. В качестве E может быть выбрано как бесключевое обратимое преобразование (в том числе тождественное), так и симметричный или асимметричный шифр. Через скрытый канал передаётся сообщение $m = E(x)$, где x — некоторая секретная информация пользователя.

Если E — бесключевое обратимое преобразование, то построенная алгоритмическая закладка является слабой. При этом $x = D(m)$ может вычислить любой противник или внутренний нарушитель, получивший информацию о структуре закладки (то есть о функциях E и D).

Если функция E является симметричным или асимметричным шифром, то построена соответственно симметричная или асимметричная закладка. В этом случае ключ расшифрования для E позволяет эффективно вычислять $D = E^{-1}$ и является ключом закладки. Владелец ключа закладки может вычислить $x = D(m)$ и получить доступ к секретной пользовательской информации.

В алгоритмических закладках на основе неявного ослабления криптоалгоритма недекларированной возможностью является такая модификация базового алгоритма, которая позволяет вычислить какие-либо пользовательские данные на основе открытых данных. То есть при формальной структурной и функциональной идентичности результата выходные данные инфицированного алгоритма удовлетворяют каким-либо дополнительным соотношениям, на основе которых можно восстановить секретную информацию.

Отметим, что две описанные классификации — по уровню стойкости и по способу реализации недекларированных возможностей — являются независимыми, то есть в результате мы определили шесть видов закладок (см. таблицу в заключении). Далее рассмотрим закладки разных видов на примере генератора ключей RSA.

2. Клептографические закладки в генераторе RSA-ключей

Алгоритм RSA [22] широко известен, и мы не будем останавливаться на его описании. Напомним только вид RSA-ключей. Пусть p и q — большие простые числа, $n = pq$. Выбираются такие числа d и e, называемые соответственно закрытой и открытой экспонентой, что

$$e \cdot d \equiv 1 \pmod{(p-1)(q-1)}.$$

Пара чисел (e, n) является открытым ключом (известна всем участникам системы), а d — закрытым (является секретом пользователя). Очевидно, что для нахождения закрытого ключа d достаточно разложить n на простые множители. В общем случае эта задача является вычислительно трудной.

Рассмотрим такие закладки в генераторе ключей RSA, зависящие в некоторых случаях от секретного значения, называемого «ключом закладки», которые позволяют владельцу этого ключа разложить n на множители p и q за полиномиальное от длины ключа время, тогда как для любого стороннего наблюдателя факторизация n по-прежнему является вычислительно сложной задачей. Для слабых (бесключевых) закладок подразумевается, что для факторизации n достаточно знания о структуре закладки.

Подробный обзор методов построения закладок на основе скрытых каналов в генераторе ключей RSA дан в [23]. В качестве скрытого канала может быть использована открытая экспонента e или часть бит открытого модуля n [6–8].

Самым тривиальным примером бесключевой закладки на основе ослабления криптоалгоритма является использование детерминированного числа p. Детерминированность в данном случае означает, что простой делитель n выбирается без использования генератора псевдослучайных чисел, например может быть заранее задан список простых чисел. Второе простое число q при этом вырабатывается случайным образом.

Число p может быть также функцией от идентификатора ключевого генератора или ещё каких-либо открытых данных. Заметим, что построение таким методом асимметричной закладки невозможно, поскольку для взлома (то есть для факторизации n) требуется ровно та же функция, что и для генерации p.

Более сложным примером бесключевой закладки является закладка на основе уязвимости ROCA, обнаруженной в 2016–2017 гг. в криптобиблиотеке RSALib компании «Infineon» [24, 25]. Исследования [25] показали, что все простые числа, вырабатываемые этой библиотекой, имеют вид

$$p = r_p \cdot M + (65537^{a_p} \bmod M). \tag{1}$$

Числа r_p и a_p выбираются, судя по всему, случайно (по крайней мере, авторы исследования не выявили каких-либо закономерностей), а M задаётся как заранее известное произведение нескольких простых чисел. Авторы [25] провели успешную атаку на ключи RSA, выработанные подобным образом, и смогли взломать RSA-512 и RSA-1024, а также оценили возможность взлома для RSA-2048 и RSA-4096.

Невозможно однозначно утверждать, была ли подобная реализация следствием ошибки разработчиков (как математиков, предложивших алгоритм, так и программистов, реализовавших его) или это было намеренным ослаблением алгоритма для упрощения работы спецслужб. Тем не менее данный метод может рассматриваться в качестве примера слабой закладки.

Примером симметричной закладки на основе неявного ослабления алгоритма является закладка Андерсона [2], в которой генератор вырабатывает простые числа вида

$$p = r_p \cdot D + a_p = r(A_p, D) \cdot D + A_p, \tag{2}$$

где D — секретное 200-битное число, называемое «ключом закладки»; $A_p < \sqrt{D}$ — 100-битные числа; $(A_p, D) = 1$; $r(A, D)$ — функция от двух переменных, возвращающая 56-битное значение.

Как можно видеть, структура закладки Андерсона похожа на ROCA (хотя исторически, конечно, правильнее сказать, что структура (1) похожа на (2)), но взлом для противника, на первый взгляд, затруднён из-за использования ключа закладки. Однако в 1994 г. закладка Андерсона всё-таки была взломана [26]: было продемонстрировано, что противник может вычислить ключ закладки, получив всего 14 различных открытых ключей, выработанных с общим значением D.

Наибольший интерес представляют асимметричные закладки. Методы построения асимметричных закладок без скрытых каналов, впервые предложенные в [16], развивают идею, положенную в основу закладки Андерсона и ROCA. Далее эти методы рассмотрены подробнее и сформулированы теоремы, обосновывающие корректность работы модифицированных генераторов, возможность доступа к закладке для владельца ключей закладки и стойкость к атакам противника и внутреннего нарушителя.

3. Асимметричная закладка в генераторе RSA-ключей: общая идея

Пусть требуется выработать ключ RSA с длиной открытого модуля L бит. Обозначим битовую длину произвольного числа X как $|X|$ при условии, что старший бит X равен 1, то есть $2^{|X|-1} \leqslant X < 2^{|X|}$.

Будем вырабатывать такие простые числа, что $|p| = |q| = L/2$. Пусть D — некоторый параметр лазейки, не обязательно секретный, $|D| = K$.

Для произвольных взаимно простых чисел a и D определим функцию

$$r'(a, D, r_0) = \begin{cases} \min\{r : r \geqslant r_0, (rD + a) \text{ — простое}\}, & \text{если } r < 2^{L/2-K} \\ & \text{и } rD + a < 2^{L/2}, \\ 0 & \text{иначе.} \end{cases} \tag{3}$$

То есть $r'(a, D, r_0)$ — это наименьший (но больший r_0) номер члена арифметической прогресии $rD+a$, являющегося простым числом. Для вычисления значения $r'(a, D, r_0)$ сначала задаётся $r = r_0$, а затем r увеличивается на 1, пока число $rD + a$ не окажется простым.

