УДК 681.3.06:519.248.681

И.Д. ГОРБЕНКО, д-р техн. наук, А.А. ПОЛЯКОВ, С.И. ЗБИТНЕВ

ПРОТОКОЛЫ – ПРИМИТИВЫ УПРАВЛЕНИЯ КЛЮЧАМИ В ГРУППАХ ТОЧЕК ЭЛЛИПТИЧЕСКИХ КРИВЫХ

Введение

В защищенных информационных технологиях определяющей процедурой является процедура выработка общего секрета (ключа). Для решения этой задачи разработаны и применяются различной степени сложности состоятельные протоколы, прежде всего Диффи-Хелмана, реализованные на основе преобразований в полях Галуа [1]. Однако развитие методов и средств криптоанализа таких криптосистем и криптопротоколов вынуждает увеличивать размеры общесистемных параметров и ключей, вследствие чего увеличивается сложность выполнения базовых операций в полях, в ряде случаев до недопустимых величин. Разрешение этого противоречия может быть достигнуто за счет выработки общего секрета в группах точек эллиптических кривых над полем Галуа $GF(q)$. Целью данной статьи есть рассмотрение основных проблемных задач реализации состоятельных протоколов, определение и анализ базовых параметров протоколов и условий их реализации с использованием преобразований в группах точек эллиптических кривых.

1. Основные понятия

В зависимости от приложений существует несколько определений протокола. Наиболее приемлемым на наш взгляд есть следующее. Протокол – это распределенный алгоритм решения некоторой совокупности объектов и субъектов любой задачи, каждый из которых достигает цели (решает задачу) с использованием частных (распределенных) алгоритмов, причем при выполнении распределенных алгоритмов все объекты и субъекты используют одинаковую спецификацию данных и действий, процедуры синхронизации и восстановление работы после сбоев и др. Можно сказать, что протокол это многосторонний алгоритм, заданный последовательностью шагов, точно и однозначно описывающий действия двух или более сторон (объектов) которые должны быть выполнены для достижения частных и общих целей. Особенностью криптографического протокола есть то, что при его выполнении с целью обеспечения конфиденциальности, целостности, наблюдаемости и доступности до информации и/или ресурсов используются криптографические преобразования [6].

Для согласованного выполнения криптографических преобразований все взаимодействующие объекты и субъекты должны выполнять процедуру установления ключей [7]. Под установлением ключей понимается процесс или протокол, посредством выполненная которого общий секрет становится доступным объектам и/или субъектам системы (технологии), что позволяет им выполнять криптографические преобразования с необходимым качеством.

Установления ключей может быть четко разделено на передачу (транспортировку) ключей и согласование ключей [2]. При передачи ключей один объект или субъект создает или получает соответствующим образом секретное значение ключа и затем передает его другим объектам и/или субъектам безопасным образом т.е. с обеспечением конфиденциальности, целостности, подлинности, доступности и наблюдаемости. При выполнении протокола согласования ключей общий секрет вырабатывается двумя или более объектами или субъектами как функция информации, связанная с каждым из них. По сути протокол согласования ключей представляет собой процедуру разделения секрета, при реализации которой только $i > t$ из n объектов и субъектов могут совместно выработать общий секрет.

Протоколы установления ключей, включающие аутентификацию, обычно требуют фазы настройки, посредством которой осуществляется распределение подлинных и возможно секретных начальных ключевых данных. Большинство протоколов имеют своей целью создание

различных ключей при каждом выполнении протокола. В некоторых случаях начальные ключевые данные задают фиксированный ключ, который каждый раз будет приводить к выполнению протокола данной парой или группой пользователей. Системы, использующие такие статические ключи, являются незащищенными от атак с известным ключом.

Многие протоколы установления ключей требуют участия централизованной или доверенной стороны для начальной системной настройки или для *интерактивных* действий (то есть в реальном времени), либо для обеих целей. Доверенная сторона обычно называется различными именами, в зависимости от выполняемых ею функций, например, *доверенная третья сторона*, *доверенный сервер*, *аутентификационный сервер*, *центр распределения ключей* (ЦРК), *центр преобразования ключей* (ЦПК) и *сертификационный орган* (СА).

К протоколам предъявляются требования, чтобы каждая из сторон при установлении ключей могла определять истинную подлинность другой(-их) стороны, что бы предотвратить несанкционированное использование результирующего ключа. В этом случае считается, что метод обеспечивает *безопасное установление ключей*. Это требуется как для секретности ключа, так и для идентификации сторон, выполняющих доступ к нему. Кроме того, требование идентификации сторон несколько, но очень важным образом, отличается от требований аутентификации объекта – здесь требованием является скорее знание подлинности сторон, которые могут получить доступ к ключу, чем подтверждение факта установления фактической связи с участием таких сторон.

Аутентификация ключей – свойство, дающее одной стороне уверенность в том, что никакая другая сторона, кроме конкретной второй стороны, не сможет получить доступ к конкретному секретному ключу. Аутентификация ключей не зависит от фактического владения таким ключом второй стороной или осведомленности первой стороны о таком фактическом владении. Фактически она не требует вообще никаких действий от второй стороны. По этой причине иногда используется более точное ее название (*неявная*) *аутентификация ключей*.

Подтверждение ключей – свойство, дающее уверенность одной стороне в том, что вторая сторона на самом деле владеет конкретным секретным ключом.

