МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

УДК 519.7 DOI 10.17223/20710410/54/3

ВЛИЯНИЕ РАНДОМИЗАЦИИ В МЕХАНИЗМАХ VKO НА БЕЗОПАСНОСТЬ СРЕДСТВ ЗАЩИТЫ ИНФОРМАЦИИ

Е. К. Алексеев, В. Д. Николаев, С. В. Смышляев

ООО «КРИПТО-ПРО», г. Москва, Россия

E-mail: alekseev@cryptopro.ru, nikolaev@cryptopro.ru, svs@cryptopro.ru

Одним из широко применяемых на практике при работе в условиях слабодоверенного окружения механизмов противодействия атакам на используемые в процедурах выработки общих секретов долговременные ключи является умножение на рандомизирующие множители с последующим применением хэш-функций. Данный подход применяется в механизмах семейства VKO, на основе которых строятся российские криптонаборы основных протоколов криптографической защиты информации (в том числе IPsec, TLS, CMS), стандартизированных в Российской Федерации. В частности, таким образом устроена выработка общих параметров в российских механизмах протокола TLS 1.2, повсеместно применяемого в массовых программных средствах защиты информации. В работе рассмотрены некоторые аспекты результирующей безопасности процедур выработки общих параметров в случае ошибок реализации, из-за которых возможны сбои при вычислениях в группах точек скрученных кривых Эдвардса составного порядка, а также в случае отсутствия гарантий константного времени вычисления кратных точек.

Ключевые слова: *модели и методы защиты информации, криптографические протоколы.*

IMPACT OF RANDOMIZATION IN VKO MECHANISMS ON OVERALL SECURITY LEVEL

E. K. Alekseev, V. D. Nikolaev, S. V. Smyshlyaev

CryptoPro, Moscow, Russia

Multiplier randomization techniques with hashing of the results is one of widely used (especially for semi-trusted environment) countermeasures against attacks on key agreement protocols in practice. This approach is used, for instance, in VKO mechanisms, which are used as building blocks for Russian cipher suites for main cryptographic protocols (including IPsec, TLS, CMS), standardized in Russia. As an important example, shared keys are produced with this technique in TLS 1.2 cipher suites, which are widespread in cryptographic software for citizens of Russia. In this paper, we consider overall security of procedures of shared key computation in the practically significant cases of implementation errors in computations on twisted Edwards elliptic curves and non-constant time of scalar multiplication operations.

Keywords: *models and methods in information security, cryptographic protocols.*

Введение

В рамках решения важной для систем защиты информации проблемы обеспечения информационной безопасности в случае неполного доверия к окружению применяются механизмы, компенсирующие потенциальные уязвимости среды. В частности, для повышения безопасности программных средств криптографической защиты в условиях аппаратного окружения, не проходящего исследования в части побочных каналов, применяется приём умножения закрытых величин на рандомизирующие множители — в этом случае многократное использование одного и того же закрытого значения не будет приводить к одним и тем же вычислениям, а некоторые методы анализа с использованием побочных каналов не будут применимы из-за невозможности получения выборки достаточного объёма. Одними из стандартизированных в России механизмов, в которых данный приём учтён непосредственно в математической конструкции, являются определённые в рекомендациях по стандартизации Р 50.1.113-2016 «Информационная технология. Криптографическая защита информации. Криптографические алгоритмы, сопутствующие применению алгоритмов электронной цифровой подписи и функции хэширования» механизмы выработки ключей обмена VKO.

В настоящей работе исследуются свойства данных механизмов, которые при наличии ошибок в программной реализации могут привести к падению уровня защищённости средств. В частности, изучаются вопросы построения методов криптографического анализа, применимых к реализациям VKO в случае таких недостатков реализации, как ошибки при вычислениях в группах точек скрученных кривых Эдвардса составного порядка (стандартизированных в России) и зависимость времени вычислений от закрытых величин.

Показано, что положительные свойства данных механизмов, крайне желательные для использования в предназначенных для работы в слабодоверенном окружении средствах защиты информации, неразрывно связаны с существенным повышением критичности последствий уязвимостей при реализации для результирующей безопасности. Рандомизирующие множители позволяют «размывать» влияние закрытых величин на значения, которые противник может перехватывать по побочным каналам, предотвращая атаки, требующие накопления существенной статистики для уменьшения перебора закрытых величин. Однако одновременно с этим существенно опаснее становятся методы атак, предполагающие получение информации о закрытых величинах по одному значению, без необходимости накопления. В этих случаях умножение на рандомизирующие множители может приводить к тому, что малозначимые на практике атаки, позволявшие сокращать перебор долговременного секрета всего лишь в 2–4 раза, могут превращаться в применимые на практике для полного восстановления долговременных секретов. Таким образом, обратной стороной методов обеспечения безопасности в слабодоверенном окружении может становиться абсолютная недопустимость ошибок реализации программной части, защитные механизмы при ошибках разработки превращаются в оружие против самих защищаемых объектов.

В п. 1 приводится описание механизмов VKO и порядок применения в них рандомизирующего множителя UKM, а также типичные схемы применения данных механизмов в криптографических протоколах. Пункт 2 посвящён подходам к эксплуатации ошибок при работе на эллиптических кривых составного порядка с целью восстановления долговременного секрета и влиянию рандомизации на последствия таких ошибок для безопасности. В п. 3 рассмотрены некоторые подходы к восстановлению информации о долговременном секрете с помощью побочного канала по времени и их развитие в случае применения рандомизирующих множителей.

В работе использованы следующие обозначения:

— p — характеристика конечного поля, над которым задана кривая;
— m — порядок группы точек эллиптической кривой;
— q — простое число, порядок простой подгруппы группы точек эллиптической кривой;
— P — базовая точка простой подгруппы группы точек эллиптической кривой;
— O — нейтральный элемент группы точек эллиптической кривой;
— $\oplus$ — операция сложения двух точек эллиптической кривой;
— $[a]P$ — операция скалярного умножения точки на число (точка P умножается на целое a).

1. Применение рандомизации при вычислении общего секрета в протоколах

1.1. Механизмы VKO

Вычисление общего ключа по схеме Диффи — Хеллмана [1] является одним из основных базовых механизмов криптографии с открытым ключом. В современных протоколах данная процедура применяется в группе точек эллиптической кривой (как правило, эллиптическая кривая задаётся над простым конечным полем $\mathbb{F}_p$): для вычисления общего секрета закрытый ключ x одного из участников протокола умножается на открытый ключ $Y = [y]P$ второго участника: элемент группы Y умножается на скалярное значение x. Результатом операции скалярного умножения точки кривой является также точка на эллиптической кривой — в общем случае представимая вовсе не битовой строкой, претендующей на неотличимость от равновероятно выбранной из множества строк той же длины. Следовательно, для использования результата операции в качестве криптографического параметра (например, симметричного ключа блочного шифра) над ним требуется произвести некоторое преобразование. В соответствии со стандартной практикой, отражённой, например, в рекомендациях [2], таким преобразованием является хэширование. Отметим, что в некоторых случаях, например в протоколе TLS 1.3 [3], оно применяется не непосредственно к результату процедуры Диффи — Хеллмана, а сразу к набору данных (в этом случае общий ключевой материал используется в HMAC для получения основных ключей безопасности).

