УДК 519.719.325 DOI 10.17223/20710410/61/4

СИСТЕМЫ С ОТКРЫТЫМИ КЛЮЧАМИ НА ОСНОВЕ ИДЕНТИФИКАЦИОННОЙ ИНФОРМАЦИИ

А. В. Черемушкин

Академия криптографии РФ, г. Москва, Россия

E-mail: avc238@mail.ru

Рассмотрены особенности практического применения криптографических систем с открытыми ключами на основе идентификаторов. Выделены математические задачи и конструкции, приведены основные подходы к построению систем шифрования, цифровой подписи, аутентификации сторон и ключевых систем с открытыми ключами на основе идентификаторов.

Ключевые слова: *криптография на основе идентификаторов, криптосистемы с открытыми ключами, эллиптическая кривая, билинейное спаривание.*

ID-BASED PUBLIC KEY CRYPTOGRAPHIC SYSTEMS

A. V. Cheremushkin

Academy of Cryptography of the Russian Federation, Moscow, Russia

A surway contains an analisys of practical aspects of ID-based public key cryptography. IDB-systems simplify a certificate management process, but trusted requirements for the key generation center (KGC) must be very stronger than for certification authority. When key escrow property is not assumed, users' private keys should be protected from malicious KGC. Many networks need hierarchical KGC architecture. In the paper, we describe a basic mathematical constructions applied in ID-based cryptosystems. We survey fundamental ID-based cryptographic primitives: Key extraction, Key Escrow, Encryption, Digital Signature, Identification Scheme and Key Agreement, which are based on the mathematical concepts of Integer Factorization, Quadratic Residues, Discret Logarithms, and Bilinear Pairings. We review several schemes to illustrate different approaches and practical solutions.

Keywords: *ID-based cryptography, public key cryptography, elliptic curve, bilinear pairing.*

Введение

Криптографическая система на основе идентификаторов (IDentity-Based cryptosystem, IDB-system) — это асимметричная криптографическая система, в которой открытые ключи вычисляются по общедоступному алгоритму на основе идентификационной информации их владельцев (в дальнейшем для краткости будем называть такие системы IDB-*системами*).

Личные ключи вырабатываются *центром генерации ключей* KGC (Key Generation Center) на основе идентификационной информации и выдаются владельцам открытых ключей при личной встрече либо с использованием защищённого канала. Идентификационная информация для формирования открытого ключа может включать:

идентификаторы пользователей информационной системы; любую персональную информацию (адрес электронной почты, фотографию, номер телефона, почтовый адрес и т. п.); любые термины и условия, такие, как действующая политика, время, выполняемая роль, любые факты, связанные с конкретной стороной. Поэтому часто такие системы определяют как системы, в которых личные ключи формируются на основе открытых, а последние могут представлять собой произвольные текстовые строки.

Поскольку ключи однозначно определяются по идентификационной информации, то необходимость в сертификатах открытых ключей отпадает, а следовательно, отпадает необходимость в создании инфраструктуры открытых ключей, содержащей множество удостоверяющих центров. Поэтому данная технология формирования ключей представляется весьма перспективной и удобной для практических применений.

Первые работы по данному направлению появились около сорока лет назад, но активный поиск новых конструкций и подходов к их построению продолжается и в настоящее время. Общее число публикаций по данному направлению уже составляет несколько сотен. Хорошие обзоры по различным способам построения и разнообразным приложениям IDB-систем содержатся в работах [7, 10, 15, 16, 20, 27, 38, 57].

Многие IDB-системы доведены до включения в международные стандарты. Так, например, в стандарте ISO/IEC 14888-2:2007 [69] описана IDB-схема цифровой подписи GQ1 на основе RSA. В стандарте ISO/IEC 14888-3:2018 [70] приведены три IDB-схемы цифровой подписи:

— системы IBS-1 и IBS-2 на основе GDH-групп[1] [12];
— китайская система IBS на основе билинейного спаривания [73].

В стандарте ISO/IEC 11770-3:2015 [71] описаны два IDB-протокола выработки общего ключа:

— на основе системы Смарта — Чена — Ченга (N. Smart, L. Chen, Z. Cheng) [17];
— на основе системы Фуджиока — Сузуки — Устаоглу (Fujioka, Suzuki, Ustaoglu),

и один IDB-протокол защищённой передачи ключа:

— на основе системы Сакаи — Касахары [55].

Стандарт IEEE P1363a-2004 [72] определяет четыре типа криптосистем:

— IBS-системы шифрования;
— IBS-системы инкапсуляции ключа (Key Encapsulation);
— IBS-системы цифровой подписи;
— IBS-системы одновременного шифрования и подписи (Signcryption).

Кроме того, рабочая группа IETF S/MIME выпустила несколько проектов, касающихся криптографических методов на основе идентификаторов.

Есть также несколько опубликованных RFC, которые доказывают интерес и доверие научного сообщества к этой криптографической технике (таблица).

[1] GDH (Gap Diffie — Hellman groups) — таким термином обозначают класс циклических групп, для которых вычислительная проблема Диффи — Хеллмана (CDHP) является трудной, в то время как проблема распознавания Диффи — Хеллмана (DDHP) оказывается простой.
Более точно: пусть G — аддитивная группа и $a, b, c \in \mathbb{Z}_p$.

1. Computation Diffie — Hellman Problem (CDHP): для (P, aP, bP) вычислить abP.
2. Decisional Diffie — Hellman Problem (DDHP): для (P, aP, bP, cP) распознать, когда $c = ab$ в $\mathbb{Z}_p$.

Группа G является GDH-группой, если DDHP разрешима за полиномиальное время, но никакой вероятностный полиномиальный алгоритм не сможет решить CDHP с непренебрежимо малым преимуществом за полиномиальное время.