Явная аутентификация ключей – свойство, использование которого позволяет осуществлять аутентификацию и подтверждение ключей.

Главное внимание в процедуре аутентификации ключей сосредотачивается на подлинности второй стороны, а при подтверждении ключей – на знании значения обратного ключа. Подтверждение ключей базируется на участии объекта или субъекта, принимающего сообщение в получении и демонстрации факта владения этим ключом.

На практике факт владения ключом может быть подтвержден различными способами, включая создание односторонней хэш-функции самого ключа, использование ключа в ключевой хэш-функции и шифрование известных данных с использование этого ключа. При этом может раскрываться некоторая информация относительно значения самого ключа. Методы, использующие протоколы с нулевым знанием, позволяют подтверждать факт владения ключом, не давая при этом никакой дополнительной информации относительно его значения.

Протокол *установления аутентифицированных ключей* является протоколом установления ключей, который обеспечивает аутентификацию ключей.

Анализ показывает, что в протоколе установления ключей, который включает также и аутентификацию объектов, взаимодействие необходимо построить так, чтобы можно было давать гарантии, в том, что сторона, подлинность которой подтверждена, является той же самой стороной, с которой устанавливается ключ. Если это не обеспечивается, то криптоаналитик может осуществить ложную аутентификацию, а затем имитировать ее в протоколе установлении ключей.

Криптографические протоколы, включающие обмены сообщениями, требуют точного определения как используемых при этом сообщений, так и действий, предпринимаемых ка-

ждой стороной. На основании указанных целей можно выделить следующие протоколы аутентификации, протоколы установления ключей и протоколы аутентифицированных ключей.

Протокол аутентификации – дает одной стороне некоторую степень гарантии относительно подлинности другой стороны, с которой она намерена вести информационный обмен.

Протокол установления ключей – устанавливает общий секрет с целью осуществления в дальнейшем защищенного информационного обмена.

Протокол установления аутентифицированных ключей – устанавливает общий секрет со стороной, чья подлинность была (или может быть) подтверждена.

Протоколы установления ключей используются для создания общих секретов, которые обычно называются *сеансовыми ключами* или используются для их получения. Идеально сеансовый ключ является *временным секретом*, то есть секретом, использование которого ограничено коротким периодом времени. Например, единственное телекоммуникационное соединение, после которого сеанс разрывается. Использование сеансовых ключей объясняется следующими причинами [3]:

- необходимостью ограничения объема шифротекста (зашифрованного на фиксированном ключе), который может использоваться для выполнения криптоаналитической атаки;
- необходимостью ограничения скомпрометированных данных по периоду времени и количеству данных в случае компрометации ключа;
- необходимостью отказа от долговременного хранения большого количества различных секретных ключей, например в случае, когда один терминал обменивается сообщениями с большим числом других терминалов посредством создания ключей только когда это действительно требуется;
- необходимостью создания сеансов или приложений, независимых на протяжении всех коммуникационных передач.

При проектировании или выборе метода установления ключей для использования важно учитывать требуемые гарантии и свойства для предполагаемого применения. Необходимо делать различия между функциональностью, обеспечиваемой для пользователя, и техническими характеристиками, отличающими механизмы на уровне реализаций. Характеристики, отличающие методы установления ключей, включают:

1. *Характер* аутентификации, под которой понимается возможность аутентификации объектов, ключей и подтверждение ключей.
2. *Взаимность* аутентификации, при которой аутентификация объектов и аутентификация ключей или подтверждение ключей может обеспечиваться для обеих сторон.
3. *Новизна ключей*. Ключ считается *новым*, если он отличается от ранее использованного ключа или использованных ключей.
4. *Управление ключами*. Под управлением понимается процедура выработки, распределения, хранения, передачи, приема, ввода, использования и уничтожения ключей.
5. *Эффективность*. При оценке эффективности учитывается:
   a) число обменов сообщениями между взаимодействующими сторонами;
   b) ширина канала пропускания, требуемая для сообщений (объем передаваемых сторонами данных);
   c) сложность вычислений, выполняемых каждой из сторон;
   d) возможность предварительных вычислений, выполняемых с целью уменьшения *интерактивной* вычислительной сложности.
6. *Требования к третьей стороне*. Основными из этих требований являются:
   a) требование *интерактивного* (в реальном времени), автономного участия третьей стороны или без участия третьей стороны;
   b) установление требуемой степени доверия к третьей стороне.

7. *Тип используемого сертификата,* если он используется. Под ним подразумевают способ распределения долговременных ключей, подлинность и целостность которых подтверждается третей стороной.
8. *Неоспоримость авторства.* Протокол обеспечивает возможность доказательства причастности объектов или субъектов к процедуре выполнения протокола.

2. Основные алгоритмы выработки общего секрета

Анализ показывает, что в основе протоколов управления ключами лежат два математических алгоритма вычисления общего секретного значения – простой алгоритм Диффи-Хелмана (ДХ) [9] и сложный алгоритм (MQV) [1, 2, 11]. Простой алгоритм ДХ обеспечивает выработку общего секрета на основе знаний одного личного ключа d, используемого, как правило, многократно.

