УДК 681.3.06

К ВОПРОСУ ПРИМЕНЕНИЯ И СОВЕРШЕНСТВОВАНИЯ СТАНДАРТА ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ В УКРАИНЕ

В.И. ДОЛГОВ, А.В. НЕЛАСАЯ, А.Н. ПОГОРЕЛЫЙ

Излагаются некоторые соображения и выводы, связанные с опытом применения национального стандарта электронной цифровой подписи ДСТУ 4145-2002 и приводится одно из предложений по совершенствованию протокола цифровой подписи вместе с аргументами по оценке его сложности и обеспечения стойкости.

The paper provides some considerations and conclusions connected with experience of using the national standard of electronic digital signature DSTU 4145-2002 and gives one of the offers on improvement of the protocol of digital signature alongside with arguments on estimation of its complexity and provision of stability.

ВВЕДЕНИЕ

В этом году нашему национальному стандарту электронной цифровой подписи ДСТУ 4145-2002 [1] исполнилось 5 лет, что является хорошим поводом для обмена опытом по его применению и обсуждения его шероховатостей (некоторых недостатков), которые хотелось бы сгладить (устранить). Само появление подобного стандарта стало заметным событием в украинской криптографии, и его **достоинства** получили заслуженную высокую оценку специалистов. К сожалению, за прошедшие 5 лет так и не состоялся процесс широкого внедрения электронной цифровой подписи (**ЭЦП**) в **реальную** жизнь граждан и государства. Как нам представляется, назревает необходимость принятия новой редакции ДСТУ, в которую помимо исправления досадных опечаток, необходимо внести существенные **дополнения**.

Этой статьей мы приглашаем специалистов к обсуждению необходимости таких дополнений и изменений. В ее небольшой первой части мы излагаем нашу позицию в вопросах применения национального стандарта, а в последующих разделах приводим одно из предложений по совершенствованию протокола цифровой подписи вместе с аргументами по оценке его сложности и сохранения стойкости.

1. О НЕКОТОРЫХ НЕДОСТАТКАХ НАЦИОНАЛЬНОГО СТАНДАРТА

Не касаясь иных, возможно, решающих аспектов внедрения ЭЦП, как математических, юридических и, наконец, просветительских, – объектом основного внимания этого раздела являются вопросы реализации алгоритма ЭЦП в технических устройствах.

Обсуждение опыта реализации процедур ЭЦП, в том числе и на форумах сети Интернет, собственный опыт разработки технических средств реализации алгоритма выработки и проверки цифровой подписи позволяют высказать следующие суждения.

На наш взгляд, потенциальному **абоненту** множества пользователей ЭЦП необходимо предоставить возможность максимальной **автономии** в вопросах генерации закрытого ключа d, выработки и проверки **ЭЦП**. В идеале – не только носитель ключа, но и модуль криптопреобразований должен быть **компактным**, содержать средства генерации случайного числа e и закрытого ключа абонента d, **защищенных** от считывания извне, обладать приемлемым **быстродействием** как при вычислении хеш-функции, так и при выработке и проверке ЭЦП. Если под внешним модулем понимать компактное устройство, подключаемое извне, и выполняющее функции формирования и проверки цифровой подписи, то, фактически, стандартом для его подключения является шина **USB** как в вопросе электропитания, так и в части информационного обмена.

В качестве основы подобного устройства могут быть универсальные микроконтроллеры, микроконтроллеры для цифровой обработки сигналов или ПЛИС. Существенным ограничением является отсутствие в системах команд этих устройств операций вычисления произведений полиномов и приведения по модулю – примитивному полиному, а в составе ПЛИС – операции вычисления полинома, обратного заданному. Тем не менее, алгоритмы вычисления хэш-функции и ЭЦП вполне реализуемы в микроконтроллерах, в частности, в таком, как ATmega.

Можно высказать также замечание общего характера. Вызывает некоторую **досаду** скудность таблицы с параметрами эллиптических кривых и порядков циклических подгрупп. На наш взгляд, в общем-то, **верное** решение – привести рекомендуемые примитивные полиномы, просит логичного завершения – указания параметров криптографически стойких кривых и порядков циклических подгрупп для соответствующих полиномов. **Статус** этих данных необходимо сохранить как **рекомендуемый**, но ни в коей мере не как справочный – это снимет часть вопросов, возникших и у нас, и, особенно, – в России, в отношении достоверности и гарантированности криптографической стойкости параметров эллиптических кривых. Рекомендуемые – значит государство гарантирует **математическую** корректность формул и расчетных соотношений, при-

веденных в стандарте и **криптографическую** стойкость формируемых на их основе ЭЦП. Далеко не каждая структура, аккредитованная на право работы с сертификатами и ключевыми данными ЭЦП, располагает высококвалифицированным математическим и криптографическим персоналом, оборудованием, методиками исследований и доказательств. Если же кто захочет отойти от ДСТУ – попутного ветра, но, повторим, что вошло в ДСТУ – **гарантируется** государством, а иначе смысл в ДСТУ?!