Документы IETF, содержащие описание IDB-систем

Номер	Год	Название
RFC 1824	1995	IBC Protocol for Authenticated Key-Exchange
RFC 5091	2007	IBC Standard #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems
RFC 5408	2009	IBE Architecture and Supporting Data Structures
RFC 5409	2009	Using the BF and BB1 Algorithms with the Cryptographic Message Syntax
RFC 6267	2011	MIKEY-IBAKE: Identity-Based Authenticated Key Exchange (IBAKE) Mode of Key Distribution in Multimedia Internet KEYing (MIKEY)
RFC 6508	2012	Sakai-Kasahara Key Encryption (SAKKE)
RFC 6509	2012	MIKEY-SAKKE: Sakai-Kasahara Key Encryption in Multimedia Internet KEYing (MIKEY)
RFC 6539	2012	IBAKE: Identity-Based Authenticated Key Exchange
RFC 7859	2016	Identity-Based Signatures for Mobile Ad Hoc Network (MANET) Routing Protocols

Далее работа построена следующим образом: в п. 1 обсуждаются особенности практического применения IDB-систем и возникающие при этом проблемы. В п. 2 приведены основные математические конструкции, применяемые для формирования личных ключей пользователей, соответствующих открытым ключам на основе их идентификаторов. В п. 3 рассмотрены конкретные примеры криптографических IDB-систем: системы шифрования (включая иерархические, анонимные, с использованием сертификатов и без них), цифровой подписи (включая signcryption- и higncryption-системы), аутентификации сторон, а также системы выработки общего ключа и замены ключа.

1. Особенности практического применения

Перечислим особенности практического применения IDB-систем, их недостатки и способы их устранения при построении приложений.

1.1. Масштабирование на несколько регионов

Нельзя утверждать, что из-за отсутствия необходимости в сертификатах потребность в инфраструктуре полностью отпадает. Для формирования личных ключей подписи пользователей необходима доверенная третья сторона — центр генерации ключей KGC. Он формирует личные ключи на основании идентификационной информации их владельцев, используя свой *главный* (*мастер-*) *ключ*, представляющий собой ключевую пару (открытый/закрытый мастер-ключ). При этом мастер-ключ должен быть один, и он должен принадлежать только центру генерации ключей, который должен быть также один.

Это накладывает ограничение на максимальное количество пользователей, так как они должны быть зарегистрированы и должны получить ключевые пары только в одном таком центре. Значит, круг использования этой технологии ограничен сотрудниками одного региона, одной организации, предприятия, либо клиентами одного банка или какого-либо другого поставщика услуг.

Для расширения возможностей данной технологии применяются так называемые иерархические IDB-системы HIDE (Hierarchical IDentity-based Encryption), в которых имеется множество локальных центров KGC, образующих древовидную структуру. Корнем дерева является главный центр, а центр каждого нижележащего уровня выдаёт ключи центрам следующего уровня. Пользователям соответствуют листья этого дерева. При этом открытый ключ каждого пользователя должен зависеть от идентификаторов центров, лежащих на пути от этой вершины до корня дерева. Такой способ формирования открытых ключей приводит к усложнению процедуры отзыва и обновления ключей.

1.2. Обеспечение анонимности в иерархических IDB-системах

В системах открытого шифрования условие анонимности получателя обычно обеспечивается путём шифрования идентификаторов получателя вместе с отправляемым сообщением. Поэтому убедиться в правильности адресата может только тот, кто владеет личным ключом получателя. С другой стороны, для формирования такого сообщения необходимо знать актуальную информацию об открытом ключе получателя, которая содержится в действующем сертификате открытого ключа. Этот сертификат должен быть своевременно получен отправителем, причём он пересылается, как правило, в открытом виде. Поэтому в коммуникационных сетях противник, анализируя трафик, имеет возможность получить информацию о получателе.

IDB-системы исключают такую возможность, поскольку там отсутствуют сертификаты, а открытый ключ вычисляется непосредственно из идентификационной информации получателя. Поэтому такие системы очень удобны для построения анонимных (иерархических) систем шифрования (Anonymous (Hierarchical) Identity-Based Encryption, A(H)IBE) для анонимных коммуникационных систем, в которых по шифртексту невозможно определить ни отправителя, ни получателя. Это понятие введено впервые в работе М. Абдала и др. [1]. Примеры A(H)IBE-схем получаются, например, на основе схемы Боне — Франклина.

Помимо коммуникационных систем, данные конструкции применяются при удалённом получении информации из баз данных. В той же работе [1] изучались *системы открытого шифрования с возможностью поиска ключевых слов* (Public-key Encryption with Keyword Search, PEKS). PEKS — это системы, в которых шифртекст ассоциирован с ключевым словом, причём выполняется требование о невозможности получения никакой информации об этом слове. Пользователь получает в центре KGC, помимо ключевой пары, ещё однонаправленную функцию для каждого используемого им ключевого слова. Для получения всех записей, содержащих ключевое слово, он может обратиться к администратору удалённой базы данных и передать ему соответствующую однонаправленную функцию. Администратор может выбрать все такие зашифрованные записи, причем он не сможет получить никакой информации ни о ключевом слове, ни о содержащейся в этих зашифрованных записях информации.

1.3. Депонирование ключа в IDB-системах

Критичным свойством IDB-систем является присущая им возможность создания системы депонирования личных ключей. Центр KGC, обладая мастер-ключом, имеет возможность вычислять ранее выданные личные ключи всех пользователей. Это позволяет без труда создать легитимную систему депонирования личных ключей, при которой при наличии решения суда правоохранительные органы могут запросить применяемые ключи у центра, а центр обязан предоставить личный ключ указанного пользователя. При этом центру нет необходимости хранить эти ключи, так как он может их заново вычислить по идентификационной информации.

1.4. Защита от нечестного центра KGC

В криптографических системах с инфраструктурой открытых ключей PKI на основе стандарта X.509 удостоверяющий центр, входящий в инфраструктуру PKI, отвечает только за подлинность соответствия между указанными в сертификате открытым ключом и идентификатором пользователя. Удостоверяющий центр выдаёт только сертификат ключа, а свой личный ключ пользователь может держать в секрете, не предъявляя его центру. Единственное, что пользователь должен сделать при получе-

нии сертификата открытого ключа, — это доказать наличие у него второй половины ключевой пары, что может быть произведено без раскрытия этого ключа.