Так как $(a, D) = 1$, по теореме Дирихле [27] арифметическая прогрессия $rD + a$ содержит бесконечно много простых чисел. При этом теоретически возможна ситуация, когда минимальное простое $rD + a \geqslant 2^{L/2}$ или $r \geqslant 2^{L/2-K}$. В этом случае считаем, что $r'(a, D, r_0) = 0$.

Пусть ID — уникальный идентификатор, определённый для каждого экземпляра генератора, $|ID| = m$; i — счётчик генераций ключей. Выберем некоторую функцию

$$T(x, y, z, i) : \mathbb{Z}_2^K \times \mathbb{Z}_2^K \times \mathbb{Z}_2^m \times \mathbb{N} \to \mathbb{Z}_2^{L/2-K}$$

и положим

$$R(x, y, z, i) = \left\lceil \frac{2^{L/2-1}}{D} \right\rceil + T(x, y, z, i), \tag{4}$$

где $\lceil X \rceil$ — верхняя целая часть, то есть $X \leqslant \lceil X \rceil < X + 1$.

Определим для произвольных взаимно простых чисел a и D

$$r_{ID}^{(i)}(a, D) = r'(a, D, R(a, D, ID, i)). \tag{5}$$

Пусть $\psi_s(\cdot) : \mathbb{Z}_D^* \times \mathbb{Z}_2^m \times \mathbb{N} \to \mathbb{Z}_D^*$ — однонаправленная (односторонняя) по первому аргументу функция с секретом s. Генератор с лазейкой работает следующим образом.

На первом шаге вырабатывается случайное число a_p с условиями $(a_p, D) = 1$, $a_p < D$.

На втором шаге вычисляются значения

$$r_p = r_{ID}^{(i)}(a_p, D), \quad p = r_p D + a_p, \tag{6}$$

где i — монотонно увеличивающийся счётчик генераций ключа.

Если $r_p = 0$, то алгоритм возвращается на первый шаг. Если реализация предусматривает использование фиксированной открытой экспоненты e, то дополнительно надо проверить, что $(e, p - 1) = 1$. При невыполнении этого условия необходимо вернуться на первый шаг.

На третьем шаге вычисляются значения

$$c = \psi_s(a_p, ID, i), \quad a_q = c \cdot a_p^{-1} \pmod{D}. \tag{7}$$

Здесь и далее считаем, что в качестве значения по модулю выбирается наименьший положительный вычет.

На четвёртом шаге вырабатывается случайное число r'_0, $|r'_0| = L/2 - K$, и вычисляются значения

$$r_q = r'(a_q, D, r'_0), \quad q = r_q D + a_q. \tag{8}$$

Если $r_q = 0$ или $(e, q - 1) \neq 1$ (при использовании фиксированной открытой экспоненты), то четвёртый шаг повторяется.

Числа p и q — простые по определению функций (3) и (5). Значения d и e вырабатываются стандартными алгоритмами, $n = pq$ — открытый RSA-модуль.

На основе описанного алгоритма можно сформулировать следующую теорему.

Теорема 1. Пусть требуется выработать ключ RSA. Пусть $ID \in \mathbb{Z}_2^m$ — идентификатор экземпляра генератора; $i \in \mathbb{N}$ — счётчик генераций ключей; $D \in \mathbb{N}$ — некоторое натуральное число, $|D| = K$; функции $r'(a, D, r_0)$ и $r_{ID}^{(i)}(a, D)$ определены формулами (3) и (5) соответственно; $\psi_s(\cdot) : \mathbb{Z}_D^* \times \mathbb{Z}_2^m \times \mathbb{N} \to \mathbb{Z}_D^*$ — односторонняя по первому аргументу функция с секретом s. Тогда для любых случайных $a_p \in \mathbb{Z}_D^*$ и $r'_0 \in \mathbb{Z}$, $(2^{L/2-1})/D < r'_0 < 2^{L/2-K}$, однозначно определены значения

$$\begin{gathered} p = r_{ID}^{(i)}(a_p, D)D + a_p, \\ q = r'\left(\psi_s(a_p, ID, i)a_p^{-1} \pmod{D}, D, r'_0\right) D + \psi_s(a_p, ID, i)a_p^{-1} \pmod{D}. \end{gathered} \tag{9}$$

При этом если $r_{ID}^{(i)}(a_p, D) \neq 0$ и $r'\left(\psi_s(a_p, ID, i)a_p^{-1} \pmod{D}, D, r'_0\right) \neq 0$, то числа p и q являются простыми, $|p| = |q| = L/2$, $|n| = |pq| \in \{L - 1, L\}$ и сложность алгоритма генерации не превышает $O(K^3 + \Psi_{Dsm} + C_D)$ битовых операций, где Ψ_{Dsm} — сложность вычисления функции ψ_s; C_D — сложность вычисления значения функции $r_{ID}^{(i)}(a_p, D)$.

Доказательство. Если $r_{ID}^{(i)}(a_p, D) \neq 0$ и $r'\left(\psi_s(a_p, ID, i)a_p^{-1} \pmod{D}, D, r'_0\right) \neq 0$, то однозначность вычисления и простота чисел p и q в (9) следуют из определения функций $r'(a, D, r_0)$ и $r_{ID}^{(i)}(a, D)$.

По формулам (3)–(5) получаем, что

$$r_{ID}^{(i)}(a,D) = r'(a,D,R(a,D,ID,i)) \geqslant R(a,D,ID,i) \geqslant \frac{2^{L/2-1}}{D},$$

следовательно, $p = r_{ID}^{(i)}(a_p,D)D + a_p > 2^{L/2-1}$.

Поскольку $r_{ID}^{(i)}(a_p,D) \neq 0$, из определения (3) следует, что $p = rD + a < 2^{L/2}$, то есть $|p| = L/2$. Условие $|q| = L/2$ следует из выбора r'_0.

Так как $2^{L/2-1} < p,\ q < 2^{L/2}$, то $2^{L-2} < pq < 2^L$, то есть $|n| = |pq| \in \{L-1, L\}$.

Оценим сложность.

Для вычисления p требуется найти значение $r_{ID}^{(i)}(a_p,D)$, для чего понадобится C_D операций.

Обратный элемент $a_p^{-1} \pmod D$ вычисляется с помощью расширенного алгоритма Евклида. Как показано в [28, гл. 4.5.3, следствие L], необходимое количество шагов деления не превышает $O(K)$; битовая сложность каждого деления не превышает $O(K^2)$. Следовательно, получаем $O(K^3)$ операций.

Далее необходимо вычислить $\psi_s(a_p,ID,i)$ и произведение $\psi_s(a_p,ID,i)a_p^{-1} \pmod D$. Это в сумме потребует $O(K^3 + \Psi_{Dsm})$ операций.