2. О ПУТЯХ И ВОЗМОЖНОСТЯХ СОВЕРШЕНСТВОВАНИЯ ПРОТОКОЛА ЦИФРОВОЙ ПОДПИСИ

В процессе создания специализированного устройства вычисления ЭЦП остро стал вопрос в отношении быстродействия алгоритма вычисления ЭЦП. Реализация алгоритма по схеме **Горнера** (удвоение-сложение) показала, что оперативной памяти микроконтроллера **ATmega128** вполне достаточно, а флэш-память имеет большой объем, доступный для записи констант. Таким образом, возможно ускорение вычислений в случае применения метода **гребня**, с прореживанием множителя "**каждый восьмой** бит". Для этого случая необходимо хранение в памяти X, Y-координат предварительно вычисленных **256** вариантов произведений базовой точки P на число. Тогда после **8** сложений и **7** удвоений получаем результат умножения P на **64**-битное число. Суммарное число операций сложения и удвоения существенно меньше числа сложений предварительно запомненных координат последовательности удвоенных точек P – не **32** (в среднем) для 64 бит множителя, а лишь постоянно **15**. Для вычислений популярного ныне варианта со степенью полинома **191** потребуется записать **3** банка предварительно вычисленных координат (каждый объемом в $(24 + 24)\times256 = 12$ кбайт) – **36** килобайт, выполнить **24** операции сложения и **7** операций удвоения, всего – **31** операцию. Оценка для случая предельной по ДСТУ степени полинома **509** выглядит следующим образом:

– объем памяти констант – 8 блоков по $(64+64) \times 256 = 32$, всего – **256** кбайт;

– количество операций сложения – **64**;

– количество удвоений – **7;**

– **всего операций – 71**.

Как видим, даже для наибольшей по ДСТУ степени полинома результаты вполне реализуемы, правда, возможно, с применением внешней постоянной (или флэш-) памяти.

К достоинствам подобного метода можно отнести и хорошие возможности распараллеливания вычислений по принципу "в каждом микроконтроллере выполняются операции умножения точки на **64** бита множителя". Завершающие операции суммирования отдельных фрагментов тоже могут быть скомпонованы по принципу дерева. Но даже для последовательного завершающего суммирования в случае полинома степени **191** потребуется **3** микроконтроллера, в каждом из которых задействована память констант **12** кбайт; всеми микроконтроллерами параллельно выполняются по 8 операций суммирования и 7 – удвоения, а всего – 15 операций; завершают вычисления 3 операции суммирования. Итоговые затраты времени – на **18** операций (11 операций суммирования и 7 – удвоения). Аналогичные выкладки для полинома **509**-й степени дадут результат: **8** микроконтроллеров, в каждом собственная память констант **32** кбайт, 15 операций выполняются всеми параллельно, 7 завершающих операций суммирования – всего **22** операции.

Реализация алгоритма **вычисления** ЭЦП требует существенно меньше времени, чем для **проверки**. Естественно, возникает вопрос: как ускорить проверку ЭЦП? Методика проверки подписи основана на операциях с индивидуальными открытыми ключами абонента, и, если реализовывать ее по ранее приведенным алгоритмам, то для проверки собственной ЭЦП понадобится удвоенная память констант - для базовой точки P и для открытого ключа Q. Но основная сложность возникает при проверке **множества** подписей абонентов в центрах или у потребителей документов множества абонентов. Нельзя ли заменить **методику** проверки ЭЦП так, чтобы использовать **одну** базовую точку множества корпоративных абонентов, а результат проверки сравнивать с открытым ключом абонента? Такой подход позволил бы хранить в базах данных корпоративные базовые точки и в полной мере использовать возможности предвычислений, которые ныне бессмысленны для множества абонентов. Разумеется, с математической и криптографической точек зрения изменения методик выработки и проверки ЭЦП не должны затрагивать вопросы криптографической стойкости и математической эквивалентности. Одно из разработанных нами предложений в этом направлении излагается ниже.