В системах с открытыми ключами на основе идентификационной информации центр KGC сам формирует личные ключи всех пользователей. Поэтому большим недостатком IDB-систем является то, что нечестный центр KGC, обладая мастер-ключом, имеет возможность самостоятельно вычислять текущие и выданные ранее личные ключи всех пользователей по их идентификаторам, а поэтому и читать шифрованную переписку и подделывать цифровую подпись каждого пользователя.

Поэтому необходимо дополнить IDB-систему механизмами, позволяющими защитить пользователей от нечестного поведения центра KGC, например применяя протокол доказательства с нулевым разглашением. Такой подход используется в системах шифрования с извлечением ключа вслепую, в которых пользователь имеет возможность получить личный ключ, не раскрывая центру ни ключа, ни своего идентификатора, и в системах защиты цифровой подписи от подделки подписи со стороны центра [18], где с помощью такого доказательства пользователь, сохраняя в тайне свой личный ключ, может доказать на его основе арбитру, что это не его подпись.

1.5. IDB-системы на основе сертификатов (CBC-системы)

Другой способ защиты от нечестного центра KGC предоставляют *системы с открытыми ключами, построенные на основе сертификатов* (Certificate-Based Cryptography, CBC) и сохраняющие преимущества PKI и IDB-систем. Теперь каждый пользователь сам формирует свою ключевую пару и запрашивает сертификат в доверенном сертификационном центре CA (Certificate Authority). При этом центр CA формирует сертификат с помощью IDB-алгоритма, но при этом он не может восстановить личный ключ пользователя. Такие системы уже не относятся непосредственно к IDB-системам, но сочетают в себе преимущества обычных систем с открытыми ключами и IDB-систем.

1.6. Системы без сертификатов (CLC-системы)

Ещё одним направлением исследований является *открытая криптография без сертификатов* (CertificateLess Cryptography, CLC) [57], где также решается проблема депонирования ключа центром, унаследованная от систем на основе идентификаторов. В данном случае не требуются ни сертификаты, ни инфраструктура PKI. Вместо них доверенная сторона — центр KGC — формирует частичные личные ключи аналогично IDB-системам. Действующий личный ключ пользователя получается путём объединения полученного частичного ключа и выбранного им самим секрета. Поэтому он остаётся неизвестным и не хранится в центре KGC, что устраняет проблему депонирования ключа центром KGC.

Хотя такие системы имеют много общего с системами шифрования на основе сертификатов, каждый подход имеет свои достоинства и отличительные особенности.

1.7. Как заменить скомпрометированные ключи?

Ещё одной проблемой IDB-систем является отзыв и замена скомпрометированных ключей. Для любых систем с открытыми ключами, основанных на PKI или на ID, должна быть обеспечена процедура отзыва скомпрометированных ключей. В традиционных PKI это решается путём включения в сертификат предустановленного срока годности и ведения актуального списка аннулированных сертификатов. В IDB-системах с открытыми ключами на основе идентификаторов замена ключей представляет проблему, поскольку непонятно, как можно заменять имеющиеся идентификаторы. Самое простое практическое решение для облегчения процедуры отзы-

ва ключей предполагает дополнение идентификаторов некоторой заменяемой информацией, например, сроком действия, ключевым словом и т. п. Более того, можно предусмотреть регулярную замену ключей независимо от того, был ли ключ скомпрометирован или нет, например используя идентификационную информацию вида "`receiver-address`$\|$`current-date`" [20, 22], где `date` может быть днём, неделей, месяцем или годом.

Такой способ оказывается неудобным для систем с большим числом пользователей, поскольку они должны постоянно контактировать с одним доверенным центром KGC, нагрузка на который резко возрастает. Кроме того, важную роль приобретает проблема обеспечения аутентичности самой идентификационной информации, так как отправители могут путать похожие наборы данных и тем самым неправильно формировать открытые ключи получателя. В результате они будут ненамеренно отправлять зашифрованные сообщения не тем адресатам, причём последние смогут прочитать содержащуюся в них информацию.

Поэтому в нескольких работах предложены IDB-системы с процедурой отзыва ключей (IDB-cryptosistem with revocation) [6, 40, 42, 58, 39], где предлагаются специальные математические конструкции, ускоряющие и облегчающие этот процесс и позволяющие упростить работу центра KGC, заменив оценку трудоёмкости с линейной на логарифмическую от числа пользователей, и при этом сохранить простоту работы для самих пользователей.

1.8. Другие приложения

По данным Voltage, сегодня технология IBE защищает данные для более чем 100 миллионов пользователей по всему миру и совместима с такими широко распространенными продуктами, как Outlook, Yahoo, Gmail и др. В [33] содержится большой обзор применений IDB-систем для сенсорных сетей, где отмечается, что в приложениях с ограниченными доступными системными ресурсами факторы, касающиеся общей производительности системы, становятся гораздо более важными. Благодаря экономичности и низким требованиям к инфраструктуре, IDB-системы хорошо соответствуют требованиям к таким сетям. Их применение приводит к значительному сокращению накладных расходов на связь, а также к более рациональному использованию пропускной способности. Иерархические модели, которые могут быть получены из IBC, по-видимому, идеально вписываются в инфраструктуру сенсорной сети. Более того, ряд специальных, ключевых особенностей IBC (распределённый центр генерации ключей, системная иерархия, отзыв ключей, делегирование) решаются простым и элегантным способом.

Далее при описании протоколов будем использовать следующие обозначения:

— $a \in_R \mathbb{Z}_n$ — выбрать случайный элемент a из $\mathbb{Z}_n$;
— $F \stackrel{?}{=} G$ — проверить совпадение F и G;
— $\mathbb{G} = \langle P \rangle$ — циклическая группа $\mathbb{G}$ порождена элементом P.