В алгоритме MQV один из ключей, например d_2, является сеансовым. Функция $avf(Q)$ определяет преобразованное (связанное) значение точки P. Несмотря на повышенную, по сравнению с алгоритмом ДХ, вычислительную сложность, алгоритм MQV в большинстве случаев является более предпочтительным. Он позволяет вырабатывать на каждый сеанс или файл сеансовый ключ, что обеспечивает защиту от компрометации ключей и осуществления криптоаналитических атак. В табл. 1 приведено описание алгоритмов выработки общего секрета ДХ и MQV для абонентов A и B.

Таблица 1

Алгоритм Диффи-Хелмана	*Алгоритм MQV*
• d_A – личный ключ объекта A; • Q_B – открытый ключ объекта B.	• Две пары ключей, долговременный $\{d_{1,A}, Q_{1,A}\}$ и сеансовый $\{d_{2,A}, Q_{2,A}\}$, принадлежащих объекту A. • Два открытых ключа $Q_{1,B}$ и $Q_{2,B}$, принадлежащих объекту B.
1. Вычислить точку $P=(x,y)=d_A\times Q_B$. 2. Проверить $P\neq O$, где O – ноль аддитивной группы (точка бесконечности). Если $P=O$, то вывести "недостоверный" и остановка. 3. Установить $Z=\pi(x_P,y_P)$, где π – функция преобразования координаты точки. В простейшем случае $Z=x_P$, $\pi(x_P,y_P)=x_P$.	1. Вычислить целое число: $r=d_{2,A}+(\mathrm{avf}(Q_{2,A})\times d_{1,A})(\mathrm{mod}\,n)$, где n – порядок базовой точки G на ЭК. 2. Вычислить точку на эллиптической кривой: $P=h\times r\times(Q_{2,B}+(\mathrm{avf}(Q_{2,B})\times Q_{1,B}))$, где h коэффициент связи порядка ЭК u и порядка базовой точки n. 3. Проверить $P\neq O$. Если $P=O$, то вывести "недостоверный" и остановка. 4. Установить $Z=x_P$, где x_P – x-координата точки P.

Из предварительного анализа приведенных алгоритмов видно, что вычислительно более сложным является алгоритм *MQV*. При выполнении алгоритма *MQV* необходимо выполнить как минимум две вычислительно сложных операции, определение значения r и вычисление точки P, используя операцию скалярного умножение. В алгоритме Диффи-Хелмана выполняется только одного скалярное умножение.

3. Основные требования к протоколам

Проведенный анализ показал [2], что протоколы управления ключами можно разделить на два класса:

- *протоколы согласования ключей*, задачей которых является выработка общего секрета (секретного ключа) на основе известных открытых ключей объектов;
- *протоколы транспортировки ключей*, задачей которых является доставка, ввод в действие и использование ключей с обеспечением их целостности, подлинности и при необходимости, конфиденциальности.

Названные протоколы по своему функциональному предназначению очень похожи, поэтому на них как правило накладываются одни и те же функциональные требования. Вместе с тем существуют различия в методах и, как следствие, средствах их реализации.

В практическом аспекте основным требованием к рассматриваемым протоколам является требование их состоятельности в смысле обеспечения целостности, подлинности, конфиденциальности, доступности и наблюдаемости ключей на всех этапах их жизненного цикла. В теоретическом смысле каждый из протоколов должен обладать свойствами полноты, корректности, а также в некоторых случаях свойством нулевого разглашения знаний.

Представляет интерес рассмотрение протоколов, реализуемых в группах точек эллиптических кривых с позиции их состоятельности, целостности, подлинности ключей и параметров, что может обеспечиваться за счет включения в протокол и выполнения при каждом обращении к ключам и параметрам дополнительных алгоритмов проверки, которые строятся на основе математических свойств ключей и параметров. Такие дополнительные проверки позволяют защититься от ряда угроз, прежде всего:

- *подмены параметров эллиптической кривой;*
- *преднамеренного или вынужденного использования слабых кривых;*
- *несоответствия и несогласованности открытых ключей с параметрами эллиптической кривой.*

Проведенный анализ показал, что в существующих протоколах управления ключами выработка конкретного значение секретного ключа производится из общего секрета посредством использования специальных функций, обозначаемых как *kdf* .

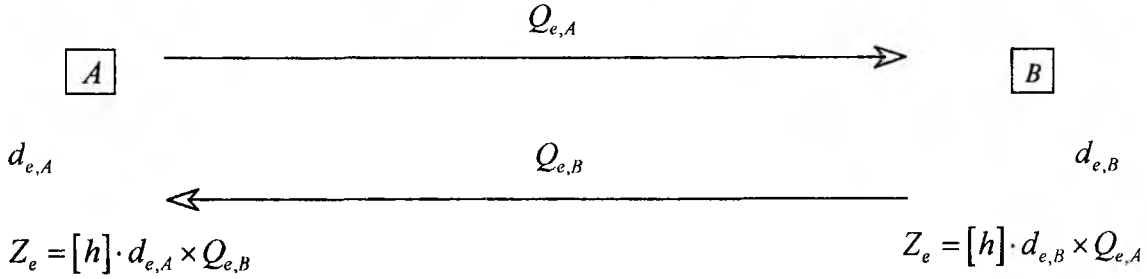
Поэтому важным является определение требований к таким функциям и выполнение этих требований.

4. Стандартные протоколы согласования ключей

Рассмотрим основные стандартные протоколы согласования ключей с целью их классификации и анализа. При этом выделим два объекта, один из которых является источником (инициатором), а другой ответчиком (приемником). В таком протоколе ключи состоят из пары сеансовых ключей.