Вычисление функции $r'(a,D,r_0)$ быстрее, чем $r_{ID}^{(i)}(a,D)$, поскольку не требуется находить значение $R(a,D,ID,i)$. Следовательно, на построение q уйдёт не более чем $O(K^3 + \Psi_{Dsm} + C_D)$ операций.

В итоге получаем общую оценку сложности $O(K^3 + \Psi_{Dsm} + C_D)$. ■

Отметим, что если $|n| = L-1$, то алгоритм возвращается на первый шаг и генерация простых чисел выполняется заново.

Рассмотрим, каким образом владелец ключа закладки может получить доступ к закрытому ключу пользователя. Для этого достаточно разложить n на простые множители. В общем случае факторизация является вычислительно сложной задачей, однако внедрённая закладка существенно упрощает вычисления.

В силу выбора p и q и условия (7) имеем $n = pq \equiv a_p a_q \equiv c \pmod D$. Поскольку владелец секрета s может обратить функцию ψ_s, то он может вычислить

$$a_p = \psi_s^{-1}(c,ID,i) = \psi_s^{-1}(n \mod D, ID, i).$$

После этого r_p вычисляется по формуле (5), $p = r_p D + a_p$, $q = n/p$.

Таким образом, доказана

Теорема 2. Пусть s — ключ лазейки, числа p и q выработаны по условиям теоремы 1 для некоторого $D \in \mathbb{N}$, $|D| = K$ и $n = pq$. Тогда

$$p = r_{ID}^{(i)}(\psi_s^{-1}(n \mod D, ID, i), D)D + \psi_s^{-1}(n \bmod D, ID, i).$$

Сложность алгоритма вычисления не превышает $O(K^3 + C_D + \Psi_{Ds^{-1}m})$, где $\Psi_{Ds^{-1}m}$ — сложность вычисления функции ψ_s^{-1} при условии знания s; C_D — сложность вычисления значения $r_{ID}^{(i)}(a_p,D)$.

Итак, знание ключа закладки действительно помогает восстановить закрытый ключ пользователя на основе знания открытого модуля n. Отметим, что если у владельца ключа закладки нет информации о текущем значении счётчика генераций, используемого в формуле (5), но при этом известно, что генератор вырабатывает небольшое количество ключей, то значения этого счётчика можно перебрать: необходимо

вычислять r_p для различных i до тех пор, когда полученное значение $p = r_p D + a_p$ окажется делителем n.

Если при взломе ключа нет возможности синхронизировать счётчик с ключевым генератором (т. е. владелец ключа закладки не может узнать, сколько ключей сгенерировано ранее данным генератором) и при этом генератор вырабатывает достаточно много значений, то можно использовать константное значение i (например, $i = 0$).

Стойкость закладки к атакам противника зависит от стойкости функции ψ_s (но не сводится только к ней).

Стойкость ψ_s (то есть возможность её обратить без знания секрета s) зависит, помимо непосредственно битовой длины s, от битовой длины первого аргумента и возвращаемого значения, то есть от битовой длины D. Выбор слишком маленького D может привести к взлому лазейки противником.

С другой стороны, слишком большие значения D сокращают возможное количество рассматриваемых кандидатов в простые. Например, в предельном случае, если битовый размер D равен размеру генерируемых простых чисел, то вычисление значения $rD + a_p$ может привести к превышению максимальной допустимой длины числа уже при $r = 1$.

Вероятно, для RSA с длиной ключа L бит наиболее оптимальным является размер D порядка $3L/8$, но данная оценка требует уточнения в каждом конкретном случае использования.

Мощность множества возможных пар простых чисел p и q при этом меньше, чем при «честной» генерации случайной пары простых чисел.

Во-первых, значение r_p определяется однозначно для каждого фиксированного a_p — то есть из каждой арифметической прогрессии $rD + a_p$ может быть выбрано ровно одно простое число, все остальные простые заведомо отсеиваются — ключевое множество сокращено. Для частичной компенсации этого эффекта в функцию R добавлена зависимость от идентификатора ключевого генератора (то есть общее множество возможных ключей для всех ключевых генераторов расширено) и от номера генерации (то есть даже при выработке одного и того же a_p на разных шагах итоговые открытые модули не будут иметь общих делителей).

Во-вторых, значение a_q тоже однозначно определяется из значения a_p — то есть не любая пара простых чисел может быть выбрана в качестве p и q. Эта зависимость также отчасти компенсируется параметрами ID и i в функции ψ_s. Более того, случайный выбор r'_0 расширяет ключевое множество.

Даже без учёта счётчика генераций у каждого генератора есть $\varphi(D)$ различных способов выбора a_p. Если число D достаточно большое, то теоретическое снижение криптографической стойкости не обязательно приводит к упрощению взлома ключа противником. Стойкость к взлому противником должна отдельно оцениваться в каждом конкретном случае в зависимости от выбранных параметров закладки.

Построим функцию ψ_s на основе задачи дискретного логарифмирования в произвольной циклической группе. Для простоты описания рассмотрим лазейку без учёта ID и счётика генераций.

Пусть $\mathbb{G} = \langle g \rangle$ — конечная циклическая группа. Задачей дискретного логарифмирования в группе $\mathbb{G}$ называется нахождение для произвольного элемента $a \in \mathbb{G}$ такого $x \in \mathbb{Z}$, что $g^x = a$. Решением задачи Диффи — Хеллмана в группе $\mathbb{G}$ называется нахождение элемента g^{xy} по известным элементам g^x и g^y без знания x и y. Будем рассматривать те группы $\mathbb{G}$, в которых задачи дискретного логарифмирования и Диффи — Хеллмана являются вычислительно сложными.

Пусть задано некоторое отображение $\gamma : \mathbb{G} \to \mathbb{Z}_D^*$ и $\theta_\gamma(a) = \{g_i : \gamma(g_i) = a\}$ — множество прообразов элемента $a \in \mathbb{Z}_D^*$ при отображении γ. Требуется, чтобы мощность $\theta_\gamma(a)$ была небольшой для любого a.

Зададим $S = g^s$ для некоторого секрета s и определим одностороннюю функцию с секретом следующим образом:

$$\psi_s(a) = \gamma(g^{c_0}), \text{ где } a = \gamma(S^{c_0}). \tag{10}$$

Очевидно, что в общем случае вычисление $\psi_s(a)$ является сложной задачей, поскольку определение c_0 требует решения задачи дискретного логарифмирования. Но если изначально задать число a в виде $\gamma(S^{c_0})$, то и $\psi_s(a)$ вычисляется легко. То есть сначала нужно выбрать случайное c_0, а потом уже вычислить a и $\psi_s(a)$.