В случаях, когда это не вызывает разночтений, будем для упрощения записи опускать знак конкатенации в аргументе хеш-функции $h(x, y, \ldots, z) = h(x\|y\| \ldots \|z)$.

2. Математические конструкции для формирования личных ключей на основе идентификаторов

Формирование личных ключей пользователей (key extraction), соответствующих открытым ключам на основе их идентификаторов, производится в центрах KGC. Перечислим применяемые при этом основные алгоритмы формирования ключевых пар.

2.1. IDB-системы на основе RSA

Первый способ построения IDB-системы предложил А. Шамир в 1984 г. [60]. За основу была взята система RSA, основанная на сложности задачи факторизации целых чисел. Пусть $h : \{0,1\}^* \to \mathbb{Z}_n$ — хеш-функция, $n = pq$.

Открытый ключ центра KGC — это пара (e, n), где e — большое простое число, не являющееся делителем числа $(p-1)(q-1)$.

Закрытый ключ центра — (p, q, d), где $d = e^{-1} \bmod (p-1)(q-1)$.

Пользователь A предъявляет центру свою идентификационную информацию $\mathrm{ID_A}$ и центр выдаёт ему ключевую пару $(pk_A, sk_A) = (h(\mathrm{ID_A}), pk_A^d)$, т. е. $h(\mathrm{ID_A}) = sk_A^e \bmod n$.

Значения d и $sk_A = pk_A^d \bmod n$ центр может вычислить, зная разложение числа n.

2.2. IDB-системы на основе квадратичных вычетов

В 2001 г. в работе [19] К. Кокс (C. Cocks) предложил схему на основе сложности задачи нахождения квадратного корня в кольце вычетов по составному модулю: $x \in \mathbb{Z}_n$, $a = x^2 \bmod n$, $n = pq$, p, q — различные большие простые числа, удовлетворяющие условию $p \equiv q \equiv 3 \pmod 4$.

Пусть $J(n)$ — множество всех элементов кольца $\mathbb{Z}_n$, которые имеют символ Якоби равный 1; $QR(n) \subset J(n)$ — множество всех квадратичных вычетов по модулю n; $u \in J(n) \setminus QR(n)$; $h\colon \{0,1\}^* \to J(n)$ — хеш-функция.

Тройка (n, u, h) составляет набор открытых параметров.

Заметим, что из условия $a \in J(n)$ следует, что $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = 1$ или $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = -1$. Поэтому либо a, либо $(-a)$ является квадратичным вычетом по модулю n.

При этом извлечь квадратный корень может только центр KGC, знающий числа p и q, являющиеся секретом KGC. Для этого он может воспользоваться формулой

$$r = a^{(n+5-(p+q))/8} \bmod n.$$

Такое r удовлетворяет условию $r^2 = a \pmod n$ или $r^2 = -a \pmod n$ в зависимости от того, что является квадратичным вычетом: a или $-a$.

Пользователь A предъявляет центру свою идентификационную информацию $\mathrm{ID_A}$ и центр выдаёт ему ключевую пару $(pk_A, sk_A) = (a, r)$, где:

— открытый ключ пользователя равен $a = h(\mathrm{ID_A}) \in J(n)$;

— личный ключ пользователя определяется как

$$r = \begin{cases} \sqrt{a} \bmod n, & a \in QR(n); \\ \sqrt{ua} \bmod n, & a \in J(n) \setminus QR(n). \end{cases} \tag{1}$$

2.3. IDB-системы на основе задачи логарифмирования в мультипликативной группе поля

К. Гюнтер (C. G. Günther) в [32] предложил способ построения IDB-системы на основе цифровой подписи центра KGC, вычисленной по схеме Эль Гамаля. Закрытым и открытым ключами центра KGC являются соответственно элементы $x_S \in \{1, \ldots, p-1\}$ и $y_S = g^{x_S} \in \mathbb{Z}_p^*$, где p — большое простое число.

Пользователь A получает в KGC цифровую подпись (u_A, v_A) для своего идентификатора ID_A, где

$$u_A = g^{k_A}; \quad v_A = (\mathrm{ID}_A - x_S u_A) k_A^{-1} \bmod (p-1); \quad k_A \in_R \mathbb{Z}_p^*; \quad (k_A, p-1) = 1.$$

Проверка подписи проводится с помощью уравнения

$$u_A^{v_A} = g^{\mathrm{ID}_A} y_S^{-u_A}.$$

Теперь ключевая пара пользователя A определяется как

$$(pk_A, sk_A) = ((\mathrm{ID}_A, u_A), v_A),$$

где в роли открытого ключа пользователя A выступает его идентификатор и первая половина подписи, а в роли личного ключа — вторая половина подписи $v_A = \log_{u_A}(g^{\mathrm{ID}_A} y_S^{-u_A})$. Поэтому проблема определения личного ключа по открытому ключу сводится к проблеме дискретного логарифмирования в $\mathbb{Z}_p^*$.

2.4. IDB-системы на основе группы точек эллиптической кривой

Фактически, все первоначально разработанные IDB-схемы на основе группы точек эллиптической кривой предполагают наличие операции *билинейного спаривания*. В общем случае эта операция определяется на двух абелевых группах $\mathbb{G}_1$ и $\mathbb{G}_2$ простого порядка q и принимает значение в третьей мультипликативной группе $\mathbb{G}_T$ того же порядка:

$$e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T.$$

При этом должны выполняться два свойства:

1) *билинейность*: при всех $w, x \in \mathbb{G}_1$ и $y, z \in \mathbb{G}_2$ выполнены тождества

$$e(w, x+z) = e(w,x)\, e(w,z), \quad e(w+x, z) = e(w,z)\, e(x,z);$$

2) *невырожденность*: для некоторых элементов $x \in \mathbb{G}_1$ и $y \in \mathbb{G}_2$ выполнено $e(x,y) \neq 1$.