Схема выработки общего ключа VKO является в некотором смысле расширением схемы Диффи — Хеллмана. Помимо точки эллиптической кривой Y, соответствующей открытому ключу одного из участников, и закрытого ключа x другого участника, схема получает на вход ненулевое число UKM. Итоговое общее ключевое значение, вырабатываемое схемой, рассчитывается по формуле

$$\mathrm{VKO}(x, Y, UKM) = \mathrm{HASH}\left(\left[\frac{m}{q}(x \cdot UKM \bmod q)\right] Y\right),$$

где m, q — порядок группы точек и порядок простой подгруппы точек используемой эллиптической кривой; $HASH$ — хэш-функция ГОСТ Р 34.11-2012 [4]. Первым документом в области стандартизации, определяющим механизм семейства VKO, является RFC 4357 [5]. В 2016 г. рабочей группой по сопутствующим криптографическим алгоритмам, определяющим ключевые системы, входящей в состав Технического комитета 26 «Криптографическая защита информации», были разработаны рекомендации [6], которые описывают механизм VKO для ключей ГОСТ Р 34.10-2012 [7]. Соответствующий документ в системе IETF — RFC 7836 [8] — появился также в 2016 г.

Необходимо отметить, что процедуры выработки общего ключа по схемам Диффи — Хеллмана и VKO могут применяться в протоколах к различным с точки зрения срока жизни ключевым парам участников:

1. Оба ключа участников одноразовые («эфемерные», «ephemeral»). Так, например, в протоколе TLS 1.3 одноразовые ключевые пары вырабатываются и клиентом, и сервером в начале соединения и используются однократно для выработки общего секрета по схеме Диффи — Хеллмана. Аутентификация клиента и сервера производится при помощи механизма электронной подписи, использующего отдельные ключевые пары.
2. Один из ключей эфемерный, а другой долговременный («ephemeral – static»). Такая схема используется, например, при выработке ключа шифрования ключа в сообщениях формата CMS, а также в протоколе TLS 1.2. В первом случае отправитель генерирует эфемерную ключевую пару, использует её в схеме VKO, а затем вместе с сообщением передаёт свой эфемерный открытый ключ. Получатель использует для выполнения схемы VKO полученный открытый ключ и свой долговременный закрытый ключ. В случае протокола TLS 1.2 эфемерная ключевая пара используется клиентом, а сервер применяет долговременную пару. Открытый эфемерный ключ клиента передаётся серверу в сообщении ClientKeyExchange. Аутентификация сервера при этом производится по долговременному ключу VKO (сервер доказывает факт владения соответствующим закрытым ключом, выработав ключевое хэш-значение с использованием выработанного общего ключа), а аутентификация клиента — средствами электронной подписи с применением отдельной ключевой пары.
3. Оба ключа участников долговременные («static – static»). Такая схема может применяться в устаревших криптонаборах протокола TLS 1.0, TLS 1.1 и TLS 1.2 (например, TLS_GOSTR341001_WITH_28147_CNT_IMIT, устаревшая версия набора TLS_GOSTR341112_256_WITH_28147_CNT_IMIT) при установлении соединения с двусторонней аутентификацией. Аутентификация и клиента, и сервера производится по долговременным ключам VKO; механизм электронной подписи при этом не применяется. Это позволяет не использовать при анализе стойкости протокола подписывающий оракул.

Использование нетривиального значения UKM обязательно для случая «static – static» (в противном случае при повторных операциях VKO будут производиться одинаковые ключи). Хэширование в этом случае помогает избежать угадывания ключей согласования разных сессий. Предположим, что схема VKO не применяла бы хэширование. В этом случае компрометация одного общего сеансового ключа могла бы привести к компрометации всех общих ключей, полученных на двух данных долговременных ключевых парах. Действительно, пусть был скомпрометирован ключ $K_1 = \left[\frac{m}{q}((x \cdot UKM_1) \bmod q)\right] Y$ и при выработке общего ключа K_2 использовалось новое значение UKM_2. Тогда противник легко восстанавливает ключ K_2:

$$K_2 = \left[\frac{UKM_2}{UKM_1} \bmod q\right] K_1.$$

1.2. Типичные способы применения VKO в протоколах

Прикладные протоколы используют механизмы VKO для установления общего секретного ключа сторон взаимодействия. Этот ключ может использоваться как непо-

средственно для создания основного ключевого материала, так и как ключ шифрования мастер-ключей. Рассмотрим эти сценарии:

1. Протокол TLS 1.2 [9]. Механизм VKO в этом случае используется с одной эфемерной парой ключей (со стороны клиента) и одной долговременной парой ключей (со стороны сервера). В качестве значения UKM применяется число, задаваемое в формате BigEndian первыми 16 байтами значения HASH(ClientRandom|ServerRandom), где HASH — хэш-функция ГОСТ Р 34.11-2012 с длиной выхода 256 бит. Полученный ключ используется для шифрования премастер-ключа, вырабатываемого клиентом и передаваемого в зашифрованном виде серверу. Пассивный противник, наблюдающий за каналом, может получить случайные значения ClientRandom, ServerRandom в открытом виде и, таким образом, получить значение UKM. Активный противник, действующий от имени клиента, может в некоторой степени управлять значениями UKM, выбирая разные значения ClientRandom по своему усмотрению, однако не имеет возможности навязать конкретное значение. Следует отдельно подчеркнуть, что вместо схемы VKO в новой версии протокола TLS 1.3 используется непосредственно схема Диффи — Хеллмана.
2. Сообщения формата CMS [10]. Механизм VKO используется для выработки ключа шифрования ключа данных в сообщениях типа EnvelopedData. Может поддерживаться один из двух режимов работы: со статическим VKO (и отправитель, и получатель используют долговременные ключи) и «полустатическим» VKO (ключ отправителя — эфемерный, получателя — долговременный). UKM при этом указывается в сообщении в открытом виде и может быть известен пассивному противнику. Активный противник, отправляющий сообщения, может вырабатывать UKM полностью по своему усмотрению.
3. Протоколы IKEv1 и IKEv2 [11, 12]). Механизм VKO используется для выработки общего ключа сторон, из которого потом выращивается базовый ключ SKEYSEED. И инициатор соединения, и ответчик применяют эфемерные ключи. В качестве UKM при этом используется константное значение 1, поэтому активный противник не может навязывать значения UKM.

2. Влияние рандомизации в случае ошибок проверки входной точки

2.1. Особенности работы схем выработки общего ключа на эллиптических кривых составного порядка

При использовании эллиптических кривых, аддитивные группы которых имеют составной порядок (например, стандартизированные в России скрученные кривые Эдвардса id-tc26-gost-3410-2012-256-paramSetA и id-tc26-gost-3410-2012-512-paramSetC [13, 14], кофактор m/q равен 4), принципиально важным для безопасности закрытых ключей является противодействие методам проведения атак с помощью навязывания точек малых порядков. Подробная информация о данных методах криптоанализа представлена в [15, 16]; напомним основные идеи.