Поскольку выполнено соотношение

$$a = \gamma(S^{c_0}) = \gamma(g^{sc_0}) = \gamma((g^{c_0})^s), \tag{11}$$

то $(g^{c_0})^s \in \theta_\gamma(a)$. Если γ является гомоморфизмом, то $\gamma((g^{c_0})^s) = (\gamma(g^{c_0}))^s$. Тогда для $c = \psi_s(a)$ выполнено

$$\psi_s^{-1}(c) = \psi_s^{-1}(\psi_s(a)) = a \overset{(11)}{=} (\gamma(g^{c_0}))^s = (\psi_s(a))^s = c^s.$$

В противном случае для $c = \psi_s(a) = \gamma(g^{c_0})$ имеет место $g_i = g^{c_0} \in \theta_\gamma(c)$, то есть

$$\psi_s^{-1}(c) = \psi_s^{-1}(\psi_s(a)) = a = \gamma((g^{c_0})^s) = \gamma(g_i^s), \text{ где } g_i \in \theta_\gamma(c).$$

Таким образом, при знании секрета s функция ψ_s обратима. Одновременно с этим верна

Теорема 3. Пусть $\mathbb{G} = \langle g \rangle$ — конечная циклическая группа, в которой задачи дискретного логарифмирования и Диффи — Хеллмана являются вычислительно сложными; $S = g^s$ для некоторого секрета s и $\psi_s(a) = \gamma(g^{c_0})$, где $a = \gamma(S^{c_0})$ для некоторого случайного c_0. Тогда:

1) для почти всех c (за исключением тривиальных) без знания s вычислительно трудно найти $\psi_s^{-1}(c)$;
2) при условии знания S, c_0, $\psi_s(a)$ вычислительно трудно найти s.

Доказательство. Пусть $c = \gamma(g^{c_0})$. Предположим, что мы можем вычислить $a = \psi_s^{-1}(c) = \gamma(S^{c_0})$ для произвольного c_0.

Рассмотрим задачу Диффи — Хэллмана в группе $\mathbb{G}$. Пусть заданы g^a и g^b, требуется найти g^{ab}. Пусть $s = a$, $c_0 = b$, тогда $S = g^a$, $c = \gamma(g^{c_0}) = \gamma(g^b)$. По предположению, мы можем вычислить

$$a = \psi_s^{-1}(c) = \gamma(S^{c_0}) = \gamma((g^a)^b) = \gamma(g^{ab}).$$

Поскольку по условиям выбора группы $\mathbb{G}$ и отображений γ и $\theta_\gamma(c) = \{g_i : \gamma(g_i) = c\}$ мощность $\theta_\gamma(c)$ является небольшой для любого c, перебор значений $\theta_\gamma(\gamma(g^{ab}))$ является быстрым — это даёт возможность найти g^{ab}.

Таким образом, мы свели решение задачи Диффи — Хеллмана к обращению функции ψ_s, а следовательно, вычисление $\psi_s^{-1}(c)$ без знания s является не менее сложной задачей, чем задача Диффи — Хеллмана в группе $\mathbb{G}$. Поскольку по условию выбора группы $\mathbb{G}$ решение задачи Диффи — Хеллмана является вычислительно трудным, доказано первое утверждение теоремы.

Предположим теперь, что для заданных S, c_0, $\psi_s(a)$ возможно вычислить s. Рассмотрим задачу дискретного логарифмирования по основанию g в группе $\mathbb{G}$. Пусть требуется вычислить x, для которого $X = g^x$.

Зададим $S = X$. Выберем случайное c_0 и вычислим $a = \gamma(S^{c_0})$, $\psi_s(a) = \gamma(g^{c_0})$. По предположению, из этих данных возможно вычислить s, то есть найти такое значение s, что $S = g^s$. Так как $\mathbb{G} = \langle g \rangle$, то $x = s$.

Таким образом, второе утверждение теоремы следует из того, что по условию выбора группы $\mathbb{G}$ задача дискретного логарифмирования в ней является вычислительно сложной. ■

Итак, заданная соотношением (10) функция ψ_s является однонаправленной для противника, то есть её обращение является вычислительно трудной задачей, что не позволяет противнику получить доступ к ключу закладки.

Далее рассмотрим конкретные примеры выбора группы $\mathbb{G}$: мультипликативную группу конечного простого поля и подгруппу группы точек эллиптической кривой.

4. Асимметричные закладки в генераторе RSA-ключей на основе функции дискретного логарифмирования по простому модулю

Пусть D — простое число, g — первообразный корень по модулю D. Выберем ключ лазейки — число s, взаимно простое с $(D-1)$. Открытый ключ $S \equiv g^s \bmod D$ встраивается в реализацию и не является секретом; S является элементом максимального порядка по модулю D.

Вариант 1. Определим функцию ψ_s следующим образом:

$$\psi_s(a) = g^{c_0} \bmod D, \text{ где } a = S^{c_0} \bmod D. \tag{12}$$

Фактически, это частный случай формулы (10) для $\mathbb{G} = \mathbb{Z}_D^*$ при $\gamma(x) = x$, $\theta_\gamma(x) = \{x\}$.

По теореме 3 функция ψ_s является стойкой от атак противника в том смысле, что противник не может за полиномиальное время получить доступ к ключу лазейки или обратить функцию ψ_s на основе данных, передаваемых по открытым каналам в процессе работы алгоритма.

Генератор работает так, как описано в п. 3. Теоремы 1 и 2 можно переформулировать следующим образом:

Теорема 4. Пусть требуется выработать ключ RSA. Пусть $ID \in \mathbb{Z}_2^m$ — идентификатор экземпляра генератора; $i \in \mathbb{N}$ — счётчик генераций ключей; $D \in \mathbb{N}$ — простое число; $|D| = K$; g — первообразный корень по модулю D; функции $r'(a, D, r_0)$ и $r_{ID}^{(i)}(a, D)$ определены формулами (3) и (5) соответственно; $s \in \mathbb{Z}_{D-1}^*$ — ключ закладки, $S = g^s \bmod D$. Тогда для любых случайных $c_0 \in \mathbb{Z}_{D-1}$ и $r_0' \in \mathbb{Z}$, $(2^{L/2-1})/D < r_0' < 2^{L/2-K}$, однозначно определены значения

$$p = r_{ID}^{(i)}(S^{c_0} \bmod D, D)D + (S^{c_0} \bmod D),$$
$$q = r'(g^{c_0} \cdot S^{-c_0} \bmod D, D, r_0')D + (g^{c_0} \cdot S^{-c_0} \bmod D).$$

При этом если $r_{ID}^{(i)}(S^{c_0} \bmod D, D) \neq 0$ и $r'(g^{c_0} \cdot S^{-c_0} \bmod D, D, r_0') \neq 0$, то числа p и q являются простыми, $|p| = |q| = L/2$, $|n| = |pq| \in \{L-1, L\}$ и сложность алгоритма генерации не превышает $O(K^3 + C_D)$ битовых операций, где C_D — сложность вычисления значения функции $r_{ID}^{(i)}(a_p, D)$.