В практических применениях используются в основном операции спаривания для групп точек эллиптической кривой. Поэтому будем рассматривать только такие операции. В этом случае группы $\mathbb{G}_1$ и $\mathbb{G}_2$ являются одинаковыми или различными циклическими подгруппами группы точек эллиптической кривой над конечным полем или его расширением, а $\mathbb{G}_T$ — это подгруппа мультипликативной группы поля. При рассмотрении групп точек эллиптических кривых над конечным полем будем обозначать точки большими латинскими буквами, а элементы поля — маленькими.

Если $P \in \mathbb{G}_1$ и $n \in \mathbb{N}$, то для кратных точек будем использовать обозначение

$$[n]P = \underbrace{P + \ldots + P}_{n}.$$

В [10] выделено два типа IDB-схем, различающихся по способу формирования ключевых пар участников:

— тип SOK (от Сакаи — Огиши — Казахара (R. Sakai, K. Ohgishi, M. Kasahara) [53]) предполагает наличие двух хеш-функций h_1 и h_2, принимающих значение соответственно в группах $\mathbb{G}_1$ и $\mathbb{G}_2$;

— тип SK (от Сакаи — Казахара (R. Sakai, M. Kasahara) [55]) использует только одну хеш-функцию h, принимающую числовое значение, которое получается представлением двоичного вектора — значения хеш-функции соответствующим натуральным числом.

Для обоих типов ключевая пара центра KGC имеет вид $([s]P, s)$, где s — закрытый ключ $(1 < s < q-1)$; $[s]P$ — открытый ключ; P — образующий элемент группы $\mathbb{G}_1$.

Для IDB-схем типа SOK ключевая пара пользователя A определяется как

$$(pk_A, sk_A) = (h_1(\mathrm{ID}_A), [s]h_1(\mathrm{ID}_A)).$$

Для IDB-схем типа SK ключевая пара пользователя A определяется как

$$(pk_A, sk_A) = ([s + h(\mathrm{ID}_A)]P, [s + h_1(\mathrm{ID}_A)]^{-1}P),$$

где открытый ключ участника A зависит от закрытого ключа s центра KGC, причём выполняется соотношение $e(pk_A, sk_A) = e(P, P)$.

2.5. IDB-системы на основе нечётких множеств

В 2005 г. А. Сахаи и Б. Уотерс (A. Sahai и B. Waters) [56] предложили новый способ построения IDB-системы на основе нечётких множеств, в которой идентификатор рассматривается как набор описательных атрибутов.

Нечёткая схема IBE позволяет использовать закрытый ключ, соответствующий открытому ключу с идентификатором ω, для расшифрования текста, зашифрованного на открытом ключе с идентификатором ω', тогда и только тогда, когда идентификаторы ω и ω' находятся близко друг к другу, что определяется метрикой, позволяющей оценить «перекрытие» множеств: $|\omega \cap \omega'| \geqslant d$ при некотором заданном параметре d.

Нечёткая схема IBE может быть применена для обеспечения шифрования с использованием в качестве идентификационной информации биометрических входных данных. Поскольку биометрические данные обладают некоторым шумом, их использование в обычных IDB-системах невозможно. Однако свойство устойчивости к ошибкам нечёткой схемы IBE позволяет расшифровать зашифрованные данные с помощью личного ключа, восстановленного из биометрических данных, который может отличаться от истинного личного ключа.

Кроме того, Fuzzy-IBE можно использовать для приложений, которые называются «шифрованием на основе атрибутов». В таких приложениях стороны отправляют зашифрованные сообщения всем пользователям, имеющим заданное множество атрибутов, например «члены комиссии», «сотрудники отдела» и т. п., которые составляют (нечёткую) идентификацию. Поэтому такие сообщения могут прочитать только те, у кого в идентификационной информации есть соответствующие атрибуты.

Преимуществом такого подхода является то, что документы могут храниться на обычном сервере без доверенных средств (удалённой) аутентификации.

Благодаря этому свойству IDB-системы на основе нечётких множеств применяются в системах, допускающих эффективную процедуру отзыва ключей. Далее в п. 3.1 описана нечёткая система шифрования Fuzzy-IBE (Fuzzy Identity-Based Encryption) из [56], а в п. 3.2 — её применение в системах с процедурой отзыва ключей из [38].

2.6. Иерархические IDB-системы

Для масштабирования IDB-систем с целью применения их в больших распределённых системах в [23] предложена иерархическая структура IDB-системы, в которой может быть несколько локальных центров KGC, образующих древовидную структуру, корнем которой является корневой центр генерации ключей rootPKG, а каждый центр выдаёт ключи только связанным с ним центрам следующего уровня (рис. 1). Листьям дерева соответствуют пользователи.

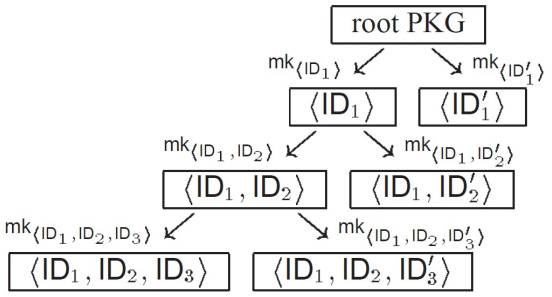

Рис. 1. Иерархическая структура локальных центров [23]

Каждый локальный центр PKG, соответствующий вершине, расположенной на i-м уровне, однозначно характеризуется набором идентификаторов $(\mathrm{ID}_1, \ldots, \mathrm{ID}_i)$, соответствующих пути от корня к этой вершине. Поэтому в качестве открытого ключа для этого центра можно выбрать значение $h(\mathrm{ID}_1 \| \ldots \| \mathrm{ID}_i)$, где h — некоторая хеш-функция, а закрытый ключ $\mathrm{mk}_{(\mathrm{ID}_1,\ldots,\mathrm{ID}_i)}$ формируется на основе открытого ключа.