В модели, в которой противник имеет возможность передать свой открытый ключ Y атакуемому серверу с закрытым ключом x для вычисления ключа согласования SK по формуле $SK = [x]Y$, этот противник может в качестве Y передать не точку основной подгруппы кривой (порядка q), а точку из малой подгруппы (например, порядка m/q). Если сервер при этом не проверит принадлежность Y основной подгруппе, вычислит SK (который будет иметь одно из m/q возможных значений) и передаст данные, зависящие от SK (например, полученный с использованием SK

шифртекст), то противник, перебрав и сравнив с вычисленным на сервере SK значения $[1]Y, [2]Y, \ldots, [(m/q-1)]Y$, сможет получить значение $x \bmod m/q$, т. е. получить $\log_2(m/q)$ бит о ключе x.

2.2. Расширение атак с использованием вспомогательных кривых

Атака, описанная выше, может быть модифицирована для случая кривых с кофактором, равным 1. Предположим, что сервер не только не проверяет принадлежность получаемой точки группе нужного порядка, но и принадлежность целевой эллиптической кривой вообще. Противник может воспользоваться тем фактом, что большинство реально используемых в прикладной криптографии эллиптических кривых в краткой форме Вейерштрасса задаются уравнением вида $y^2 = x^3 - 3x + b$, $b \in \mathbb{F}_p$. В этом случае противник может, перебирая значения b, находить произвольные кривые, порядки групп точек которых имеют малые делители. Поскольку некоторые формулы скалярного умножения точек эллиптической кривой в краткой форме Вейерштрасса не зависят от коэффициента b, сервер при выполнении операции умножения не получит результат в виде точки, не лежащей на целевой кривой. Противник в этом случае может последовательно высылать точки Q_2 порядка 2, Q_3 порядка 3, Q_5 порядка 5 с разных кривых, тем самым получая значения $x \bmod 2$, $x \bmod 3$, $x \bmod 5$ и так далее, что позволит определить многие биты закрытого ключа x. Атака такого рода впервые описана в [17].

2.3. Расширение атак в случае применения рандомизации

Атака для кривых составного порядка может быть расширена, если противник может управлять значениями UKM, а также получать информацию об успехе или неуспехе операции, проводимой с помощью общего ключа.

Предположим, что противник передаёт точку эллиптической кривой $Y = Q \oplus T$, где Q — точка, принадлежащая подгруппе порядка q (Q может быть равно O), а T — точка малого порядка (в случае российских стандартизированных скрученных кривых в форме Эдвардса это может быть точка T_2 порядка 2 или точка T_4 порядка 4). Если операция VKO на стороне получателя реализована правильным образом, а именно по формуле $\left[\frac{m}{q}(UKM \cdot x \bmod q)\right] Y$ (где x — долговременный закрытый ключ сервера), то будет получена результирующая точка

$$\left[\frac{m}{q}(UKM \cdot x \bmod q)\right] Q \oplus \left[\frac{m}{q}(UKM \cdot x \bmod q)\right] T = \left[\frac{m}{q}(UKM \cdot x \bmod q)\right] Q,$$

принадлежащая целевой подгруппе порядка q группы точек эллиптической кривой. В случае неправильной реализации, заключающейся в умножении на кофактор $\frac{m}{q}$ по модулю q, в ходе операции VKO будет получен следующий результат:

$$\left[\left(\left(\frac{m}{q} \cdot UKM \cdot x\right) \bmod q\right)\right] Q \oplus \left[\left(\left(\frac{m}{q} \cdot UKM \cdot x\right) \bmod q\right)\right] T = \left[\left(\left(\frac{m}{q} \cdot UKM \cdot x\right) \bmod q\right)\right] Q \oplus U,$$

где U — точка группы малого порядка, равная $[((m/q) \cdot UKM \cdot x) \bmod q]T$. Такой ход выполнения операции VKO позволит противнику проводить следующую атаку, направленную на полное определение закрытого ключа сервера x:

1. Противник вырабатывает открытый ключ Y, который не принадлежит целевой подгруппе порядка q. Сделать это он может, выработав ключевую пару $(y, [y]P)$, где $y \in \{1, \dots q-1\}$ — закрытый ключ, а далее положив $Y = [y]P \oplus T$, где T — точка малого порядка.
2. Противник передаёт Y получателю в рамках используемого протокольного решения для проведения операции VKO.
3. Противник получает значение общего ключа $K = [((m/q) \cdot UKM \cdot x) \bmod q]Q \oplus \oplus [((m/q) \cdot UKM \cdot x) \bmod q]T$.
4. Противник ожидает уведомления о корректной выработке общего ключа получателем.

   а) В случае протокола TLS версий 1.0–1.2 данное уведомление может быть получено следующим образом. В рамках сообщения ClientKeyExchange противник пересылает свой открытый ключ Y, а также зашифрованный на ключе K премастер-ключ PMS. Противник вырабатывает из премастер-ключа мастер-ключ MS, после чего вырабатывает значение ClientFinished, равное выходу ключевой хэш-функции, ключом которой является MS. Отметим, что данное сообщение пересылается в защищённом виде (будучи зашифрованным и совместно со значением имитовставки). Ключи шифрования и имитозащиты для данного сообщения также вырабатываются на основе ключа MS. В свою очередь, сервер, получив сообщение ClientKeyExchange и, следовательно, точку Y, может выполнить операцию VKO, получить ключ K, расшифровать зашифрованный ключ PMS, получить ключ MS, расшифровать и проверить целостность сообщения ClientFinished и проверить его. Если хотя бы одна из этих операций завершается с ошибкой, сервер пересылает клиенту уведомление об ошибке (alert) и прерывает исполнение протокола. В противном случае сервер формирует сообщение ServerFinished и успешно завершает исполнение протокола. Таким образом, у противника получается чёткий критерий выработки общего ключа K сервером.

   б) В случае использования зашифрованных сообщений формата CMS противник вырабатывает случайное значение ключа KEK, используемого для шифрования и имитозащиты данных, и шифрует его на общем ключе K. Получатель расшифровывает ключ KEK, а затем расшифровывает и проверяет целостность данных. Если хотя бы одна из этих операций завершается с ошибкой, получатель отвергает сообщение, иначе сообщение принимается. Если получатель сигнализирует отправителю о факте ошибки или успеха, противник получает чёткий критерий выработки общего ключа K получателем.
5. Противник, зная общий ключ и удостоверившись в его знании получателем, определяет значение

$$U = K \oplus \left[-1 \cdot \left(\frac{m}{q} \cdot UKM \cdot x\right) \bmod q\right] Q = \left[\frac{m}{q} \cdot UKM \cdot x \bmod \frac{m}{q}\right] T.$$

   Так как точка U имеет малый порядок, он перебором определяет l, такое, что $U = [l]T$, при этом $l = ((m/q) \cdot UKM \cdot x) \bmod q$. Для российских стандартизированных скрученных кривых в формате Эдвардса $m/q = 4$, это означает получение информации о двух битах закрытого ключа сервера. Отметим, что в корректной реализации протокола VKO (даже если реализация сервера не осуществляет проверку принадлежности точки Y целевой подгруппе) атака бу-

дет прервана именно на этом пункте — противник получит $U = O$, что не даст ему никакой информации о ключе x.