Протокол 1. Сеансовый протокол согласования ключей

Секретными являются ключи $d_{e,A}$ и $d_{e,B}$, открытые $Q_{e,A}$ и $Q_{e,B}$.



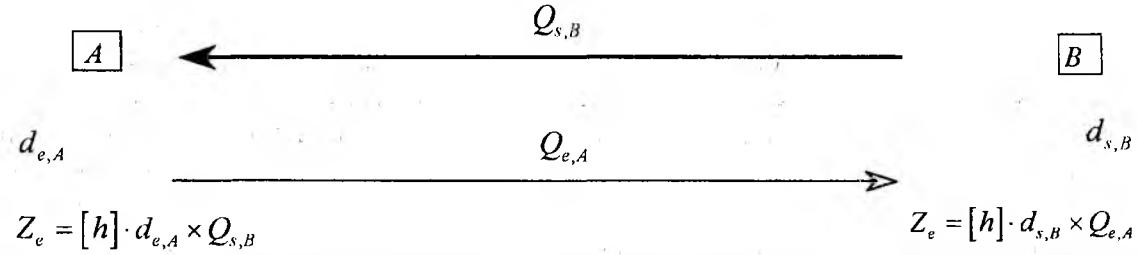
Общим секретом является Z_e, вырабатываемый пользователями A и B, где h-кофактор. Конкретное секретное значение вырабатывается как

$$\textit{ключ} = \mathrm{kdf}\left(Z_e\right).$$

Таким образом, протокол реализует выработку сеансовых пар ключей и производит обмен открытыми сеансовыми ключами, на основании которых вырабатывается общее секретное значение.

Протокол 2. Однопроходной протокол Диффи-Хелмана

В протоколе используются две пары ключей $\{d_{e,A}, Q_{e,A}\}$ и $\{d_{s,B}, Q_{s,B}\}$, одна из которых является сеансовой, другая главной. Открытый главный ключ $Q_{s,B}$ передается заранее.



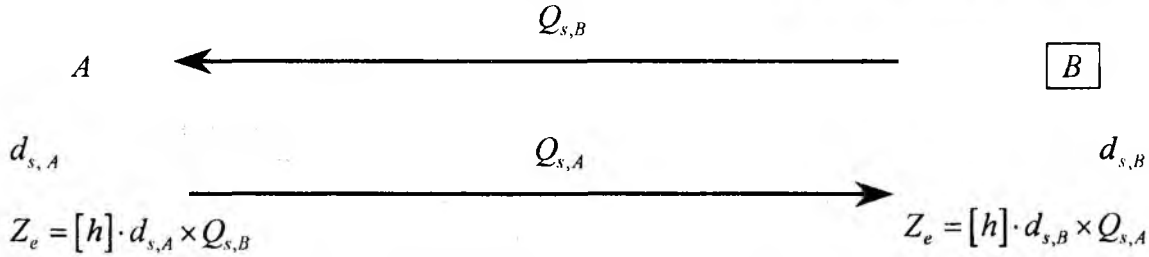
Жирными линиями выделены те передачи ключей, которые не участвуют в протоколе, они были произведены заранее. Их илюстрация является информативной.

В протоколе выполняется только одна передача сеансового ключа $Q_{e,A}$. Значение секретного ключа вырабатывается как

$$ключ = \mathrm{kdf}(Z_e).$$

Протокол 3. Протокол на главных ключах

В протоколе используются только главные пары ключей $\{d_{s,A}, Q_{s,A}\}$ и $\{d_{s,B}, Q_{s,B}\}$.

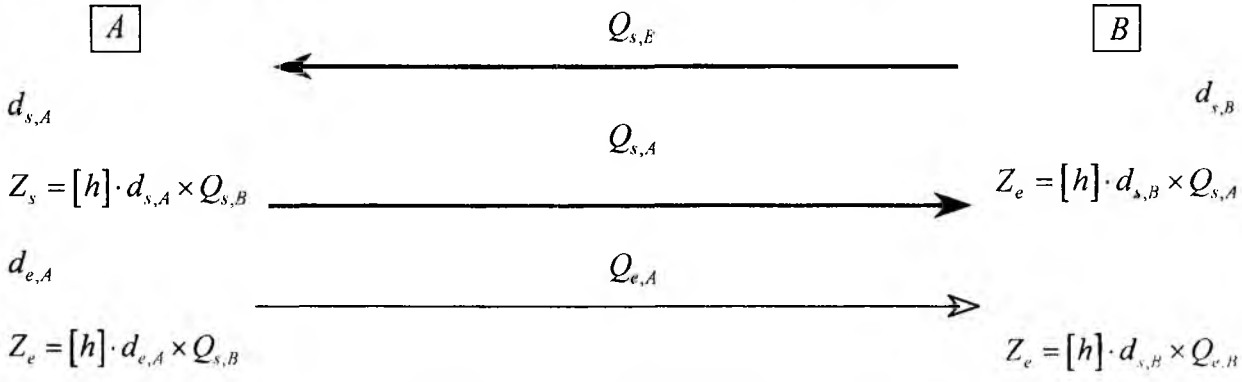


Протокол выполняет вычисление общего секрета, основываясь только на главных ключах, что позволяет не производить передачи открытых ключей во время выполнения протокола. Секретный ключ вычисляется с использованием следующей функции:

$$ключ = \mathrm{kdf}(Z_e).$$

Протокол 4. Однопроходной протокол с использованием главных ключей

Ключевой материал, использующийся в протоколе, состоит из двух пар главных ключей $\{d_{s,A}, Q_{s,A}\}$, $\{d_{s,B}, Q_{s,B}\}$ и одной пары сеансовых ключей $\{d_{e,A}, Q_{e,A}\}$.