В п. 3.1 приведён пример иерархической IDB-системы шифрования HIDE [23].

Другие способы формирования ключевых пар пользователей рассмотрены далее при описании конкретных классов IDB-систем.

3. Примеры криптографических IDB-систем

3.1. IDB-системы шифрования (IBE)

IBE-схема на основе квадратичных вычетов

В [19] К. Кокс (C. Cocks) предложил IDB-схему шифрования, использующую описанный в п. 2.2 способ формирования ключевых пар на основе проблемы извлечения квадратного корня в кольце вычетов $\mathbb{Z}_n$, $n = pq$, где p и q — простые числа, удовлетворяющие условию $p \equiv q \equiv 3 \pmod 4$. Пользователь A предъявляет центру свою идентификационную информацию ID_A и получает ключевую пару $(pk_A, sk_A) = (a, r)$, где $a = h(\mathrm{ID}_A) \in J(n)$, а r находится по формуле (1).

Зашифрование происходит побитно: для зашифрования бита m (который закодирован $+1$ или -1) надо:

1) выбрать случайные элементы $t_0, t_1 \in \mathbb{Z}_n$;
2) вычислить

$$d_i = \frac{t_i^2 + u^i a}{t_i}, \quad c_i = m\left(\frac{t_i}{n}\right), \quad i \in \{0, 1\}.$$

Шифртекст состоит из двух элементов $((d_o, c_0), (d_1, c_1))$. Поэтому при зашифровании одного бита получится $O(\log n)$ битов шифртекста.

Расшифрование:

1) определить $i \in \{0, 1\}$, такое, что $r^2 = u^i a$;
2) вычислить $g = d_i + 2r$, которое можно записать так:

$$g = d_i + 2r = \frac{t_i^2 + r^2}{t_i} + 2r = \frac{(t_i + r)^2}{t_i} = \left[\frac{t_i + r}{t_i}\right]^2 t_i.$$

Отсюда следует, что $\left(\frac{g}{n}\right) = \left(\frac{t_i}{n}\right)$;

3) вычислить $m = c_i \left(\frac{t_i}{n}\right)$.

IBE-система Боне — Франклина на основе билинейного спаривания (BF-IDE)

Первая практически эффективная иерархическая система предложена в работе D. Boneh и M. Franklin в 2001 г. [9]. Она основана на операции спаривания для эллиптических кривых $e\colon \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$, где $\mathbb{G}_1$ — подгруппа группы точек эллиптической кривой; $\mathbb{G}_T$ — подгруппа мультипликативной группы поля. Пусть также имеются хеш-функции $h_1\colon \{0,1\}^* \to \mathbb{G}_1$ и $h_2\colon \mathbb{G}_T \to \{0,1\}^n$.

Центр KGC обладает ключевой парой (s, H), $H = [s]P$, $P \in \mathbb{G}_1$.

Участник A получает в центре KGC открытый ключ $Q_A = h_1(\mathrm{ID}_A)$ и личный ключ $S_A = [s]Q_A$. Для зашифрования сообщения $M \in \{0,1\}^*$, отправляемого участнику A, надо выбрать случайное значение $r \in \mathbb{Z}_q^*$ и сформировать шифртекст

$$C = (\,[r]P, M + h_2(e(Q_A, H)^r)\,) = (U, V).$$

Получив это сообщение, A вычисляет открытый текст по формуле

$$M = C + h_2(e(S_A, U)),$$

так как $e(S_A, U) = e([s]Q_A, [r]P) = e(Q_A, [s]P)^r = e(Q_A, H)^r$.

Данная схема в качестве конструктивного блока нашла многочисленные практические применения в протоколах для иерархических, облачных, широковещательных и др. IDB-систем [23, 22, 34, 2, 67, 14].

Позднее Д. Галиндо (D. Galindo) [21] обнаружил уязвимость этого протокола и предложил исправленный и улучшенный вариант.

Система Fuzzy-IBE на основе нечётких множеств

В нечёткой системе шифрования Fuzzy-IBE (Fuzzy Identity-Based Encryption), предложенной в 2005 г. А. Сахаи и Б. Уотерс (A. Sahai, B. Waters) [56], идентификатор рассматривается как набор описательных атрибутов.

Пусть $e\colon \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ — операция билинейного спаривания для групп $\mathbb{G}_1$ и $\mathbb{G}_T$ простого порядка p и P_0 — образующий элемент группы $\mathbb{G}_1$. Определим коэффициент Лагранжа для $i \in \mathbb{Z}_p$ и $S \subset \mathbb{Z}_p$ равенством

$$\delta_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j} = \begin{cases} 1, & x = i \in S; \\ 0, & x \in \mathbb{Z}_p \setminus \{i\}. \end{cases} \tag{2}$$

Пусть идентификаторами являются подмножества ω некоторого множества $\mathcal{U}$. Множество $\mathcal{U}$ может быть множеством всевозможных атрибутов или множеством оцифрованных значений некоторой биометрической системы распознавания. Каждому элементу подмножества ω однозначно сопоставлен некоторый элемент из $\mathbb{Z}_p^*$. Пусть для простоты элементам из $\mathcal{U}$ соответствуют первые $|\mathcal{U}|$ элементов $\{1, \ldots, |\mathcal{U}|\}$ из $\mathbb{Z}_p^*$.

Выберем случайно и равновероятно $t_1, \ldots, t_{|\mathcal{U}|}, y \in_R \mathbb{Z}_p^*$. Открытыми параметрами центра являются

$$(T_1 = [t_1]P_0,\ \ldots,\ T_{|\mathcal{U}|} = [t_{|\mathcal{U}|}]P_0,\ Y = e(P_0, P_0)^y),$$

а мастер-ключ определяется как $(t_1, \ldots, t_{|\mathcal{U}|}, y)$.