6. При проведении атаки на схему VKO в рамках конкретного прикладного протокола противник может повторять шаги 2.3–2.3 N раз (где N определяется битовым размером задачи и вычислительными возможностями противника), каждый раз используя новое значение UKM. Для сообщений CMS он имеет возможность полностью управлять значением UKM, в случае протокола TLS — рандомизировать (конкретное значение UKM будет зависеть ещё и от случайного значения, выбираемого сервером), отбрасывая неподходящие для проведения атаки значения (например, случайно совпавшие с прошлыми). Получаемые значения $a_i = ((m/q) \cdot UKM_i \cdot x \bmod q) \bmod (m/q)$, $1 \leqslant i \leqslant N$, для случая $m/q = 4$ будут иметь два содержательных бита, которые обозначим $a_{i,0}, a_{i,1}$. Информация, полученная противником, может быть представлена в виде системы двух булевых уравнений

$$\begin{cases} a_{i,0} = f_0(q, x_1, \ldots, x_n, UKM_{i,1}, \ldots, UKM_{i,n}), \\ a_{i,1} = f_1(q, x_1, \ldots, x_n, UKM_{i,1}, \ldots, UKM_{i,n}). \end{cases}$$

Здесь через n обозначена битовая длина закрытого ключа; $x_1, \ldots, x_n$ — переменные, соответствующие битам закрытого ключа; $UKM_{i,j}$ — константные биты, составляющие представление чисел UKM_i, $1 \leqslant i \leqslant N$, $1 \leqslant j \leqslant n$. Функции f_i, $i = 1, 2$, являются булевыми многочленами, определяемыми операцией приведения по модулю q.

7. Искомое значение закрытого ключа можно определить из полученной системы N уравнений с единственным неизвестным x вида

$$\begin{cases} \left(\dfrac{m}{q} \cdot UKM_1 \cdot x \bmod q\right) \bmod \dfrac{m}{q} = a_1, \\ \ldots \\ \left(\dfrac{m}{q} \cdot UKM_N \cdot x \bmod q\right) \bmod \dfrac{m}{q} = a_N \end{cases} \tag{1}$$

либо из представления этой системы в виде $2N$ булевых уравнений с n неизвестными $x_1, \ldots, x_n$:

$$\begin{cases} a_{1,0} = f_0(q, x_1, \ldots, x_n, UKM_{1,1}, \ldots, UKM_{1,n}), \\ a_{1,1} = f_1(q, x_1, \ldots, x_n, UKM_{1,1}, \ldots, UKM_{1,n}), \\ \ldots \\ a_{N,0} = f_0(q, x_1, \ldots, x_n, UKM_{N,1}, \ldots, UKM_{N,n}), \\ a_{N,1} = f_1(q, x_1, \ldots, x_n, UKM_{N,1}, \ldots, UKM_{N,n}). \end{cases}$$

Замечание 1. Системы такого вида могут решаться в общем виде с использованием SAT-решателей и алгоритмов, основанных на построении базисов Грёбнера. Такой подход, однако, при первом рассмотрении представляется неэффективным, так как трудоёмкость этих алгоритмов, по-видимому, будет значительно превышать трудоёмкость задачи дискретного логарифмирования в группах точек практически применяемых эллиптических кривых.

В некоторых аналогичных случаях находятся способы использовать структурные особенности систем для построения более эффективных алгоритмов. В качестве примера результативного подхода к анализу структур систем уравнений для уменьшения

трудоёмкости их решения алгоритмами, основанными на применении аппарата базисов Грёбнера, приведём случай систем уравнений, получаемых при решении задачи дискретного логарифмирования в группах точек эллиптических кривых над конечными полями с использованием многочленов Семаева (впервые предложенными в [18]). Различные подходы [19 – 22], использующие структурные особенности систем, в отдельных случаях понижают асимптотическую сложность итоговых алгоритмов дискретного логарифмирования до теоретически субэкспоненциальной.

По мнению авторов, существенно более перспективным является подход, связанный с рассмотрением уравнения (1).

С точки зрения теоретико-информационного подхода представляется возможным, что построенные таким образом системы при правильном выборе параметров будут разрешимы и иметь единственное решение. Для демонстрации наличия такой возможности проведён следующий эксперимент:

1) для $q = 2^{32} - 5$ были выработаны случайные битовые векторы x_i, $1 \leqslant x_i < q$, $1 \leqslant i \leqslant 20$, играющие роль закрытых ключей;
2) каждый из этих битовых векторов умножался по модулю q на случайные значения $UKM_{i,j}$, $1 \leqslant UKM_{i,j} < q$, $1 \leqslant i \leqslant 20$, $1 \leqslant j \leqslant 32$;
3) полученные значения приводились по модулю 4. Набор из 32 значений по модулю 4 для вектора x_i далее будем называть спектром вектора x_i по $(UKM_{i,1}, \ldots, UKM_{i,32})$;
4) для каждого x', такого, что $1 \leqslant x' < q$, и для всех i, таких, что $1 \leqslant i \leqslant 20$, строились спектры x' по $(UKM_{i,1}, \ldots, UKM_{i,32})$;
5) построенные спектры для векторов x' сравнивались с соответствующими спектрами для векторов x_i, $1 \leqslant i \leqslant 20$.

Целью эксперимента являлась проверка гипотезы о том, что спектр вектора x_i однозначно определяет этот вектор среди всех возможных. Если бы спектр x' по $(UKM_{i,1}, \ldots, UKM_{i,32})$ для некоторого i совпал со спектром x_i по тому же кортежу значений UKM, но при этом $x' \neq x_i$, гипотеза была бы опровергнута. Экспериментально, однако, таких случаев выявлено не было, откуда следует, что во всех случаях целевой вектор x_i определялся однозначно своим спектром, что, в свою очередь, является свидетельством в пользу того, что в спектре содержится достаточно информации для восстановления искомого закрытого ключа.

3. Влияние рандомизации в случае неконстантного времени вычисления кратной точки

3.1. Механизмы вычисления кратных точек и побочные каналы по времени

Потребность в высокопроизводительных криптографических системах привела к необходимости разработки эффективных алгоритмов выполнения базовых криптографических операций. Одной из таких операций является задача скалярного умножения точки эллиптической кривой на множитель (или вычисления кратной точки). В последние десятилетия эта задача получила большое число решений для использования в разных криптографических примитивах и с разными представлениями эллиптических кривых. Интересный обзор таких решений можно найти в [23].

Время выполнения операции скалярного умножения зависит от битового вектора, представляющего множитель. Поэтому при выборе алгоритма вычисления кратной точки важно учитывать наличие побочного канала по времени. Кратко рассмотрим

два классических представителя семейств алгоритмов вычисления кратных точек и оценим возможность утечки данных по временному каналу при использовании таких методов.