3. МОДИФИЦИРОВАННЫЙ ПРОТОКОЛ ЦИФРОВОЙ ПОДПИСИ ДЛЯ КОРПОРАТИВНОЙ СЕТИ ECPR

Анализ протоколов цифровой подписи на эллиптических кривых [1-4] таких, как **ДСТУ 4145, ГОСТ Р 34.10, ECDSA**, **ECSS, EC-GDSA, EC-KCDSA,** алгоритм **Шнорра,** показал, что в процедуре верификации подписи во всех перечисленных протоколах присутствует операция скалярного умножения открытого ключа Q на число. Как упоминалось ранее, для этой операции использование предвычислений бессмысленно, так как в отличие от базовой точки P, которая является единой для всех абонентов корпоративной сети, открытый ключ Q у каждого свой.

Главная идея предлагаемого авторами подхода – уйти от умножения открытого ключа Q на какой бы то ни было числовой множитель. В табл. 1 приведен один из вариантов реализации

Таблица 1

Модифицированный протокол цифровой подписи на эллиптической кривой ECPR

Формирование ЭЦП	**Верификация ЭЦП**
Вход: эллиптическая кривая C, секретный ключ d, открытый ключ $Q = -d \times P$, P – базовая точка, n – порядок базовой точки, сообщение M. **Выход**: ЦП $\langle r, s \rangle$ для сообщения M.	**Вход**: эллиптическая кривая C, открытый ключ Q, P – базовая точка, n – порядок базовой точки, ЦП $\langle r', s' \rangle$, для сообщения M'. **Выход:** Подпись действительна или нет.
1. $h = H(M) \bmod n$; **2.** Генерируем случайное $k \in \{1, ..., n-1\}$; **3.** $t = \frac{k}{h} \bmod n$; **4.** $R = t \times P$; **5.** $w = \pi(R) \bmod n$; **6.** $(x, y) = w \times R = w \times t \times P$; **7.** $r = \pi(x, y) \bmod n$; **8.** $s = (wk + hd) \bmod n$.	**1.** $h' = H(M) \bmod n$; **2.** $t = \frac{s'}{h'} \bmod n$; **3.** $(x', y') = t \times P + Q$; **4.** $v = \pi(x, y) \bmod n$; **5.** $r' \overset{?}{=} v$

подобного подхода. Назовем этот протокол ECPR (Elliptic Curve with PRecomputation).

Обозначения: $h(M)$ – хеш-функция, π – функция выделение $x \bmod n$.

Определяющее соотношение для предложенного протокола выглядит следующим образом:

$$\frac{(wk + hd)}{h} P + Q = \frac{wk}{h} P + dP - dP = \frac{wk}{h} P .$$

Как видно из процедуры формирования подписи $r = \pi\left(\frac{wk}{h} P\right)$. Соответственно в процедуре верификации $v = \pi\left(\frac{wk}{h'} P\right)$. Следовательно, значения r и v равны, если сообщения и подписи в обоих случаях идентичны.

4. ОЦЕНКА СЛОЖНОСТИ ПРОТОКОЛА ЦИФРОВОЙ ПОДПИСИ ECPR

В табл. 2 для сравнения приведены данные по количеству различных операций при использовании предлагаемого авторами протокола цифровой подписи и других (основных) стандартов ЭЦП на эллиптических кривых.

За счет введения одной дополнительной операции умножения в поле, одной операции инверсии в поле и одной операции **скалярного умножения на базовую точку**, которую можно выполнять с использованием предвычислений, в процедуре формирования подписи, разгружается процедура проверки подписи. В этом случае, в отличие от других стандартов, **отсутствует** наиболее трудоемкая операция скалярного умножения на открытый ключ Q, которую затруднительно выполнять с предвычислениями в корпоративной сети.

Приведенные алгоритмы были реализованы авторами с использованием библиотеки Miracl. В случае простого поля длина модуля преобразований была выбрана 256 бит. В случае расширенного поля в качестве модуля был выбран полином 191 степени. Величина окна для предвычислений в обоих случаях выбрана равной пяти. При измерении времени выполнения операций формирования и проверки время предвычислений не учитывалось. В случае простого поля оно составило

Таблица 2

Сравнительный анализ протоколов цифровой подписи по количеству выполняемых операций

Операция	ECPR		ГОСТ Р 34.10		**ДСТУ 4145**		**ECDSA**	
	Формирование	Проверка	Формирование	Проверка	Формирование	Проверка	Формирование	Проверка
$h(M)$	1	1	1	1	1	1	1	1
$\pi(x, y) \bmod n$	2	1	1	1	2[1]	1[2]	1	1
Умножение в F_n	3	1	2	2	2	1	2	2
Инверсия в F_n	1	1	–	1	–	–	1	1
Групповое сложение точек ЭК	–	1	–	1	–	1	–	1
Скалярное умножение на P	2	1	1	1	1	1	1	1
Скалярное умножение на Q	–	–	–	1	–	1	–	1

[1] В процедуре формирования один раз используется операция π и один раз операция преобразования элемента поля $GF(2^m)$ в число.