Теорема 5. Пусть s — ключ лазейки, числа p и q выработаны по условиям теоремы 4 для некоторого $D \in \mathbb{N}$, $|D| = K$ и $n = pq$. Тогда

$$p = r_{ID}^{(i)}(n^s \bmod D, D)D + (n^s \bmod D).$$

Сложность алгоритма вычисления не превышает $O(K^3 + C_D)$, где C_D — сложность вычисления значения функции $r_{ID}^{(i)}(a_p, D)$.

Доказательство обеих теорем тривиально, если заметить, что

$$n^s \equiv (g^{c_0})^s \equiv S^{c_0} \equiv p \pmod{D}.$$

При реализации следует отсеивать тривиальные значения c_0: 1, -1, $(D-1)/2$.

Вариант 2. Определим функцию ψ_s следующим образом:

$$\psi_s(a) = g^{c_0} \bmod D, \text{ где } a_0 = S^{c_0} \bmod D,\ a = g^{a_0} \bmod D. \tag{13}$$

Как и в варианте 1, число a сразу вырабатывается в виде (13), т. е сначала выбирается c_0 — случайное целое число из интервала $\{2, \ldots, D-2\}$, $c_0 \neq (D-1)/2$, затем вычисляются a_0 и a, после чего находится $c = \psi_s(a) = g^{c_0} \bmod D$.

Так как выполнено соотношение $a_0 \equiv S^{c_0} \equiv g^{sc_0} \equiv (g^{c_0})^s \equiv c^s \pmod{D}$, то для $c = \psi_s(a)$ имеет место $\psi_s^{-1}(c) = \psi_s^{-1}(\psi_s(a)) = a \equiv g^{a_0} \equiv g^{(c^s \bmod D)} \pmod{D}$. Таким образом, владелец ключа лазейки может эффективно обращать функцию ψ_s.

Итак, после выбора c_0 и вычислений по формуле (13) генератор вычисляет

$$a_p \equiv g^{a_0} \pmod{D},\ b_0 = c_0 - a_0 \pmod{D-1},\ a_q \equiv c \cdot a_p^{-1} \equiv g^{c_0 - a_0} \equiv g^{b_0} \pmod{D}.$$

Далее простые числа p и q вычисляются по формулам (6) и (8).

Владелец ключа лазейки осуществляет доступ к пользовательским ключам по формуле

$$a_p \equiv g^{(n \bmod D)^s \bmod D} \pmod{D}.$$

5. Асимметричная закладка в генераторе RSA-ключей на основе функции дискретного логарифмирования в группе точек эллиптической кривой

Пусть D — большое простое число. Рассмотрим эллиптическую кривую в форме Вейерштрасса

$$\mathbb{E} : y^2 = x^3 + ax + b \bmod D). \tag{14}$$

Пусть P — точка этой эллиптической кривой, имеющая порядок t, где t — большое простое число. Будем использовать кривую, состоящую ровно из t точек. Большая часть рассуждений остаётся верной, если порядок кривой больше t или если t является составным числом, однако мы вводим дополнительные ограничения для повышения стойкости алгоритма.

Для произвольной точки A эллиптической кривой $\mathbb{E}$ будем обозначать x-координату этой точки как $x_G(A)$. Пусть $\mathbb{O}$ — нейтральная (нулевая) точка кривой $\mathbb{E}$, то есть $t \cdot A = \mathbb{O}$ для любой точки A.

Выберем ключ лазейки (число $s < t$) и вычислим точку $S = s \cdot P$.

Определим функцию ψ_s следующим образом:

$$\psi_s(a) = x_G(c_0 \cdot P), \text{ где } a = x_G(c_0 \cdot S). \tag{15}$$

Очевидно, что функция $\psi_s(a)$ определена не для всех a, но для построения лазейки это не имеет значения.

Данный вариант также является частным случаем лазейки, описанной в п. 3, с функцией ψ_s, задаваемой формулой (10), для $\mathbb{G} = \mathbb{E}$ при $\gamma(X) = x_G(X)$, $\theta_\gamma(x) = \{X, -X\}$.

Теоремы 1 и 2 можно переформулировать (конкретизировать для выбранных параметров) следующим образом:

Теорема 6. Пусть требуется выработать ключ RSA. Пусть $ID \in \mathbb{Z}_2^m$ — идентификатор экземпляра генератора; $i \in \mathbb{N}$ — счётчик генераций ключей; $D \in \mathbb{N}$ — простое число; $|D| = K$; P — точка эллиптической кривой (14), имеющая порядок t, где t — большое простое число; функции $r'(a, D, r_0)$ и $r_{ID}^{(i)}(a, D)$ определены формулами (3) и (5) соответственно; $s \in \mathbb{Z}_t^*$ — ключ закладки; $S = sP$.

Тогда для любых случайных $c_0 \in \mathbb{Z}_t^*$ и $r_0' \in \mathbb{Z}$, $(2^{L/2-1})/D < r_0' < 2^{L/2-K}$, однозначно определены следующие значения:

$$p = r_{ID}^{(i)}(x_G(c_0S), D)D + x_G(c_0S),$$
$$q = r'\left(x_G(c_0S)^{-1} \cdot x_G(c_0P) \pmod D, D, r_0'\right) D + (x_G(c_0S)^{-1} \cdot x_G(c_0P) \pmod D).$$

При этом если $r_{ID}^{(i)}(x_G(c_0S), D) \neq 0$ и $r'(x_G(c_0S)^{-1} \cdot x_G(c_0P) \pmod D, D, r_0') \neq 0$, то числа p и q являются простыми, $|p| = |q| = L/2$, $|n| = |pq| \in \{L-1, L\}$ и сложность алгоритма генерации не превышает $O(K^3 + C_D + CP_t)$ битовых операций, где C_D — сложность вычисления значения функции $r_{ID}^{(i)}(a_p, D)$; CP_t — сложность вычисления кратной точки.

Пусть функция $\Theta(x) : \mathbb{Z}_D \to \mathbb{E}$ возвращает точку на кривой $\mathbb{E}$ с заданной x-координатой. Заметим, что точек с заданной x-координатой x_0 либо не существует, либо ровно две (возможно, совпадающие): $Q_1 = (x_0, y_1)$ и $Q_2 = (x_0, y_2)$. При этом $Q_1 = -Q_2$, то есть $y_1 \equiv -y_2 \pmod D$.