Введём следующие обозначения:

— умножаемую точку будем обозначать через P (точка на эллиптической кривой), множитель — через m (неотрицательное целое число). Таким образом, требуется по P и m найти точку $[m]P$;
— бинарное разложение m будем обозначать $m_{(2)} = (m_{n-1}, \ldots, m_1, m_0)$, где $m_i \in \in \{0, 1\}$, $i = 0, 1, 2, \ldots, n-1$; n — длина бинарного разложения. С точностью до старших нулей бинарное разложение однозначно. Через $\mathrm{wt}_2(m)$ будем обозначать вес бинарного разложения числа m.

Для всякого алгоритма умножения точки на число будем использовать следующие обозначения:

— ID — число удвоений точек на этапе предварительных вычислений;
— IA — число сложений точек на этапе предварительных вычислений;
— IS — объём хранимых предвычисленных данных (в точках на кривой);
— $D_{\max}, D_{\min}, D_{\exp}$ — максимальное, минимальное и среднее (по всем m длины n бит) количество требуемых удвоений точек;
— $A_{\max}$, $A_{\min}$, $A_{\exp}$ — максимальное, минимальное и среднее (по всем m длины n бит) количество требуемых сложений точек.

Классическая схема Горнера

Выражение $[m]P$ можно представить в следующем виде:

$$[m]P = \Big[\underbrace{\sum_{i=0}^{n-1} 2^i m_i}_{=m}\Big] P = [2]([2](\ldots[2]([2]([2][m_{n-1}]P \oplus [m_{n-2}]P) \oplus [m_{n-3}]P) \oplus \ldots) \oplus [m_1]P) \oplus [m_0]P.$$

Обозначим для $k = 0, 1, \ldots, n-1$ через P_k точку $\left[\sum\limits_{i=k}^{n-1} 2^{i-k} m_i\right] P$. Тогда $[m]P = P_0$, $P_{n-1} = [m_{n-1}]P$, $P_{k-1} = [2]P_k \oplus [m_{k-1}]P$, $k = n-1, \ldots, 2, 1$. Теперь

$$[m]P = \overbrace{[2](\overbrace{[2](\ldots[2]([2]([2]\underbrace{[m_{n-1}]P}_{P_{n-1}} \oplus [m_{n-2}]P) \oplus [m_{n-3}]P) \oplus \ldots) \oplus [m_1]P}^{P_1}) \oplus [m_0]P}^{P_0}.$$

$$\underbrace{\qquad\qquad\qquad\qquad}_{P_{n-2}}$$

Алгоритм 1 описывает действия по этой схеме. Предварительные вычисления отсутствуют.

Заметим, что количество удвоений (D) в алгоритме GornerClassic совпадает с номером старшей единицы бинарного разложения числа m и равно $\lfloor \log_2(m) \rfloor$; на шаге 5 сложение осуществляется только при $m_k = 1$, поэтому количество сложений точек (A) на 1 меньше количества единиц в бинарном разложении числа m. Таким образом, для алгоритма 1 точные количества сложений и удвоений точек зависят от m:

$$A = \mathrm{wt}_2(m) - 1, \quad D = \lfloor \log_2(m) \rfloor.$$

Алгоритм 1. GornerClassic

Вход: $m = \sum\limits_{i=0}^{n-1} m_i 2^i$ — натуральное число; P — элемент некоторой аддитивной группы.
Выход: $Q = [m]P$ — элемент, m-кратный элементу P.
1: $Q := P$;
2: $n' :=$ номер старшей единицы в бинарном разложении $m_{(2)}$;
3: $k := n' - 1$.
4: **Пока** $k \geqslant 0$:
5: $\quad Q := [2]Q \oplus [m_k]P$;
6: $\quad k := k - 1$.
7: **Вернуть** Q

Основные вычисления:

— $D_{\max} = n - 1$, $D_{\min} = 0$, $D_{\exp} = n - 2 + \dfrac{1}{2^{n-1}} = \sum\limits_{k=0}^{n-1} \dfrac{k}{2^{n-k}}$;

— $A_{\max} = n - 1$, $A_{\min} = 0$, $A_{\exp} = \dfrac{n}{2} - 1$.

Обратим внимание на то, что время выполнения алгоритма напрямую зависит от веса двоичного разложения m. Таким образом, в случае применения схемы Горнера по времени вычисления можно получить информацию о весе ключа, тем самым резко ограничив размер класса возможных закрытых ключей. Простейшим приёмом, позволяющим избавиться от подобного побочного канала по времени, является добавление лишнего сложения при $m_k = 0$, результат которого игнорируется. На рис. 1 показано влияние веса w, $0 \leqslant w < 512$, для исходного (чёрный график) и доработанного с помощью указанного приёма (серый график) алгоритмов на время выполнения скалярного умножения точки эллиптической кривой.

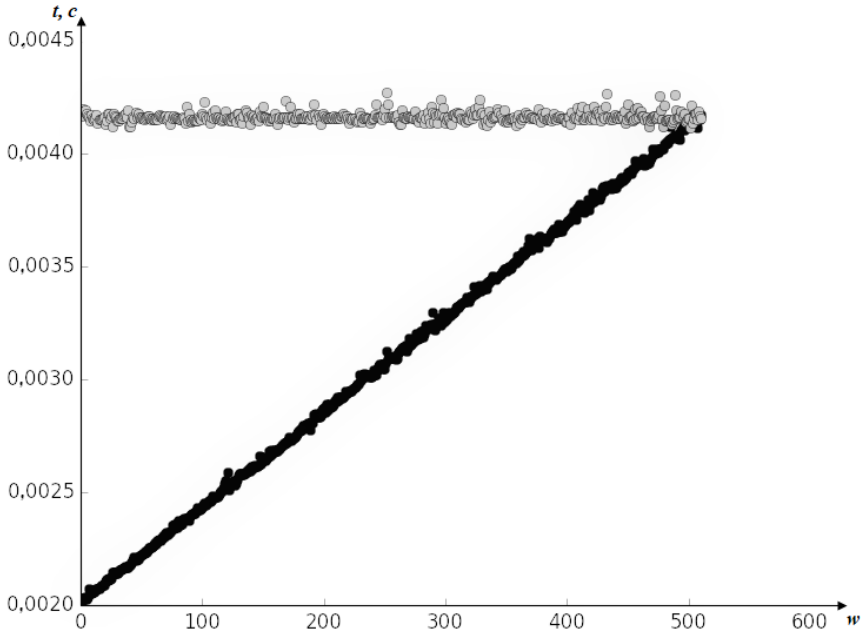

Рис. 1. Зависимость времени вычисления кратной точки методом Горнера от битового веса множителя

Алгоритм WTNAF

Алгоритм WTNAF является представителем семейства так называемых оконных алгоритмов и основан на w-TNAF-представлении числа m, которым называется набор

$$((m^{(s-1)}, p^{(s-1)}), (m^{(s-2)}, p^{(s-2)}), \ldots, (m^{(1)}, p^{(1)}), (m^{(0)}, p^{(0)})),$$

$m^{(i)} \in \{-2^{w+1}+1, -2^{w+1}+3, \ldots, 2^{w+1}-3, 2^{w+1}-1\}$, $p^{(i)} \in \mathbb{N}$, $i = 0, 1, \ldots, s-1$, такой, что $m = \sum\limits_{i=0}^{s-1} 2^{\sum\limits_{j=0}^{i} p^{(j)}} m^{(i)}$; $m^{(s-1)} \neq 0$.