[2] В процедуре проверки используется операция преобразования элемента поля $GF(2^m)$ в число.

Таблица 3

Сравнительный анализ протоколов по скорости выполнения операций формирования и проверки цифровой подписи

	ECPR над $GF(p)$		ГОСТ Р 34.10 над $GF(p)$		ECDSA над $GF(p)$		ECPR над $GF(2^m)$		ДСТУ 4145 над $GF(2^m)$	
	Формирование	Проверка	Формирование	Проверка	Формирование	Проверка	Формирование	Проверка	Формирование	Проверка
Время, мкс	2,6E+08	1,4E+08	1,3E+08	5,0E+08	1,3E+08	5,2E+08	1,6E+08	8,7E+07	8,0E+07	2,4E+08

приблизительно 2,2E+04 мкс, в случае расширенного – 1,0 E+04 мкс. Эксперименты проводились на компьютере Dell Inspiron I6400 Intel(R) Core(TM)2 CPU T5600@1.83GHz 987МГц, 512 МБ ОЗУ. В табл. 3 приведены оценки времени выполнения 100000 операций формирования и проверки цифровой подписи для различных протоколов.

Приведенные результаты свидетельствуют, что для всех алгоритмов над простым полем мы проигрываем приблизительно в два раза в процедуре формирования подписи, но при этом выигрываем в четыре раза при проверке подписи. Над расширенным полем выигрыш при выполнении проверки подписи достигает 2,75 раза. Эти показатели можно улучшить, если увеличить окно предвычислений.

Предложенный подход был также опробован на моделях алгоритмов ЭЦП в соответствии с ДСТУ 4145 и ECPR над $GF(2^m)$, написанных на языке Паскаль. Проверялись соотношения затрат времени на основные операции для степеней расширений поля 191 и 431 при вычислениях по схеме удвоение-суммирование без предварительных вычислений, затем с предвычисленными значениями координат точек 2eM∗P (для M точек) и предвычисленными координатами 256 точек для комбинаций с выборкой через 24 бита (для M = 191) или 54 бита (для M = 431). Другие приемы оптимизации вычислений (метод Карацубы и т.п.) и специализированные библиотеки не применялись. В табл. 4 приведен средний коэффициент выигрыша по времени выполнения операций с предвычисленными значениями координат относительно времени выполнений без предвычислений.

Впоследствии эта модель алгоритма была реализована в микроконтроллере ATmega128. В табл. 5 приведены основные характеристики быстродействия вариантов алгоритмов:

– удвоение-суммирование без предвычислений;

– удвоение-суммирование предвычисленных значений 256 комбинаций точек с выборкой через 24 или 54 бита, аналогично указанному выше.

Таблица 4

Сравнительный анализ операций вычисления ЭЦП по ДСТУ 4145 (без предвычислений) и ECPR над $GF(2^m)$ (с предвычислениями).

Степень полинома M	Операции	Без предвычислений, удвоение-суммирование	С предвычисленными координатами точек 2eM∗P	С предвычисленными координатами 256 точек
191	Количество операций «Удвоение точки»	190	Нет	24
	Коэффициент ускорения	1	–	7
	Среднее к-во операций «Суммирование точек»	96	96	24
	Коэффициент ускорения	1	1	4
	Ускорение времени вычисления ЭЦП	1	3	6
	Объем памяти таблиц предвычисленных значений, байт	–	9168	12288
431	Количество операций Удвоение точки	430	Нет	54
	Коэффициент ускорения	1	–	7,7
	Среднее к-во операций Суммирование точек	216	216	54
	Коэффициент ускорения	1	1	4
	Ускорение времени вычисления ЭЦП	1	3	6
	Объем памяти таблиц предвычисленных значений, байт	–	46548	27648

Вариант с предвычисленными координатами последовательно удвоенных базовых точек не применялся, как существенно уступающий по быстродействию во всем диапазоне полиномов, и не имеющий существенного выигрыша в длине таблиц даже для малых степеней расширения (а для больших – существенно проигрывающих, см. табл. 4).