Если рассматривать наименьшие положительные вычеты, то либо $y_1 = y_2 = 0$ (т. е. $Q_1 = Q_2$), либо ровно одно из чисел y_1, y_2 принадлежит диапазону $\{1, \ldots, (D-1)/2\}$. То есть можно однозначно определить $\Theta(x)$ следующим образом:

$$\Theta(x_0) = \begin{cases} Q = (x_0, y_0), & 0 \leqslant y_0 \leqslant (D-1)/2, \\ \mathbb{O}, & \text{если такой точки не существует.} \end{cases}$$

Теорема 7. Пусть s — ключ лазейки, числа p и q выработаны по условиям теоремы 6 для некоторого $D \in \mathbb{N}$, $|D| = K$ и $n = pq$. Тогда

$$p = r_{ID}^{(i)}(x_G(s \cdot \Theta(n \bmod D)), D)D + x_G(s \cdot \Theta(n \bmod D)).$$

Сложность алгоритма вычисления не превышает $O(K^3 + C_D + CP_t)$, где C_D — сложность вычисления значения функции $r_{ID}^{(i)}(a_p, D)$; CP_t — сложность вычисления кратной точки.

Доказательства теорем 6 и 7 тривиальны. Подробно вычисления объяснены в [16].

Отметим, что $\theta_\gamma(n \bmod D) = \{\Theta(n \bmod D), -\Theta(n \bmod D)\}$. В общем случае нужно было бы вычислить два кандидата в простые делители: p_0 и p_1, а затем в качестве p выбирать тот из них, который является делителем открытого модуля n. Однако нетрудно заметить, что $s \cdot \Theta(n \bmod D) = -\left(s(-\Theta(n \bmod D))\right)$, а следовательно,

$$x_G(s \cdot \Theta(n \bmod D)) = x_G\left(s(-\Theta(n \bmod D))\right),$$

то есть $p_0 = p_1$. Таким образом, кандидаты в простые делители совпадают.

Стойкость функции ψ_s к атакам противника следует из теоремы 3.

6. Асимметричная закладка, аналогичная ROCA

Рассмотрим ещё один вариант закладки. Его структура обобщает вид (1) простых чисел, генерируемых в библиотеке RSALib, подверженной уязвимости ROCA [25].

Пусть $D = \prod_{i=1}^{k} p_i^{\alpha_i}$, где p_i — небольшие простые. Отметим, что генерация чисел вида (6) и (8) для составного D позволяет исключить из рассмотрения составные числа, делящиеся на все p_i. Например, если D — чётное, то количество кандидатов в простые числа сокращается вдвое.

При этом необходимо, чтобы $(a_p, D) = 1$, иначе все числа вида (6) будут составными. Кроме того, поскольку a_q также должно быть взаимно просто с D, необходимо проверить, что $(c, D) = 1$, где c берётся из формулы (7). Для того чтобы избежать проверок $(a_p, D) = (c, D) = 1$, изменим общий вид простых чисел.

Пусть g — элемент максимального порядка по модулю D. Для поиска такого элемента достаточно выбрать первообразные корни $g_1, \ldots, g_m$ по модулям $p_1^{\alpha_1}, \ldots, p_m^{\alpha_m}$, а затем решить систему

$$\begin{cases} x \equiv g_1 \pmod{p_1^{\alpha_1}}, \\ \ldots \\ x \equiv g_m \pmod{p_m^{\alpha_m}}. \end{cases}$$

Поскольку модули взаимно просты, система имеет решение по китайской теореме об остатках:

$$x \equiv g \pmod{p_1^{\alpha_1} \cdot \ldots \cdot p_m^{\alpha_m}}.$$

Так как порядок каждого элемента g_i максимален по соответствующему модулю p_i, то и порядок g максимален по модулю D и равен $\lambda(D)$, где $\lambda(D) = \text{НОК}(\lambda(p_1^{\alpha_1}), \ldots, \lambda(p_m^{\alpha_m}))$ — функция Кармайкла.

Генератор вырабатывает простые числа вида

$$p = r_p D + g^{a_p}, \quad q = r_q D + g^{a_q}, \tag{16}$$

где $a_p, a_q < D$; r_p, r_q задаются по аналогии с формулами (6) и (8).

Мы рассматривали подобный вид простых в п. 4 во втором варианте лазейки, но для простого числа D.

Внедрение закладки похоже на описанный ранее вариант. Пусть есть односторонняя функция ψ_s с секретом s. Сначала выбирается число a_p, для которого затем вычисляются

$$c = \psi_s(a_p), \quad a_q \equiv c - a_p \pmod{\lambda(D)}, \quad r_p = r_{ID}^{(i)}(g^{a_p}, D), \quad r_q = r'(g^{a_q}, D, r'_0),$$

где функции $r_{ID}^{(i)}$ и r' определены, как и раньше, формулами (3) и (5), r'_0 выбирается случайно с условием $(2^{L/2-1})/D < r'_0 < 2^{L/2-K}$. Если какое-то из значений r_p и r_q оказалось равным 0, то необходимо вернуться на первый шаг алгоритма, к выбору a_p.

Числа p и q, вычисленные по формулам (16), являются простыми по определению функций $r_{ID}^{(i)}$ и r'. При этом

$$n \equiv g^c \pmod{D}.$$

Это условие равносильно системе

$$\begin{cases} n \equiv g_1^c \pmod{p_1^{\alpha_1}}, \\ \ldots \\ n \equiv g_m^c \pmod{p_m^{\alpha_m}}. \end{cases} \tag{17}$$

Поскольку все p_i небольшие, то можно решить каждое из уравнений системы (17), после чего найти c по китайской теореме об остатках (решение гарантированно существует по построению).

Далее владелец ключа s может обратить ψ_s, то есть вычислить $a_p = \psi_s^{-1}(c)$, а затем r_p, p и q. В качестве функции ψ_s можно использовать формулы (12) и (15).

Подобный генератор снижает мощность множества возможных пар простых чисел p и q, поскольку мультипликативная группа вычетов по модулю составного D не является циклической, а значит, не все простые числа могут быть представлены в виде (16) для фиксированных D и g. Тем не менее при больших D ключевое множество остаётся достаточно большим, при этом использование составного D позволяет увеличить скорость работы ключевого генератора.

Нетрудно заметить, что вид чисел (1) в библиотеке RSALib фактически повторяет вид (16). Однако использование числа 65537 вместо элемента g максимального порядка существенно снижает стойкость генератора к атакам противника. Именно благодаря этому генератор оказался подвержен атаке Копперсмита.

Заключение

Классифицированы алгоритмические (в частности, клептографические) закладки по способу реализации недекларированных возможностей и по уровню стойкости, выделены шесть основных классов закладок. Для каждого класса в п. 2 приведены примеры практического построения закладок (таблица); подробно рассмотрен самый значимый класс — асимметричные закладки на основе неявного ослабления алгоритма.