В этом случае

$$[m]P = \sum_{i=0}^{s-1} \left[2^{\sum\limits_{j=0}^{i} p^{(j)}} m^{(i)} \right] P.$$

Заметим, что можно заранее вычислить значения $[r]P$ для всех $r \in \{1, 3, \ldots, 2^{w+1}-3, 2^{w+1}-1\}$. При необходимости сложения произвольной точки с точкой $[r']P$, где $r' = -r$ для $r \in \{1, 3, \ldots, 2^{w+1}-3, 2^{w+1}-1\}$, сложение заменяется на вычитание точки $[r]P$.

Эти действия описаны в алгоритме 2.

Алгоритм 2. WTNAF

Вход: m — натуральное число вместе с его w-TNAF-представлением $((m^{(s-1)}, p^{(s-1)}), (m^{(s-2)}, p^{(s-2)}), \ldots, (m^{(1)}, p^{(1)}), (m^{(0)}, p^{(0)}))$,
P — элемент некоторой аддитивной группы,
таблица точек $[r]P$ для всех $r \in \{1, 3, \ldots, 2^{w+1}-3, 2^{w+1}-1\}$.
Выход: $Q = [m]P$ — элемент, m-кратный P.
1: $k := s-1$; $Q := O$.
2: **Пока** $k \neq 0$:
3: $Q := [2^{p^{(k)}}](Q \oplus [m^{(k)}]P)$;
4: $k := k-1$.
5: **Вернуть** Q

Предварительные вычисления: $ID = 1$; $IA = 2^w - 1$; $IS = 2^w$.

Основные вычисления:

— $D_{\max} = n-1$, $D_{\min} = 0$, $D_{\exp} \approx n-2$;
— $A_{\max} = \left\lceil \dfrac{n}{w+2} \right\rceil$, $A_{\min} = 0$, $A_{\exp} \approx \dfrac{n}{w+3}$.

На каждой итерации алгоритма происходит одна операция удвоения точки и максимум одна операция сложения. При этом разное время исполнения итераций может быть получено только за счёт того, что на некоторых итерациях $m^{(k)} = 0$ и сложения не требуется. Этого можно избежать одним из двух способов:

1) в некоторых представлениях точка O может быть представлена в аффинных координатах и при $m^{(k)} = 0$ можно осуществлять полноценную операцию сложения с такой точкой;
2) в таблице можно хранить не точки $[m^{(k)}]P$ непосредственно, а точки $[m^{(k)}]P \oplus M$, где M — маскирующая точка, такая, что $[m^{(k)}]P \oplus M \neq O$ для любых $0 \leqslant k \leqslant s-1$. Такой подход потребует дополнительного вычитания константной точки в конце алгоритма.

3.2. Влияние рандомизации на атаки

Подход, приводящий к возможности построения атаки на схему выработки общего ключа VKO при некорректном умножении на кофактор подгруппы точек эллиптической кривой при применении случайных значений UKM, может быть применён и при анализе информации, поступающей по побочным каналам при использовании специальных алгоритмов вычисления кратных точек. Далее через x будем обозначать закрытый ключ честной стороны, который пытается определить противник.

Как было показано, при фиксированном значении UKM в конечном протоколе противник может (при применении честной стороной схемы Горнера) определить вес вектора $UKM \cdot x \bmod q$, однако дополнительной информации он более не получит. Если используются рандомизированные значения UKM, противник может получить битовые веса ряда векторов вида $x \cdot UKM_i \bmod q$, $1 \leqslant i \leqslant N$, где N — параметр, определяемый вычислительными ресурсами противника и ограничениями протокола. Отметим, что, в отличие от случая некорректной реализации схемы VKO, в данном случае нужно учитывать «зашумлённость» получаемых значений битовых весов. Для избавления от «шума» необходимо сделать несколько повторов операций с каждым конкретным значением UKM_i, что возможно в случае сообщений формата CMS (где значения UKM выбираются противником), но крайне затруднено в случае протокола TLS (противник может рандомизировать значение UKM, но не контролирует его выработку полностью). Фактически противник получает следующую систему уравнений (через $a_1, \ldots, a_N$ обозначим конкретные полученные противником значения весов):

$$\begin{cases} \mathrm{wt}(x \cdot UKM_1 \bmod q) = a_1, \\ \ldots \\ \mathrm{wt}(x \cdot UKM_N \bmod q) = a_N. \end{cases} \tag{2}$$

Вопрос существования эффективного алгоритма, который позволяет находить единственное значение x по данным значениям UKM_i и полученным весам a_i множителей вида $x \cdot UKM_i \bmod q$, остаётся открытым, но можно сделать предположение о достаточности информации в полученных противником данных для восстановления закрытого ключа x с точки зрения теоретико-информационного подхода. Для иллюстрации этого предположения был проведён эксперимент, сходный с экспериментом для случая некорректной реализации схемы VKO.

Цель эксперимента — проверка гипотезы о том, что по весам множителей вида $x \cdot UKM_i \bmod q$ однозначно определяется подходящее значение x из ключевого пространства. Эксперимент проводился для q длин 24, 28 и 32 бита и состоял из следующих шагов:

- вырабатывались битовые векторы x_i, $1 \leqslant x_i < q$, $1 \leqslant i \leqslant 20$, играющие роль закрытых ключей;
- каждый из векторов x_i умножался на случайные значения $UKM_{i,j}$, $1 \leqslant UKM_{i,j} < q$, $1 \leqslant i \leqslant 20$, $1 \leqslant j \leqslant 32$;
- для каждого из векторов $x_i \cdot UKM_{i,j} \bmod q$ считался его битовый вес a_i. Далее для фиксированного i совокупность 32 таких весовых значений будем называть спектром вектора x_i по $(UKM_{i,1}, \ldots, UKM_{i,32})$;
- для каждого x', $1 \leqslant x' < q$, и для всех i, $1 \leqslant i \leqslant 20$, строились спектры x' по $(UKM_{i,1}, \ldots, UKM_{i,32})$;
- построенные спектры для векторов x' сравнивались с соответствующими спектрами для векторов $x_i, 1 \leqslant i \leqslant 20$.

Если бы спектр x' по $(UKM_{i,1}, \ldots, UKM_{i,32})$ для некоторого i совпал со спектром x_i по тому же кортежу значений UKM, но при этом $x' \neq x_i$, гипотеза была бы опровергнута. Экспериментально, однако, таких случаев выявлено не было, откуда следует, что во всех случаях целевой вектор x_i определялся однозначно своим спектром, что, в свою очередь, свидетельствует в пользу истинности предположения о наличии в спектре достаточной информации для восстановления искомого закрытого ключа.