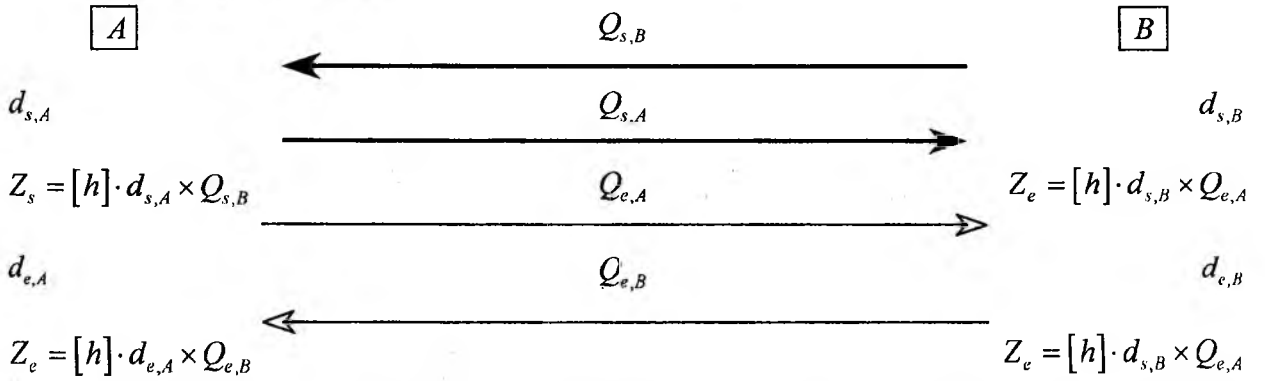


Протокол является усиленной версией однопроходного протокола ДХ. В протоколе выполняется одна передача открытого сеансового ключа. На основании используемого ключевого материала вычисляется два общих секретных значения, Z_s и Z_e. Вычисление секретного ключа осуществляется на основании этих значений, вначале выполняется операция конкатенация (операция объединения) над значениями Z_s и Z_e, а затем вычисляется секретный ключ

$$ключ = \mathrm{kdf}(Z_s \,||\, Z_e).$$

Протокол 5. Полный протокол согласования ключей

Данный протокол является полным протоколом согласования ключей. Ключевой материал состоит из двух пар главных ключей и $\{d_{s,A}, Q_{s,A}\}$, $\{d_{s,B}, Q_{s,B}\}$ и двух пар сеансовых ключей $\{d_{e,A}, Q_{e,A}\}$, $\{d_{e,B}, Q_{e,B}\}$.



В данном протоколе выполняется две передачи сеансовых открытых ключей. С использованием сеансовых и долговременных открытых и личных ключей, формируемых общие секретные значения Z_s и Z_e. Вычисление секретного ключа производится по формуле

$$ключ = \mathrm{kdf}(Z_s \| Z_e).$$

Приведенные выше базовые протоколы-примитивы являются основными для формирования прикладных протоколов:

На рис. 1 в качестве примера приведена схема, реализующая полный протокол согласования ключей. Данный протокол является полным протоколом согласования ключей с подтверждением целостности и аутентификацией ключей. Ключевые данные состоят из двух пар главных ключей $\{d_{s,A}, Q_{s,A}\}$, $\{d_{s,B}, Q_{s,B}\}$ и двух пар сеансовых ключей $\{d_{e,A}, Q_{e,A}\}$, $\{d_{e,B}, Q_{e,B}\}$, а также ключей цифровой подписи $\{d_{sig,A}, Q_{sig,A}\}$ [3].

5. Стойкость криптопротоколов

Криптопротоколы, использующие криптопреобразования в группах точек ЭК, должны обеспечивать требуемую стойкость, т.е. относятся к классу теоретически стойких или доказуемо стойких алгоритмов.

Алгоритм согласования ключей Диффи-Хеллмана реализует скалярное умножение точки, которое возможно реализовать в рамках математического аппарата ЭК. Для компрометации протоколов установления ключей необходимо произвести обратную операцию к скалярному умножению, т.е. решить задачу дискретного логарифма. Таким образом, криптографическая стойкость протоколов основана сложности решения задачи ДЛЭК[4] с известной наилучшей атакой ρ-метод Полларда. В табл. 2 приведены числовые характеристики.