Основные параметры для сравнения – средняя длительность процедуры вычисления в периодах тактовой частоты микроконтроллера. Время приведено для тактовой частоты 7,37 МГц.

По приведенным результатам можно сделать вывод, что даже без распараллеливания, время вычислений ЭЦП соответствует потребностям большого, или даже преобладающего числа абонентов. К тому же имеются резервы повышения быстродействия:

– в 2 раза за счет повышения тактовой частоты микроконтроллера до 14,75 МГц (допустимая для семейства частота 16 МГц);

– в 2-4-6-8 раз за счет распараллеливания вычислений между соответствующим количеством микроконтроллеров;

– применение 32-разрядных и высокочастотных процессоров обработки сигналов.

5. ОЦЕНКА СТОЙКОСТИ ПРОТОКОЛА ЦИФРОВОЙ ПОДПИСИ ECPR

Мы начнем с того, что напомним результаты анализа ряда существующих протоколов ЦП, приведенные в работах [5, 6]. Их основной смысл сводится к следующему.

1. Угрозы от атак на ЦП на связанных ключах могут быть реализованы только для двух наперед заданных (известных) сообщений.

2. Для EC-DSA при известных хэш значениях сообщений h_1 и h_2 (т.е. известных сообщениях M_1 и M_2), а также при всегда доступных значениях $r_1 = r_2$ и n нарушитель всегда может определить личный ключ. В результате, использование π функции вырезания только x координаты при вычислении открытого сеансового параметра r ЦП делает стандарт EC-DSA чувствительным как к возможности создания коллизий ЦП, так и к возможности определения личного ключа (если известны сообщения M_1 и M_2). Если эти сообщения обрабатываются в системе, то возникает угроза и для самого нарушителя – его личный ключ d при выполнении жульничества также компрометируется. Для защиты от этой угрозы владелец личного ключа d должен его изменить сразу после появления двух таким образом подписанных сообщений.

3. Для протокола ЦП EC-KCDSA значения открытых ключей r_1 и r_2 соответственно для k_1 и $k_2 = n - k_1$ не совпадают, то есть в алгоритме EC-KCDSA при этих значениях практически невозможна коллизия r компонент ЦП для двух произвольных сообщений M_1 и M_2. Для осуществления атаки необходимо вычислить k_1. Но k_1 зависит от значения r_1, которое в свою очередь сложно, однонаправлено через $H(c_1)$ зависит от k_1. Решение уравнения для нахождения k_1 имеет субэкспоненциальную сложность. Поэтому осуществление атаки на связанных ключах для EC-KCDSA экспоненциально сложно и практически невозможно.

Таким образом, алгоритм ЦП EC-DSA стандарта ISO/IEC 15946-2 является незащищенным от атаки на связанных ключах, в то время как алгоритм EC- KCDSA – защищен от такой атаки.

4. Алгоритмы ЦП по ДСТУ 4145-2002 и ECSS имеют слабую защиту от атак на связанных ключах. Авторы работы [1] отмечают, что для определения уровня защищенности ДСТУ 4145-2002 необходимо провести дополнительные исследования, при проведении которых предлагается особенное внимание обратить на то, что компоненты s_1 и s_2 совпадают, а также на тот факт, что параметры r_1 и r_2 являются взаимосвязанными.

5. Для алгоритмов ЦП EC-DSA, EC-GDSA, EC-KCDSA, ДСТУ 4145-2002, ECSS и ГОСТ Р 34.10-2001 существуют атаки на программную реализацию ЦП. Если нарушителю удается «вынудить» программу выработки подписи два раза использовать одно и то же значение параметра k для двух сообщений, то у него появляется возможность в реальном времени определить личный долгосрочный ключ d и поэтому он сможет как навязывать ошибочные сообщения, так и изменять истинные. Для защиты от такой атаки, отмечают авторы работы [2], необходимо использовать надежные средства КЗИ типа ЦП в смысле наличия на них сертификатов соответствия, экспертных выводов и обязательного непрерывного

Таблица 5

Сравнительные характеристики продолжительности вычислений по ДСТУ 4145 (без предвычислений) и ECPR над $GF(2^m)$ (с предвычислениями)

Степень полинома M	Длительность операции	Без предвычислений, удвоение-суммирование		С предвычисленными координатами 256 точек	
		Периодов	Секунд	Периодов	Секунд
191	Удвоение точки	650 000	0,088	650 000	0,088
	Суммирование точек	505 000	0,069	505 000	0,069
	Вычисление ЭЦП	172 000 000	23,5	27 800 000	3,8
431	Удвоение точки	2 920 000	0,40	2 920 000	0,40
	Суммирование точек	2 280 000	0,31	2 280 000	0,31
	Вычисление ЭЦП	1 760 000 000	239	285 000 000	39

контроля целостности и истинности программы выработки ЦП. Наилучшим способом защиты от отмеченной угрозы является аппаратная реализация процедур выработки и проверки ЦП.