Классы алгоритмических закладок

Классы закладок	Слабые	Симметричные	Асимметричные
На основе скрытых каналов	— Закладка Крепо — Слакмона HSD [8] — Закладка Крепо — Слакмона HSPE [8] — Скрытый канал в старших/младших битах [6, 7] — PAP (Pretty-Awful-Privacy) [29] — и другие...		
	При условии использования бесключевой обратимой функции преобразования сообщения	При условии использования симметричного криптографического преобразования сообщения	При условии использования асимметричного криптографического преобразования сообщения
На основе неявного ослабления алгоритма	— Фиксированное p — ROCA [25]	Закладка Андерсона [2]	Закладки Маркеловой [16]

При рассмотрении закладок, базирующихся на идеях Р. Андерсона, возникает вопрос о количестве простых чисел вида (1), (2), (6), (16). По теореме Дирихле, в арифметической прогрессии содержится бесконечно много простых чисел, но строгие оценки наименьшего простого в арифметической прогрессии (теорема Линника [30]), а также количества простых чисел в арифметической прогрессии на заданном интервале (аналог теоремы о распределении простых чисел [27]) на данный момент слишком неточны, поэтому на их основе сложно получить аналитическую оценку времени работы ключевого генератора с закладкой.

Для экспериментальной оценки возможности практической реализации описанных закладок (в том числе на малоресурсных платформах) была сделана модификация генератора ключей RSA российской смарт-карточной ОС «Вигрид». Численные эксперименты проводились на аппаратном эмуляторе платформы P5CC081. В результате

получено, что генератор с закладкой Андерсона работает в среднем то же время, что и неоптимизированный генератор без закладки. Генератор с уязвимостью ROCA работает в среднем в 2 раза быстрее, чем генератор без закладки. Описанный в п. 6 генератор с асимметричной закладкой работает то же время, что и генератор с уязвимостью ROCA.

Подробности численных экспериментов, а также аналитические подходы к оценке быстродействия генераторов с закладками будут описаны в последующих работах.

Можно сделать вывод, что описанные варианты лазеек эффективны и могут использоваться в малоресурсных устройствах (таких, как смарт-карты, usb-токены, устройства интернета вещей). Кроме того, поскольку эти закладки являются асимметричными, то их можно применять в системах с открытым исходным кодом.

ЛИТЕРАТУРА

1. *Young A. and Yung M.* Kleptography: using cryptography against cryptography // EUROCRYPT'97. LNCS. 1998. V. 1233. P. 62–74.
2. *Anderson R. J.* Practical RSA trapdoor // Electronics Lett. 1993. V. 29. No. 11. P. 995.
3. FBI 'planted backdoor' in OpenBSD. 2010. `https://www.theregister.com/2010/12/15/openbsd\_backdoor\_claim`.
4. *Жуков А. Е.* Криптосистемы со встроенными лазейками // BYTE/Россия. 2007. № 2. С. 45–51.
5. *Bernstein D. J., Lange T., and Niederhagen R.* Dual EC: A standardized back door // The New Codebreakers. LNCS. 2016. V. 9100. P. 256–281.
6. *Desmedt Y.* Abuses in cryptography and how to fight them // LNCS. 1990. V. 403. P. 375–389.
7. *Lenstra A. K.* Generating RSA moduli with a predetermined portion // LNCS. 1998. V. 1514. P. 1–10.
8. *Crépeau C. and Slakmon A.* Simple backdoors for RSA key generation // LNCS. 2003. V. 2612. P. 403–416.
9. *Blaze M.* Protocol failure in the Escrowed Encryption Standard // Proc. CCS'94. 1994. `http://www.mattblaze.org/papers/eesproto.pdf`.
10. Deputy Attorney General Rod J. Rosenstein Delivers Remarks on Encryption at the United States Naval Academ. Annapolis, MD, October 10, 2017. `https://www.justice.gov/opa/speech/deputy-attorney-general-rod-j-rosenstein-delivers-remarks-encryption-united-states-naval`.
11. *Levy I. and Robinson C.* Principles for a More Informed Exceptional Access Debate. 2018. `https://www.lawfareblog.com/principles-more-informed-exceptional-access-debate`.
12. *Barker E. and Kelsey J.* NIST Special Publication 800-90. Recommendation for Random Number Generation Using Deterministic Random Bit Generators. June 2006. `https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-90.pdf`.
13. *Menn J.* Exclusive: Secret contract tied NSA and security industry pioneer. December 21, 2013. `https://www.reuters.com/article/us-usa-security-rsa/exclusive-secret-contract-tied-nsa-and-security-industry-pioneer-idUSBRE9BJ1C220131220`.
14. *Thomson K.* Reflection on trusting trust // Comm. ACM. 1984. V. 27. No. 8. P. 761–763.
15. *Schneier B.* Evaluating the GCHQ Exceptional Access Proposal. January 17, 2019. `https://www.lawfareblog.com/evaluating-gchq-exceptional-access-proposal`.
16. *Markelova A. V.* Embedding asymmetric backdoors into the RSA key generator // J. Computer Virology Hacking Techniques. 2021. No. 17. P. 37–46.

17. Методический документ. Методика определения угроз безопасности информации. Утвержден ФСТЭК России 5 февраля 2021 г.
18. Словарь криптографических терминов / под ред. Б. А. Погорелова, В. Н. Сачкова. М.: Изд-во МЦМНО, 2006.
19. *Жуков А. Е., Маркелова А. В.* Криптография и клептография: скрытые каналы и лазейки в криптоалгоритмах // Информационная безопасность. 2019. № 1. С. 36–41.
20. *Young A. and Yung M.* Malicious Cryptography. Exposing Cryptovirology. Wiley Publ., 2004. 392 p.
21. ГОСТ Р 53113.1-2008. Информационная технология. Защита информационных технологий и автоматизированных систем от угроз информационной безопасности, реализуемых с использованием скрытых каналов. Ч. 1. Общие положения. М.: Стандартинформ, 2018.
22. *Rivest R. L., Shamir A., and Adleman L. M.* A method for obtaining digital signatures and public-key cryptosystems // Comm. ACM. 1978. V. 21. P. 120–126.
23. *Жуков А. Е., Маркелова А. В.* Криптография и клептография. Скрытые каналы и клептографические закладки на их основе в криптосистеме RSA // Защита информации. Инсайд. 2020. № 2(92). С. 58–67.
24. *Švenda P., Nemec M., Sekan P., et al.* The million-key question — investigating the origins of RSA public keys // Proc. 25th USENIX Security'16. USENIX Association, 2016. P. 893–910.
25. *Nemec M., Sys M., Svenda P., et al.* The return of Coppersmith's attack: Practical factorization of widely used RSA moduli // Proc. CCS'17. ACM, 2017. P. 1631–1648.
26. *Kaliski B. S.* Anderson's RSA trapdoor can be broken // Electronics Lett. 1993. V. 29. No. 15. P. 1387.
27. *Дэвенпорт Г.* Мультипликативная теория чисел. М.: Наука, 1971. 200 с.
28. *Кнут Д.* Искусство программированя. Т. 2. Получисленные алгоритмы. 3-е изд. М.: Вильямс, 2001. 832 с.
29. *Young A. and Yung M.* The dark side of black-box cryptography // CRYPTO'96. LNCS. 1997. V. 1109. P. 89–103.
30. *Linnik Yu. V.* On the least prime in an arithmetic progression I. The basic theorem // Матем. сб. 1944. Т. 15(57). № 2. С. 139–178.