Таблица 2

Длина в битах	Количество операций в группе.
128	$1{,}63480 \times 10^{19}$
192	$7{,}02141 \times 10^{28}$
256	$3{,}01567 \times 10^{38}$
320	$1{,}29522 \times 10^{48}$
352	$8{,}48836 \times 10^{52}$
384	$5{,}56293 \times 10^{57}$
416	$3{,}64572 \times 10^{62}$
448	$2{,}38926 \times 10^{67}$
480	$1{,}56583 \times 10^{72}$
512	$1{,}02618 \times 10^{77}$
1024	$1{,}18824 \times 10^{154}$

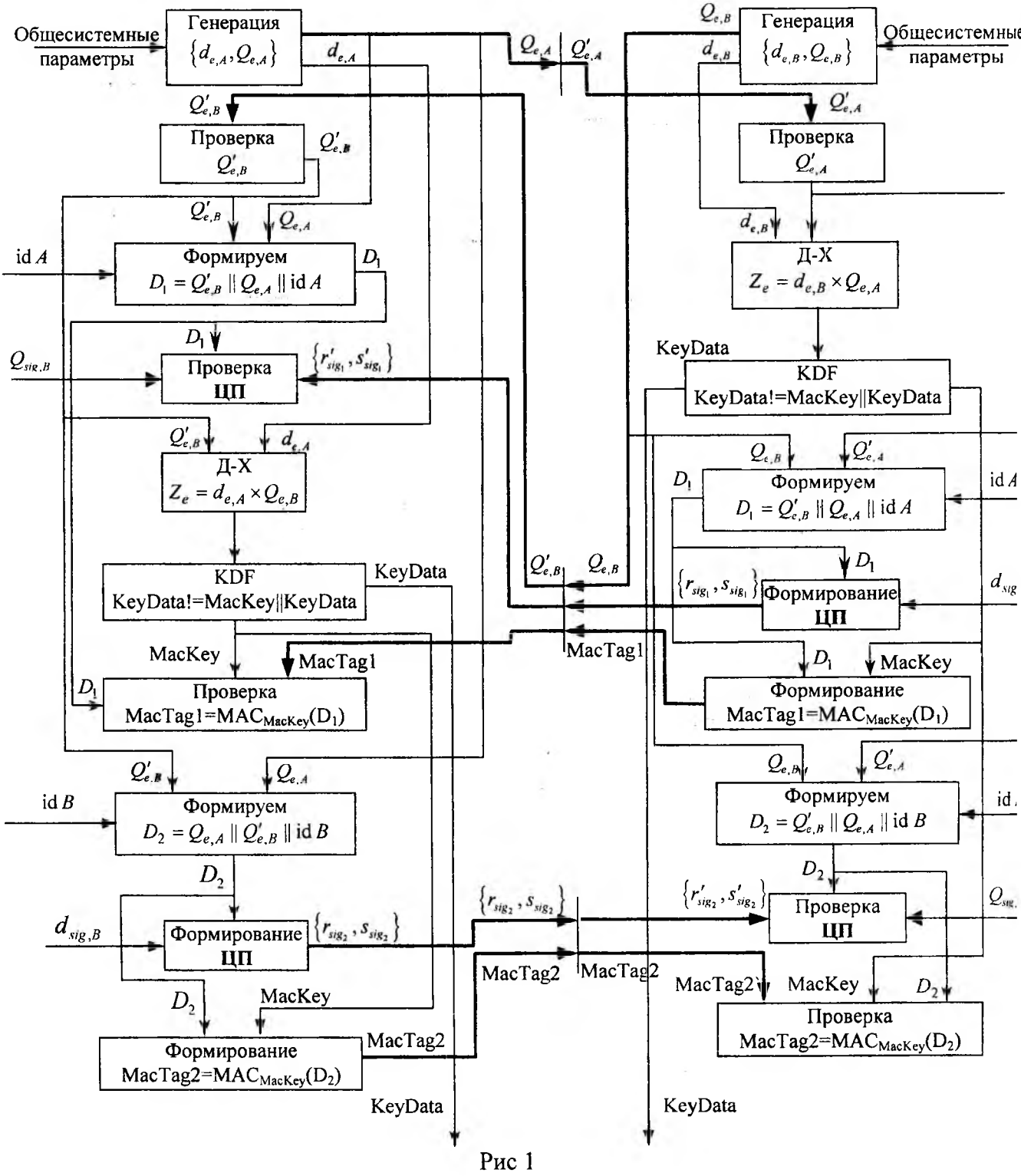


Рис 1

6. Требования к длине ключа

Задача схем установления ключей состоит в установлении секретных ключевых данных, разделяемых двумя объектами. Сложность атаки схемы (протокола) установления ключей должна быть не меньше сложности атаки полного перебора ключей. То есть, когда кто-нибудь устанавливает симметричные ключи, он хочет иметь гарантию, что схема установления ключей будет иметь ту же криптоаналитическую сложность, что и для симметричного алгоритма.

Такое основное условие должно быть выполнено при выборе размера параметров ЭК. Это требование связано с тем, что условие $n > 2^{160}$ в настоящее время является недостаточным для обеспечения защиты, оно не дает требуемого уровня защиты для 256-битного ключа симметричного шифрования. Поэтому минимальным рекомендуемым значением является $n \geq 2^{512}$ [2, 5].

В практически применяемых системах обеспечивается вычислительная сложность. Так в [2] определены требования к длине ключей криптопреобразований для симметричных криптоалгоритмов. Они заключаются в том, что для первого класса сложности, в симметричных криптоалгоритмах, длина ключа должна быть не менее 256 бит, 2-го – 128 бит и 3-го – 128 бит.

При решении данной проблемы необходимо знать сложность наилучшей криптоаналитической атаки на эллиптические кривые. Наилучшей атакой на ЭК в настоящее время считается алгоритм ρ-Полларда [4]. Его сложность можно оценить количеством операций сложения на эллиптической кривой

$$I_{ЭК} = \sqrt{\frac{\pi \cdot n}{4}}.$$

Таблица 3

Название симметричного криптоалгоритма	Длина ключа симметричного криптоалгоритма, бит	Длина модуля преобразования ЭК, бит
DEA	56	112
2-ключевой 3-DES	112	224
RIJNDAL	128	256
3-ключевой 3-DES	168	336
RIJNDAL	192	384
RIJNDAL	256	512

В таб. 3 приведены длины модулей преобразований в группах точек эллиптической кривой, при которых обеспечивается такая же стойкость ключа, как и в симметричных криптоалгоритмах.