Общий вывод, который можно сделать по представленным выше результатам, состоит в том, что существующим протоколам ЦП свойственны определенные недостатки, с которыми надо считаться.

Нашей задачей является анализ защищенности модифицированного протокола ECPR, применение которого позволяет защититься от ряда отмеченных угроз.

5.1. Анализ защищенности алгоритма ЦП ECPR от атак на связанных ключах

Речь идет о формировании одной и той же цифровой подписи при передаче двух разных сообщений M_1 и M_2.

Выберем, как и в работе [1] связанные ключи k_1, $k_2 = n - k_1$ и выясним условия, при которых ЦП $\langle r_1, s_1 \rangle$ и $\langle r_2, s_2 \rangle$ для сообщений M_1 и M_2 будут одинаковыми.

Алгоритм выработки ЦП для сообщений M_1 и M_2 предопределяет выполнение операций, указанных в табл. 6:

При разных сообщениях M_1 и M_2 равенство $r_1 = r_2$ возможно либо при наличии коллизии:

$$h_1 = h_2 \rightarrow t_2 = -t_1 \rightarrow R_2 = -R_1 \rightarrow w_1 = w_2$$
$$\rightarrow x_{R_1} = x_{R_2} \rightarrow (x_1, y_1) = (x_{R_1} \cdot t_1) \times P,$$
$$(x_2, y_2) = w_2 \times R_2 = (x_{R_1} \cdot (-t_1)) \times P =$$
$$= -(x_{R_1} \cdot t_1) \times P \rightarrow r_1 = r_2,$$

либо равенство $t_1 = t_2$ можно получить при разных хэш-значениях $h_1 \neq h_2$. Но при $\frac{k_1}{h_1} = \frac{k_2}{h_2}$, равенство $k_2 = n - k_1$ приводит к условию $\frac{k_1}{h_1} = \frac{n - k_1}{h_2}$, либо $\frac{k_1}{h_1} = \frac{-k_1}{h_2} \rightarrow k_1 h_2 = -k_1 h_1 \rightarrow k_1 (h_2 + h_1) = 0 (\mathrm{mod}\, n)$.

Так как необходимо, чтобы выполнялось еще и равенство $s_1 = s_2$, то, приравнивая соответствующие выражения в восьмой строке, получим:

В первом случае при $h_1 = h_2$, а, следовательно, при $w_1 = w_2$ имеем

$$(w_1 k_1 + h_1 d) (\mathrm{mod}\, n) = (w_1 (n - k_1) + h_1 d) (\mathrm{mod}\, n) \rightarrow$$
$$\rightarrow 2 w_1 k_1 \equiv 0 (\mathrm{mod}\, n),$$

что маловероятно.

Во втором случае при $h_1 \neq h_2$ получим

$$(w_1 k_1 + h_1 d) (\mathrm{mod}\, n) = (w_2 (n - k_1) + h_2 d) (\mathrm{mod}\, n) \rightarrow$$
$$(w_1 k_1 + h_1 d + w_2 k_1 - h_2 d) \equiv 0 (\mathrm{mod}\, n) \rightarrow$$
$$\rightarrow k_1 (w_1 + w_2) + d (h_1 - h_2) \equiv 0 (\mathrm{mod}\, n),$$

что также маловероятно.

В результате можно сделать вывод, что предложенный протокол является более устойчивым к атаке со связанными ключами, чем протоколы, рассмотренные в [5,6].

5.2. Атака при повторном использовании разового ключа

В этом случае пользователь для двух разных сообщений M_1 и M_2 с соответствующими хэш-кодами h_1 и h_2 при формировании подписи повторно использует одно и то же случайное значение k.