Данные эксперимента показывают, что в случае появления математического алгоритма, восстанавливающего значение закрытого x по его весовому спектру по кортежу значений UKM небольшой длины, рандомизацию в схеме VKO следует использовать только совместно с методами противодействия побочным каналам по времени.

4. Задачи для дальнейших исследований

Построение эффективных методов решения систем уравнений (1) и (2), определяемых в соответствии с описанными сценариями атак, представляется весьма сложной и важной задачей для дальнейших исследований. Она может оказаться интересной, например, для специалистов в области теории решёток.

Отметим, что разбиение процесса построения законченных методов криптоанализа реальных систем на два самоценных этапа является распространённой практикой. Первым является этап сведения задачи реализации актуальной для системы угрозы к некоторой формальной математической задаче, а за ним следует этап исследования методов решения этой задачи. При этом в ряде случаев эти этапы исследуются независимо в разных, разнесённых во времени и авторству работах. Ярким примером является история криптографического анализа схемы подписи вслепую Шнорра. В 2001 г. в работе [24] Шнорр свёл задачу криптоанализа этой схемы к решению так называемой «ROS-задачи», не предложив эффективного алгоритма её решения. И только в 2021 г. в работе [25] другой группой исследователей предложен полиномиальный алгоритм решения этой задачи. Учитывая широту спектра применяемых сегодня в криптоанализе математических методов и постоянное усложнение исследуемых криптосистем, тенденция к разделению задачи оценки стойкости таких систем на два указанных этапа вполне естественна. Другие примеры подобного характера можно найти в [26, разд. 4.1].

Заключение

Рассмотренные аспекты позволяют сделать следующий общий вывод о принципах обеспечения защиты информации в слабодоверенном окружении. Применение механизмов, позволяющих в ряде случаев обеспечивать защищённость программных компонент средств защиты информации при отсутствии полного доверия к программному и аппаратному окружению, требует максимально внимательной разработки самих этих компонент. Цена за возможность обеспечивать защиту программных модулей в слабодоверенном окружении — недопустимость ошибок в самих этих модулях, предельно высокие требования к качеству и глубине исследований программных средств защиты информации, предполагающих применение в средах, которые не подвергаются полноценному анализу.

Авторы благодарят Владимира Пузырёва за существенную помощь в организации вычислительных экспериментов.

Авторы выражают глубокую признательность рецензенту за детальные и содержательные комментарии о подходах к решению получаемых в результате описанных сценариев атак систем уравнений.

ЛИТЕРАТУРА

1. *Diffie W. and Hellman M.* New directions in cryptography // IEEE Trans. Inform. Theory. 1976. V. 22. No. 6. P. 644–654.
2. Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography. NIST Special Publication 800-56A Revision 3. `https://doi.org/10.6028/NIST.SP.800-56Ar3`. 2018.
3. *Rescorla E.* The Transport Layer Security (TLS) Protocol Version 1.3. `https://tools.ietf.org/html/rfc8446`. 2018.
4. ГОСТ Р 34.11-2012 «Информационная технология. Криптографическая защита информации. Функция хэширования». М.: Стандартинформ, 2012.
5. *Popov V., Kurepkin I., and Leontiev S.* Additional Cryptographic Algorithms for Use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms. `https://tools.ietf.org/html/rfc4357`. 2006.
6. Рекомендации по стандартизации Р 50.1.113–2016 «Информационная технология. Криптографическая защита информации. Криптографические алгоритмы, сопутствующие применению алгоритмов электронной цифровой подписи и функции хэширования». М.: Стандартинформ, 2016.
7. ГОСТ Р 34.10-2012 «Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи». М.: Стандартинформ, 2012.
8. *Smyshlyaev S., Alekseev E., Popov V., and Leontiev S.* Guidelines on the Cryptographic Algorithms to Accompany the Usage of Standards GOST R 34.10-2012 and GOST R 34.11-2012. `https://tools.ietf.org/html/rfc7836`. 2016.
9. Рекомендации по стандартизации Р 1323565.1.020-2020 «Информационная технология. Криптографическая защита информации. Использование российских криптографических алгоритмов в протоколе безопасности транспортного уровня (TLS 1.2)». М.: Стандартинформ, 2020.
10. Методические рекомендации МР 26.2.002-2013 «Использование алгоритмов ГОСТ 28147-89, ГОСТ Р 34.10 и ГОСТ Р 34.11 в криптографических сообщениях формата CMS». М.: Стандартинформ, 2013.
11. Техническая спецификация ТС 26.2.001-2015 «Использование ГОСТ 28147-89, ГОСТ Р 34.11-2012 и ГОСТ Р 34.10-2012 в протоколах обмена ключами IKE и ISAKMP». М.: Стандартинформ, 2015.
12. Рекомендации по стандартизации «Информационная технология. Криптографическая защита информации. Использование российских криптографических алгоритмов в сети Интернет версии 2 (IKEv2)» (проект).
13. *Алексеев Е. К., Ошкин И. Б., Попов В. О. и др.* О перспективах использования скрученных эллиптических кривых Эдвардса со стандартом ГОСТ Р 34.10-2012 и алгоритмом ключевого обмена на его основе // Проблемы информационной безопасности. Компьютерные системы. 2014. № 3. С. 60–66.
14. *Alekseev E. K., Nikolaev V. D., and Smyshlyaev S. V.* On the security properties of Russian standardized elliptic curves // Матем. вопр. криптогр. 2018. Т. 9. № 3. С. 5–32.
15. SafeCurves: Choosing Safe Curves for Elliptic-Curve Cryptography. `https://safecurves.cr.yp.to/index.html`.
16. *Lim C. H. and Lee P. J.* A key recovery attack on discrete log-based schemes using a prime order subgroup // LNCS. 1997. V. 1294. P. 249–263.
17. *Biehl I., Meyer B., and Muller V.* Differential fault attacks on elliptic curve cryptosystems (extended abstract) // LNCS. 2000. V. 1880. P. 131–146.

18. *Semaev I. A.* Summation Polynomials and the Discrete Logarithm Problem on Elliptic Curves. Cryptology ePrint Archive: Report 2004/031. https://eprint.iacr.org/2004/031.pdf.
19. *Petit C. and Quisquater J.-J.* On polynomial systems arising from a Weil descent // LNCS. 2012. V. 7658. P. 451–466.
20. *Semaev I. A.* New Algorithm for the Discrete Logarithm Problem on Elliptic Curves. Cryptology ePrint Archive: Report 2015/310. https://eprint.iacr.org/2015/310.pdf.
21. *Courtois N.* On Splitting a Point with Summation Polynomials in Binary Elliptic Curves. Cryptology ePrint Archive: Report 2016/003. https://eprint.iacr.org/2016/003.pdf.
22. *Petit C., Kosters M., and Messeng A.* Algebraic approaches for the elliptic curve discrete logarithm problem over prime fields // LNCS. 2016. V. 9615. P. 3–18.
23. *Hankerson D., Menezes A. J., and Vanstone S.* Guide to Elliptic Curve Cryptography. N.Y.: Springer Verlag, 2004.
24. *Schnorr C.-P.* Security of blind discrete log signatures against interactive attacks // LNCS. 2001. V. 2229. P. 1–12.
25. *Benhamouda F., Lepoint T., Loss J., et al.* On the (in)security of ROS' // LNCS. 2021. V. 12696. P. 33–53.
26. *Koblitz N. and Menezes A.* Critical perspectives on provable security: Fifteen years of "another look" papers // Adv. Math. Commun. 2019. V. 13. P. 517–558.