REFERENCES

1. *Young A. and Yung M.* Kleptography: using cryptography against cryptography. EUROCRYPT'97, LNCS, 1998, vol. 1233, pp. 62–74.
2. *Anderson R. J.* Practical RSA trapdoor. Electronics Lett., 1993, vol. 29, no. 11, pp. 995.
3. FBI 'planted backdoor' in OpenBSD. 2010. `https://www.theregister.com/2010/12/15/openbsd\_backdoor\_claim`.
4. *Zhukov A. E.* Kriptosistemy so vstroennymi lazeykami [Cryptosystems with built-in trapdoors]. BYTE/Russia, 2007, no. 2, pp. 45-51. (in Russian)
5. *Bernstein D. J., Lange T., and Niederhagen R.* Dual EC: A standardized back door. The New Codebreakers, LNCS, 2016, vol. 9100, pp. 256–281.
6. *Desmedt Y.* Abuses in cryptography and how to fight them. LNCS, 1990, vol. 403, pp. 375–389.
7. *Lenstra A. K.* Generating RSA moduli with a predetermined portion. LNCS, 1998, vol. 1514, pp. 1–10.
8. *Crépeau C. and Slakmon A.* Simple backdoors for RSA key generation. LNCS, 2003, vol. 2612, pp. 403–416.
9. *Blaze M.* Protocol failure in the Escrowed Encryption Standard. Proc. CCS'94, 1994, `http://www.mattblaze.org/papers/eesproto.pdf`.

10. Deputy Attorney General Rod J. Rosenstein Delivers Remarks on Encryption at the United States Naval Academ. Annapolis, MD, October 10, 2017, https://www.justice.gov/opa/speech/deputy-attorney-general-rod-j-rosenstein-delivers-remarks-encryption-united-states-naval.
11. *Levy I. and Robinson C.* Principles for a More Informed Exceptional Access Debate. 2018, https://www.lawfareblog.com/principles-more-informed-exceptional-access-debate.
12. *Barker E. and Kelsey J.* NIST Special Publication 800-90. Recommendation for Random Number Generation Using Deterministic Random Bit Generators. June 2006, https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-90.pdf.
13. *Menn J.* Exclusive: Secret contract tied NSA and security industry pioneer. December 21, 2013, https://www.reuters.com/article/us-usa-security-rsa/exclusive-secret-contract-tied-nsa-and-security-industry-pioneer-idUSBRE9BJ1C220131220.
14. *Thomson K.* Reflection on trusting trust. Comm. ACM, 1984, vol. 27, no. 8, pp. 761–763.
15. *Schneier B.* Evaluating the GCHQ Exceptional Access Proposal. January 17, 2019, https://www.lawfareblog.com/evaluating-gchq-exceptional-access-proposal.
16. *Markelova A. V.* Embedding asymmetric backdoors into the RSA key generator. J. Computer Virology Hacking Techniques, 2021, no. 17, pp. 37–46.
17. Metodicheskiy dokument. Metodika opredeleniya ugroz bezopasnosti informatsii [Methodical Document. Methodology for Determining Threats to Information Security]. Approved by the FSTEC of Russia on February 5, 2021. (in Russian)
18. Slovar' kriptograficheskikh terminov [Dictionary of Cryptographic Terms]. B. A. Pogorelov and V. N. Sachkov (eds.), Moscow, MCCME Publ., 2006. (in Russian)
19. *Zhukov A. E. and Markelova A. V.* Kriptografiya i kleptografiya: skrytye kanaly i lazeyki v kriptoalgoritmakh [Cryptography and kleptography: hidden channels and trapdoors in cryptoalgorithms]. Informatsionnaya Bezopasnost', 2019, no. 1, pp. 36–41. (in Russian)
20. *Young A. and Yung M.* Malicious Cryptography. Exposing Cryptovirology. Wiley Publ., 2004, 392 p.
21. GOST R 53113.1-2008. Informatsionnaya tekhnologiya. Zashchita informatsionnykh tekhnologiy i avtomatizirovannykh sistem ot ugroz informatsionnoy bezopasnosti, realizuemykh s ispol'zovaniem skrytykh kanalov. Ch. 1. Obshchie polozheniya [GOST R 53113.1-2008. Information Technology. Protection of Information Technologies and Automated Systems from Information Security Threats Implemented using Covert Channels. P. 1. General Principles]. Moscow, Standartinform Publ., 2018. (in Russian)
22. *Rivest R. L., Shamir A., and Adleman L. M.* A method for obtaining digital signatures and public-key cryptosystems. Comm. ACM, 1978, vol. 21, pp. 120–126.
23. *Zhukov A. E. and Markelova A. V.* Kriptografiya i kleptografiya. Skrytye kanaly i kleptograficheskie zakladki na ikh osnove v kriptosisteme RSA [Cryptography and kleptography. Covert channels and kleptographic backdoors based on them in the RSA cryptosystem]. Zaŝita Informacii. Inside, 2020, no. 2(92), pp. 58–67. (in Russian)
24. *Švenda P., Nemec M., Sekan P., et al.* The million-key question — investigating the origins of RSA public keys. Proc. 25th USENIX Security'16, USENIX Association, 2016, pp. 893–910.
25. *Nemec M., Sys M., Svenda P., et al.* The return of Coppersmith's attack: Practical factorization of widely used RSA moduli. Proc. CCS'17, ACM, 2017, pp. 1631–1648.
26. *Kaliski B. S.* Anderson's RSA trapdoor can be broken. Electronics Lett., 1993, vol. 29, no. 15, pp. 1387.
27. *Davenport H.* Mul'tiplikativnaya teoriya chisel [Multiplicative Number Theory]. Moscow, Nauka, 1971. 200 p. (in Russian)

28. *Knuth D.* The Art of Computer Programming, vol. 2: Seminumerical Algorithms. 3rd ed. Addison-Wesley, 1998.
29. *Young A. and Yung M.* The dark side of black-box cryptography. CRYPTO'96, LNCS, 1997, vol. 1109, pp. 89–103.
30. *Linnik Yu. V.* On the least prime in an arithmetic progression I. The basic theorem. Rec. Math. (Mat. Sbornik) N.S., 1944, vol. 15(57), no. 2, pp. 139–178.