В случае протокола ECPR имеем:

$$\begin{cases} s_1 = (w_1 k + h_1 d) \bmod n, \\ s_2 = (w_2 k + h_2 d) \bmod n; \end{cases} \rightarrow$$
$$\rightarrow \begin{cases} h_1 d = (s_1 - w_1 k) \bmod n, \\ h_2 d = (s_2 - w_2 k) \bmod n. \end{cases}$$

Складывая и вычитая последние равенства, можем получить

$$\begin{cases} d = \dfrac{s_1 + s_2 - (w_1 + w_2) k}{h_1 + h_2} (\mathrm{mod}\, n), \\ d = \dfrac{s_1 - s_2 - (w_1 - w_2) k}{h_1 - h_2} (\mathrm{mod}\, n). \end{cases}$$

Таблица 6

Алгоритм выработки ЦП

Для сообщения M_1	Для сообщения M_2
1. Определяется $h_1 = H(M_1) (\mathrm{mod}\, n)$;	1. Определяется $h_2 = H(M_2) (\mathrm{mod}\, n)$;
2. Выбирается $0 < k_1 < n$;	2. Вычисляется $k_2 = n - k_1$;
3. Вычисляется $t_1 = \frac{k_1}{h_1} (\mathrm{mod}\, n)$;	3. Вычисляется $t_2 = \frac{k_2}{h_2} = \frac{n - k_1}{h_2} \rightarrow \frac{n - k_1}{h_2} \equiv -\frac{k_1}{h_2} (\mathrm{mod}\, n)$;
4. $R = (t_1 \times P) = (x_{R_1}, y_{R_1})$;	4. $R_2 = (t_2 \times P) = (x_{R_2}, y_{R_2}) \rightarrow R_1 \neq R_2$, если $t_1 \neq t_2$;
5. $w_1 = \pi(R_1) \bmod n = x_{R_1}$;	5. $w_2 = \pi(R_2) \bmod n = x_{R_2}$;
6. $(x_1, y_1) = w_1 \times R_1 = (x_{R_1} \cdot t_1) \times P$;	6. $(x_2, y_2) = w_2 \times R_2 = (x_{R_2} \cdot t_2) \times P$;
7. $r_1 = \pi(x_1, y_1) = x_1 (\mathrm{mod}\, n)$;	7. $r_2 = \pi(x_2, y_2) = x_2 (\mathrm{mod}\, n)$;
8. $s_1 = (w_1 k_1 + h_1 d) (\mathrm{mod}\, n)$.	8. $s_2 = (w_2 (n - k_1) + h_2 d) (\mathrm{mod}\, n)$.

В этих уравнениях известными являются суммы $s_1 + s_2$, $h_1 + h_2$ и разности $s_1 - s_2$, $h_1 - h_2$. Остальные компоненты: случайное значение k, секретный ключ d и значения $w_1 + w_2$ и $w_1 - w_2$ – неизвестны. Поскольку $w = \pi\left(\frac{k}{h}P\right)$, где π – функция выделения координаты x точки эллиптической кривой, то определение значений $w_1 + w_2$ и $w_1 - w_2$ связано с решением задач уровня дискретного логарифмирования (экспоненциальной сложности).

Общий вывод состоит в том, что если в существующих протоколах не допускается повторное использование ключа, то предлагаемый протокол оказывается защищенным от такой атаки.

5.3. Экзистенциальная подделка

Экзистенциальная подделка означает нахождение другого сообщения, возможно бессмысленного, которое подходит под подпись истинного сообщения. В этом случае по заданной подписи необходимо найти такой хэш, и далее, если хэш-функция недостаточно стойкая, можно подобрать ложное сообщение, подходящее под этот хэш.

Напомним, что обратная точка, вычисленная с использованием $(-k)$ или $(n-k)$ при одинаковых значениях хэш-кодов h_1 и h_2, дает то же значение одной из компонентов подписи, что и при значении случайного разового ключа k. В этом случае точки $\frac{k}{h_1}\times P$ и $\frac{-k}{h_1}\times P$ имеют одинаковую x-координату, которая используется в дальнейших расчетах. В этом случае мы получим одинаковый параметр w. Согласно формуле вычисления параметра подписи s (если мы хотим получить и вторую компоненту подписи, совпадающую с подписью сообщения на ключе k) в этом случае имеем:

$$\begin{cases} s = (wk + h_1 d) \bmod n, \\ s = (-wk + h_2 d) \bmod n. \end{cases}$$

Складывая эти уравнения, получим:

$$2s = (h_1 d + h_2 d) (\bmod n).$$

Следовательно,

$$h_2 = \left[(2s - h_1 d)/d\right] (\bmod n) = \left(\frac{2s}{d} - h_1\right) (\bmod n).$$

В этом случае, если злоумышленником является легальный пользователь, имеющий доступ к секретному ключу d, он может на основании истинного сообщения с хэшем h_1 и подписью $<r,s>$ по приведенной выше формуле вычислить хэш-значение ложного сообщения h_2 и подобрать под него ложное сообщение. Это может позволить ему отказаться от отправки истинного сообщения, утверждая, что он отправил ложное сообщение. Однако такая угроза является практически неосуществимой, если используемая хэш-функция отвечает требованиям криптографической стойкости.