7. Общесистемные параметры ЭК

Для использования эллиптических преобразований в криптографии необходимо иметь средства генерации общесистемных параметров: параметров эллиптической кривой a, b, поле F_q, над которым определена кривая, порядок эллиптической кривой $u = \#E(F_q)$, кофактор h, базовая точка G, порядок базовой точки $n = u/h$, в случае $q = 2^m$ примитивный полином $f(x)$. В настоящее время используются следующие алгоритмы формирования общесистемных параметров [1-2].

- Через подполя, случай когда $q = 2^{de}$, d и e целые числа.
- Комплексное умножение.
- Алгоритм «малых» и «больших» шагов.
- Алгоритм Скуфа.
- Построение общесистемных параметров через якобиан.

Существует два способа распространения параметров:

- Централизованное, через центр управления и сертификации ключей
- Каждая группа пользователей генерирует общесистемные параметры себе.

Учитывая то, что генерация общесистемных параметров является трудоемком процедурой, вариант централизованной поставки позволяет существенно уменьшить вычислительные и временные затраты при смене общесистемных параметров. Кроме того сертификационный центр или центральный орган распространения общесистемных параметров (ОП) имеет возможность заранее сформировать ОП. Недостатком централизованной рассылки ОП является необходимость доверительных отношений всех участников, использующих распространяемые параметры. Если отсутствует уверенность корректности полученных ОП, пользователь может провести ряд проверочных процедур:

- Проверить условие MOV [10].
- Сотсутствие аномальности кривой $\#E(F_q) \neq q$.

- Проверить, что для $q = 2^m$ $b \neq 0$, $q = p$ $4a^3 + 27b^2 \neq 0 \bmod p$.
- Порядок базовой точки n простой и $n > 4\sqrt{p}$.
- $p - 2\sqrt{p} \leq \#E(F_q) \leq p + 2\sqrt{p}$.
- $nG = O$.
- Координаты точки G удовлетворяют неравенству $y^2 + xy = x^3 + ax^2 + b$ при $q = 2^m$ или $y^2 = x^3 + ax + b$ при $q = p$.

В случае недоверия сертификационному центру группа пользователей может организовать свой центр распределения и генерации общесистемных параметров. Ниже приведены некоторые общесистемные параметры, удовлетворяющие вышеуказанным требованиям:

Параметры для $F(2^m)$:

m =193, $f(x) = x^{193} + x^{15} + 1$

a =0017858FEB7A98975169E171F77B4087DE098AC8A911DF7B01,
b = 00FDFB49BFE6C3A89FACADAA7A1E5BBC7CC1C2E5D831478814,
$G(x, y)$ = 0301F481BC5F0FF84A74AD6CDF6FDEF4BF6179625372D8C0C5E1 – сжатая форма
$G(x, y)$ = 401F481BC5F0FF84A74AD6CDF6FDEF4BF6179625372D8C0C5E10025E399F29037\
12CCF3EA9E3A1AD17FB0B3201B6AF7CE1B05,
n = 010000000000000000000000C7F34A778F443ACC920EBA49,
h = 02.

m =283, $f(x) = x^{283} + x^{12} + x^7 + x^5 + 1$

a = 000000000000000000000000000000000000000000000000000000000000000000000000000001,
b = 027B680AC8B8596DA5A4AF8A19A0303FCA97FD7645309FA2A581485AF6263E313B79\
A2F5,
$G(x, y)$ = 0305F939258DB7DD90E1934F8C70B0DFEC2EED25B8557EAC9C80E2E198F8CDB\
ECD86B12053 – сжатая форма,
$G(x, y)$ = 0405F939258DB7DD90E1934F8C70B0DFEC2EED25B8557EAC9C80E2E198F8CDB\
ECD86B1205303676854FE24141CB98FE6D4B20D02B4516FF702350EDDB0826779\
C813F0DF45BE8112F4,
n = 03FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEF90399660FC938A90165B042A7CEFAD\
B307,
h = 02.

m =571, $f(x) = x^{571} + x^{10} + x^5 + x^2 + 1$

a = 000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000001,
b = 02F40E7E2221F295DE297117B7F3D62F5C6A97FFCB8CEFF1CD6BA8CE4A9A18AD84F\
FABBD8EFA59332BE7AD6756A66E294AFD185A78FF12AA520E4DE739BACA0C7FFE\
FF7F2955727A,
$G(x, y)$ = 030303001D34B856296C16C0D40D3CD7750A93D1D2955FA80AA5F40FC8DB7B2\
ABDBDE53950F4C0D293CDD711A35B67FB1499AE60038614F1394ABFA3B4C850
D927E1E7769C8EEC2D19 – сжатая форма,
$G(x, y)$ = 040303001D34B856296C16C0D40D3CD7750A93D1D2955FA80AA5F40FC8DB7B2\
ABDBDE53950F4C0D293CDD711A35B67FB1499AE60038614F1394ABFA3B4C850
D927E1E7769C8EEC2D19037BF27342DA639B6DCCFFFEB73D69D78C6C27A6009\
CBBCA1980F8533921E8A684423E43BAB08A576291AF8F461BB2A8B3531D2F048\
5C19B16E2F1516E23DD3C1A4827AF1B8AC15B,

n = 03FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF\
FFFE661CE18FF55987308059B186823851EC7DD9CA1161DE93D5174D66E8382E9BB2F\
E84E47,

h = 02.