Пользователь с идентификатором $\omega \subset \mathcal{U}$ получает личный ключ на основе случайного многочлена $q(x) \in \mathbb{Z}_p[x]$ степени $d-1$ с $q(0) = y$ как упорядоченный набор $(D_i)_{i\in\omega}$, где $D_i = \left[\dfrac{q(i)}{t_i}\right] P_0$, $i \in \omega$.

Зашифрование. Для зашифрования сообщения $m \in \mathbb{G}_T$ на открытом ключе ω' выбирают случайный элемент $r \in \mathbb{Z}_p$ и формируют шифртекст вида

$$C = (\omega',\ c' = mY^r,\ \{C_i = [r]T_i\}_{i\in\omega}).$$

Расшифрование. Если шифртекст получен на ключе ω', то для его расшифрования на ключе ω, удовлетворяющем условию $|\omega \cap \omega'| \geqslant d$, следует выбрать произвольное подмножество $\sigma \subseteq \omega \cap \omega'$ мощности d и вычислить

$$c' / \prod_{i\in\sigma} e(G_i, C_i)^{\delta_{i,\sigma}(0)} = m \cdot e(P_0, P_0)^{ry} / \prod_{i\in\sigma} e\left(\left[\frac{q(i)}{t_i}\right] P_0, [rt_i]P_0\right)^{\delta_{i,\sigma}(0)} =$$

$$= m \cdot e(P_0, P_0)^{ry} / \prod_{i\in\sigma} \left(e(P_0, P_0)^{rq(i)}\right)^{\delta_{i,\sigma}(0)} = m \cdot e(P_0, P_0)^{ry} / e(P_0, P_0)^{r\sum\limits_{i\in\sigma} q(i)\delta_{i,\sigma}(0)} = m.$$

Данное выражение вытекает из формулы интерполяции стоящего в показателе многочлена $q(x)$ степени $d-1$ по d точкам:

$$\sum_{i\in\sigma} q(i)\delta_{i,\sigma}(0) = q(0) = y.$$

Иерархические IBE-системы

В [23] предложена иерархическая структура IDB-системы — HIDE (Hierarchical IDentity-based Encryption), в которой локальные центры KGC образуют древовидную структуру, корнем которой является корневой центр генерации ключей rootPKG, а каждый центр выдаёт ключи только связанным с ним центрам следующего уровня (см. рис. 1). Конечные вершины, соответствующие листьям этого дерева, представляют пользователей.

Данная система была основана на конструкции Боне и Франклина [9].

Параметры корневого центра rootPKG:

1) Пусть p — k-битовое простое число, $e\colon \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ — операция спаривания, где $\mathbb{G}_1, \mathbb{G}$ — циклические группы порядка p; P_0 — случайно выбранный образующий элемент группы $\mathbb{G}_1$.
2) При некотором $n > 0$ выбираются криптографические хеш-функции

$$h_1\colon \{0,1\}^* \to \mathbb{G}_1,$$
$$h_2\colon \mathbb{G}_T \to \{0,1\}^n,$$
$$h_3\colon \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}_p,$$
$$h_4\colon \{0,1\}^n \to \{0,1\}^n.$$

3) Выбирается случайный элемент $s_0 \in \mathbb{Z}_p^*$ — корневой мастер-ключ, и формируется ключевая пара центра rootPKG $(s_0, Q_0 = [s_0]P_0)$.

Формирование ключей для центра, расположенного на i-м уровне.

При $i \geqslant 1$ каждый центр KGC_i, расположенный на i-м уровне, выбирает в качестве секрета элемент $s_i \in \mathbb{Z}_p^*$, $s_i \neq s$, и вычисляет $Q_i = [s_i]P_0$.

Центр KGC_i однозначно характеризуется набором идентификаторов $(\mathrm{ID}_1, \ldots, \mathrm{ID}_i)$, соответствующих пути от корня к этому центру. В качестве открытого ключа он выбирает значение $P_i = h_1(\mathrm{ID}_1 \| \ldots \| \mathrm{ID}_i)$, а для получения закрытого ключа обращается в центр KGC_{i-1} с идентификатором $(\mathrm{ID}_1 \| \ldots \| \mathrm{ID}_{i-1})$.

Пусть центр KGC_{i-1} обладает секретом s_{i-1} и мастер-ключом

$$mk_{ID_{i-1}} = (S_{i-1}, Q_1, \ldots, Q_{i-1})$$

(при $i = 1$ центр KGC_1 обладает секретом s_1 и мастер-ключом $mk_1 = (S_1, Q_1)$, где S_1 — единичный элемент группы $\mathbb{G}_1$).

Для вычисления личного ключа центра KGC_i центр KGC_{i-1}:

1) вычисляет $P_i = h_1(\mathrm{ID}_1 \| \ldots \| \mathrm{ID}_i)$;
2) вычисляет $S_i = S_{i-1} + [s_{i-1}]P_i = \sum\limits_{j=1}^{i} [s_{j-1}]P_i$;
3) возвращает $(S_i, Q_1, \ldots, Q_{i-1})$, где $Q_j = [s_j]P_0$, $1 \leqslant j \leqslant i-1$.

Теперь подчинённый ему центр KGC_i получает закрытый ключ

$$mk_{ID_i} = (S_i, Q_1, \ldots, Q_i).$$

Для *зашифрования сообщения* $M \in \{0,1\}^*$ для центра KGC_i надо:

1) вычислить $P_i = h_1(\mathrm{ID}_1, \ldots, \mathrm{ID}_j)$, $1 \leqslant j \leqslant i$;
2) вычислить $g = e(Q_0, P_1)$;
3) выбрать случайное $\sigma \in \{0,1\}^n$ и вычислить $r = h_3(\sigma, M)$;
4) определить

$$C = ([r]P_0,\ [r]P_2,\ \ldots,\ [r]P_i,\ \sigma \oplus h_2(rg),\ M \oplus h_4(\sigma)).$$

Расшифрование.