5.4. Атака при условии наличия нескольких подписанных сообщений

При атаке на секретный ключ d аналитик может иметь в распоряжении последовательность M_1, M_2,..., M_m подписанных сообщений, для которых вычисление параметра подписи s

$$s = (wk + ed) \bmod n,$$

в соответствии с протоколом позволяет получить систему уравнений:

$$d = (s_i - w_i k_i) h_i^{-1} \bmod n =$$
$$= (s_i - \pi(\frac{k_i}{h_i} P) \cdot k_i) h_i^{-1} \bmod n,$$

где $i = 1, 2, \ldots, m$.

Здесь k_i и d – неизвестные криптоаналитику параметры. Таким образом, для нахождения d необходимо решить систему из m уравнений с $m+1$ неизвестными.

Способы решения подобных систем (с участием функции π) имеют сложность уровня дискретного логарифмирования.

ЗАКЛЮЧЕНИЕ

Таким образом, преимущество протокола ECPR заключается в том, что и при формировании и при проверке ЦП выполняется умножение только на базовую точку, которое в корпоративной сети может быть эффективно выполнено с предвычислениями. Экспериментальная проверка показала, что в сравнении с известными алгоритмами ЦП на эллиптических кривых, такими как ECDSA, ДСТУ 4145, ГОСТ 34.10, процедура формирования замедляется приблизительно в 2 раза, но процедура проверки, которая выполняется гораздо чаще, ускоряется приблизительно в 3-4 раза.

Проведенная оценка защищенности предложенного протокола показала, что он не уступает, а в некоторых случаях и является более защищенным, чем существующие стандарты.

Использование предложенного протокола позволит увеличить скорость процедуры верификации цифровой подписи, в частности, при реализации на специализированных технических устройствах. В качестве алгебраической структуры для криптографических преобразований с помощью данного протокола может быть также выбрана группа дивизоров гиперэллиптической кривой, как это было показано, например, в работе [7].

Литература.

[1] ДСТУ 4145-2002. Державний стандарт України. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що грунтується на еліптичних кривих. Формування та перевірка. Київ:-Держстандарт України, 2003.

[2] ГОСТ Р 34.10-2001. Государственный стандарт российской федерации. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки цифровой подписи. М.: Госстандарт России, 2001.

[3] *Don Johnson, Alfred Menezes, Scott Vanstone.* The Elliptic Curve Digital Signature Algorithm (ECDSA)// Certicom Research, Canada, 2001.

[4] *И.Д. Горбенко, С.И. Збитнев, А.А. Поляков.* Сравнительный анализ ЦП в группах точек эллиптических кривых // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002.

[5] *М.Ф. Бондаренко, Ю.І. Горбенко, С. С. Батюшко.* Аналіз захищеності існуючих ЦП від атак на зв'язаних ключах. Прикладная радиоэлектроника. Том 5, № 1. 2006. С. 52-58.

[6] *М.Ф. Бондаренко, Ю.І.Горбенко, С.С. Батюшко.* Аналіз захищеності існуючих ЦП від атак на реалізацію. Прикладная радиоэлектроника. Том 5, № 1. 2006. С. 59-61.

[7] *Неласая А.В.* Протокол цифровой подписи на гиперэллиптических кривых// Радіоелектроніка. Інформатика. Управління. №1. – Запоріжжя, 2006.

Поступила в редколлегию 22.09.2009

Долгов Виктор Иванович, доктор технических наук, профессор кафедры БИТ ХНУРЭ. Область научных интересов: математические методы защиты информации.

Неласая Анна Викторовна, ассистент кафедры программных средств ЗНТУ, соискатель кафедры БИТ ХНУРЭ. Область научных интересов: протоколы цифровой подписи и направленного шифрования на эллиптических кривых.

Погорелый Анатолий Николаевич, начальник КБ ГП «Радиоприбор», г. Запорожье.