Параметры для $F(p)$:

$|p| = 192$,

p = FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFEE37,

a = 000000000000000000000000000000000000000000000000,

B 000000000000000000000000000000000000000000000003,

$G(x, y)$ = 03DB4FF10EC057E9AE26B07D0280B7F4341DA5D1B1EAE06C7D – сжатая форма,

$G(x, y)$ = 04DB4FF10EC057E9AE26B07D0280B7F4341DA5D1B1EAE06C7D9B2F2F6D9C562\
8A7844163D015BE86344082AA88D95E2F9D,

n = FFFFFFFFFFFFFFFFFFFFFFE26F2FC170F69466A74DEFD8D,

h = 01.

$|p| = 256$,

p = FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFC2F,

a = 000000000000000000000000000000000000000000000000000000000000000000,

b = 000000000000000000000000000000000000000000000000000000000000000007,

$G(x, y)$ = 0279BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81\
798 – сжатая форма,

$G(x, y)$ = 0479BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81\
798483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10\
D4B8,

n = FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEBAAEDCE6AF48A03BBFD25E8CD0364141,

h = 01.

$|p| = 521$,

p = 01FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF\
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF,

a = 01FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF\
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC,

b = 0051953EB9618E1C9A1F929A21A0B68540EEA2DA725B99B315F3B8B489918EF109E15\
6193951EC7E937B1652C0BD3BB1BF073573DF883D2C34F1EF451FD46B503F00,

$G(x, y)$ = 0200C6858E06B70404E9CD9E3ECB662395B4429C648139053FB521F828AF606B4D\
3DBAA14B5E77EFE75928FE1DC127A2FFA8DE3348B3C1856A429BF97E7E31C2E\
5BD66,

$G(x, y)$ = 0400C6858E06B70404E9CD9E3ECB662395B4429C648139053FB521F828AF606B4D\
3DBAA14B5E77EFE75928FE1DC127A2FFA8DE3348B3C1856A429BF97E7E31C2E\
5BD66011839296A789A3BC0045C8A5FB42C7D1BD998F54449579B446817AFBD1\
7273E662C97EE72995EF42640C550B9013FAD0761353C7086A272C24088BE94769\
FD16650,

n = 01FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFA\
51868783BF2F966B7FCC0148F709A5D03BB5C9B8899C47AEBB6FB71E91386409,

h = 01.

8. Вывод

Использование протоколов установления и выработки ключей в группах точек эллиптической кривой позволяет согласованно выработать ключи и обеспечить функцию причастности. Использование преобразований в группах точек ЭК по сравнению с преобразованиями в кольцах и полях [3] позволяет в 4 – 6 и более раз сократить длины открытых ключей и общесистемных параметров или при тех же параметрах существенно повысить стойкость.

Используемые на практике состоятельные протоколы, реализуемые за счет преобразований в кольцах и полях, являются состоятельными и при использовании в группах точек эллиптических кривых.

Следует ожидать, что в ближайшие годы при реализации состоятельных протоколов будут использоваться алгоритмы направленного шифрования, цифровой подписи и выработки ключей, построенные на основе преобразований в группах точек эллиптической кривой.

Список литературы: 1. X9.42 Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Algorithm Keys Using Diffie-Hellman, 1996. Working Draft. 2. X9.63 Public Key Cryptograph For The Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography. 1999. 207 c. 3. IEEE P1363 / D9 (Draft Version 9). Standard Specifications for Public Key Cryptography. Number-Theoretic Background. 1999. 4. *И.Д. Горбенко, С.И. Збитнев, А.А. Поляков* Криптографические преобразования в группах точек эллиптических кривых методом Полларда // Радиотехника: Всеукр. межвед. науч-тех. сб 2001. Вып. 119. С. 43-50. 5. ISO/IEC CD 15946-3 Information Technology – Security Techniques – Cryptographic Techniques Based on Elliptic Curves – Pert 3http://crypto.nessie.org. 6. *A. Menezes, P. van Orschot, S. Vanstone* Handbook of Applied Cryptography. CRC Press, 1997. 7. *S. Blake-Wilson, D. Johnson, A. Menezes* Key agreement protocols and their Security Analysis. 1997. 8. M. Bellare, R. Canetti, H. Krawczyk. Keying hash functions for message authentication. In *Advances* in Cryptology: Crypto `96, v 1-15, 1996. 9. *Diffie W., Hellman M.E.* New Direction in Cryptography / *IEEE Trans. Inf. Theory.*, Nov. 1976, IT-22, 644-654. 10. *A.J. Menezes, T. Okamoto, and S.A. Vanstone.* Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39, pages 1639-1646, 1993. 11. *A.J. Menezes, M. Qu, and S.A. Vanstone.* Some new key agreement protocols providing implicit authentication. Workshop record, *2nd Workshop on Selected Areas in Cryptography (SAC '95)*, Ottawa, Canada, May 18-19, 1995.

Харьковский национальный университет радиоэлектроники

Поступила в редколлегию 22.04.2002