REFERENCES

1. *Diffie W. and Hellman M.* New directions in cryptography. IEEE Trans. Inform. Theory, 1976, vol. 22, no. 6, pp. 644–654.
2. Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography. NIST Special Publication 800-56A Revision 3. https://doi.org/10.6028/NIST.SP.800-56Ar3, 2018.
3. *Rescorla E.* The Transport Layer Security (TLS) Protocol Version 1.3. https://tools.ietf.org/html/rfc8446, 2018.
4. GOST R 34.11-2012 "Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Funktsiya kheshirovaniya" [GOST R 34.11-2012 "Information Technology. Cryptographic Data Security. Hash Function"]. Moscow, Standartinform, 2012. (in Russian)
5. *Popov V., Kurepkin I., and Leontiev S.* Additional Cryptographic Algorithms for Use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms. https://tools.ietf.org/html/rfc4357. 2006.
6. Rekomendatsii po standartizatsii R 50.1.113–2016 "Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Kriptograficheskie algoritmy, soputstvuyushchie primeneniyu algoritmov elektronnoy tsifrovoy podpisi i funktsii kheshirovaniya" [Recommendations for Standardization R 50.1.113–2016 "Information Technology. Cryptographic Data Security. Additional Cryptographic Algorithms for Digital Signature Algorithms and Hash Function"]. Moscow, Standartinform, 2016. (in Russian)
7. GOST R 34.10-2012 "Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Protsessy formirovaniya i proverki elektronnoy tsifrovoy podpisi" [GOST R 34.10-2012 "Information Technology. Cryptographic Data Security. Processes of Digital Signature Creation and Verification"]. Moscow, Standartinform, 2012. (in Russian)
8. *Smyshlyaev S., Alekseev E., Popov V., and Leontiev S.* Guidelines on the Cryptographic Algorithms to Accompany the Usage of Standards GOST R 34.10-2012 and GOST R 34.11-2012. https://tools.ietf.org/html/rfc7836. 2016.
9. Rekomendatsii po standartizatsii R 1323565.1.020-2020 "Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Ispol'zovanie rossiyskikh kriptograficheskikh algoritmov v protokole bezopasnosti transportnogo urovnya (TLS 1.2)" [Recommendations

for Standardization R 1323565.1.020-2020 “Information Technology. Cryptographic Data Security. Usage of Russian Cryptographic Algorithms in TLS 1.2 Protocol”]. Moscow, Standartinform, 2020. (in Russian)

10. Metodicheskie rekomendatsii MR 26.2.002-2013 “Ispol’zovanie algoritmov GOST 28147-89, GOST R 34.10 i GOST R 34.11 v kriptograficheskikh soobshcheniyakh formata CMS” [Methodical Recommendations MR 26.2.002-2013 “Usage of GOST 28147-89, GOST R 34.10 and GOST R 34.11 in CMS”]. Moscow, Standartinform, 2013. (in Russian)
11. Tekhnicheskaya spetsifikatsiya TS 26.2.001-2015 “Ispol’zovanie GOST 28147-89, GOST R 34.11-2012 i GOST R 34.10-2012 v protokolakh obmena klyuchami IKE i ISAKMP” [Technical Specification TS 26.2.001-2015 “Usage of GOST 28147-89, GOST R 34.11-2012 and GOST R 34.10-2001 for Key Agreement in IKE and ISAKMP Protocols”]. Moscow, Standartinform, 2015. (in Russian)
12. Rekomendatsii po standartizatsii “Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Ispol’zovanie rossiyskikh kriptograficheskikh algoritmov v seti Internet versii 2 (IKEv2)” (proekt) [Recommendations for Standardization “Information technology. Cryptographic data security. Usage of Russian cryptographic algorithms in Internet key exchange protocol version 2 (IKEv2)” (proekt)]. (in Russian)
13. *Alekseev E. K., Oshkin I. B., Popov V. O., et al.* O perspektivakh ispol’zovaniya skruchennykh ellipticheskikh krivykh Edvardsa so standartom GOST R 34.10-2012 i algoritmom klyuchevogo obmena na ego osnove [On the prospects of using twisted Edwards elliptic curves with the GOST R 34.10-2012 standard and the key exchange algorithm based on it]. Problemy Informatsionnoy Bezopasnosti. Komp’yuternye Sistemy, 2014, no. 3, pp. 60–66. (in Russian)
14. *Alekseev E. K., Nikolaev V. D., and Smyshlyaev S. V.* On the security properties of Russian standardized elliptic curves // Matem. Vopr. Kriptogr., 2018, vol. 9, iss. 3, pp. 5–32.
15. SafeCurves: Choosing Safe Curves for Elliptic-Curve Cryptography. `https://safecurves.cr.yp.to/index.html`.
16. *Lim C. H. and Lee P. J.* A key recovery attack on discrete log-based schemes using a prime order subgroup. LNCS, 1997, vol. 1294, pp. 249–263.
17. *Biehl I., Meyer B., and Muller V.* Differential fault attacks on elliptic curve cryptosystems (extended abstract). LNCS, 2000, vol. 1880, pp. 131–146.
18. *Semaev I. A.* Summation Polynomials and the Discrete Logarithm Problem on Elliptic Curves. Cryptology ePrint Archive: Report 2004/031. `https://eprint.iacr.org/2004/031.pdf`.
19. *Petit C. and Quisquater J.-J.* On polynomial systems arising from a Weil descent. LNCS, 2012, vol. 7658, pp. 451–466.
20. *Semaev I. A.* New Algorithm for the Discrete Logarithm Problem on Elliptic Curves. Cryptology ePrint Archive: Report 2015/310. `https://eprint.iacr.org/2015/310.pdf`.
21. *Courtois N.* On Splitting a Point with Summation Polynomials in Binary Elliptic Curves. Cryptology ePrint Archive: Report 2016/003. `https://eprint.iacr.org/2016/003.pdf`.
22. *Petit C., Kosters M., and Messeng A.* Algebraic approaches for the elliptic curve discrete logarithm problem over prime fields. LNCS, 2016, vol. 9615, pp. 3–18.
23. *Hankerson D., Menezes A. J., and Vanstone S.* Guide to Elliptic Curve Cryptography. N.Y., Springer Verlag, 2004.
24. *Schnorr C.-P.* Security of blind discrete log signatures against interactive attacks. LNCS, 2001, vol. 2229, pp. 1–12.
25. *Benhamouda F., Lepoint T., Loss J., et al.* On the (in)security of ROS’. LNCS, 2021, vol. 12696, pp. 33–53.
26. *Koblitz N. and Menezes A.* Critical perspectives on provable security: Fifteen years of “another look” papers. Adv. Math. Commun., 2019, vol. 13, pp. 517–558.