Получив сообщение $C = (U_0, U_2, \ldots, U_i, V, W)$, участник $(\mathrm{ID}_1, \ldots, \mathrm{ID}_i)$ должен:

1) вычислить

$$g' = \frac{e(U_0, S_t)}{\prod\limits_{j=2}^{i} e(Q_{j-1}, U_j)};$$

2) вычислить $\sigma = V \oplus h_2(g')$;
3) вычислить $M = W \oplus h_4(\sigma)$;
4) вычислить $r = h_3(\sigma, M)$. Если $U_0 \neq [r]P_0$, то прервать протокол. Если нет, то принять открытый текст M.

Заметим, что вместо гаммирования $W = M \oplus h_4(\sigma)$ можно использовать любой алгоритм блочного шифрования $W = E_{h_4(\sigma)}(M)$.

IBE-протокол Уотерса

Система шифрования, предложенная B. Waters в [66], использует симметричное спаривание эллиптических кривых $e\colon \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$, $\mathbb{G}_1 = \langle P_0 \rangle$. Закрытый ключ центра — случайный элемент $x \in_R \mathbb{Z}_p$, открытым ключом является $Q = [x]P_0$. Пусть пользователь A характеризуется идентификатором

$$\mathrm{ID}_A = (a_1, \ldots, a_n) \in \{0,1\}^n.$$

Обозначим $\omega = \{i\colon a_i = 1\} \subseteq \{1, \ldots, n\}$.

Для формирования ключа пользователя центр KGC выбирает случайные P_2, U', $U_1, \ldots, U_n \in_R \mathbb{G}_1$, и пусть $\mathbf{U} = (U_1, \ldots, U_n)$. Центр объявляет $(P_0, Q_0, P_2, \mathbf{U})$ открытыми параметрами пользователя и выдаёт ему личный ключ $d = e([x]P_2 + [r]V, [r]P_0)$, где $r \in_R \mathbb{Z}_p^*$ — случайное; $V = U' + \sum\limits_{i\in\omega} U_i$.

Зашифрование.

Для зашифрования сообщения $m \in \mathbb{Z}_p^*$ необходимо вычислить значение $V = U' + \sum\limits_{i\in\omega} U_i$, выбрать случайное $t \in_R \mathbb{Z}_p$ и сформировать шифртекст вида

$$C = (e(Q_0, P_2)^t \cdot m,\ [t]P_0, [t]V).$$

Расшифрование.

Для расшифрования шифртекста $C = (c_1, C_2, C_3)$ надо с использованием личного ключа $d = (d_1, d_2)$ вычислить $m = \dfrac{e(d_2, C_3)}{e(d_1, C_2)} \cdot c_1$.

Действительно,

$$\frac{e(d_2, C_3)}{e(d_1, C_2)} \cdot c_1 = \frac{e([r]P_0, [t]V)}{e([x]P_2 + [r]V, [t]P_0)} \cdot c_1 = \frac{e([r]P_0, [t]V)}{e([x]P_2, [t]P_0) \cdot e([r]V, [t]P_0)} \cdot c_1 =$$
$$= \frac{e(P_0, V)^{rt}}{e([x]P_0, P_2)^t \cdot e(P_0, V)^{rt}} \cdot e(Q_0, P_2)^t \cdot m = m.$$

Данная схема может быть модифицирована в иерархическую IDB-систему, в которой идентификатор центра i-го уровня имеет вид $\mathrm{ID} = (\mathrm{ID}_1, \ldots, \mathrm{ID}_i)$. В этом случае для каждого уровня надо генерировать свои параметры U' и $\mathbf{U}$.

Системы IBE вслепую (Blind IBE)

Для схем шифрования на основе идентификаторов существует протокол извлечения ключа (key extraction protocol), в котором пользователь отправляет строку с идентификационными данными центру KGC, который затем возвращает соответствующий секретный ключ для этого идентификатора. Этот протокол может быть выполнен для нескольких известных схем IBE эффективно вслепую, то есть пользователь может получить секретный ключ, соответствующий его идентификатору, без того чтобы главный центр узнал что-либо об этом идентификаторе. Схемы, поддерживающие протокол извлечения вслепую, называются *системами IBE вслепую* (blind IBE).

В [28] М. Грин и С. Хохенбергер (M. Green, S. Hohenberger) предложили эффективные протоколы извлечения ключа вслепую, удовлетворяющие этому определению, для схем IBE Боне — Бойена [8] и Уотерса [66] (используя обобщение, предложенное независимо Неккаче (D. Naccache) [46] и Чатержи — Сархара (Chatterjee, Sarkar) [13]). Последний протокол похож на схему подписи вслепую Окамото [49].

Рассмотрим эффективный протокол извлечения ключа вслепую, названный BlindExtract, для следующих IBE-систем:

(1) Боне — Бойена [8];
(2) обобщённой версии протокола IBE Уотерса [66], предложенной независимо Нэккаче [46] и Чатержи — Сархаром [13].

Так как обе схемы основаны на одинаковых конструкциях, сначала опишем их общие элементы. Пусть $\mathbb{G}_1 = \langle P \rangle$, $|\mathbb{G}_1| = q$, $f : \{0,1\}^* \to \mathbb{G}_1$ и $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ — операция билинейного спаривания. Выбираем $\alpha \in_R \mathbb{Z}_q^*$ и $H, P_2 \in_R \mathbb{G}_1$ и полагаем $P_1 = [\alpha]P$, $msk = [\alpha]P_2$ — закрытый ключ центра KGC.

Пользователь A отправляет центру свой идентификатор ID_A и получает от центра личный ключ вида

$$(D_0, D_1) = ([\alpha]P_2 + [r]f(\mathrm{ID}_A),\ [r]P) \in \mathbb{G}_1^2,$$

где $r \in_R \mathbb{Z}_q$. Корректность этой пары проверяется тестом

$$e(P_1, P_2)\, e(D_1, f(\mathrm{ID}_A)) = e(D_0, P).$$

Для зашифрования текста $m \in \mathbb{G}_T$ вырабатывается случайное число $s \in_R \mathbb{Z}_q$ и формируется шифртекст

$$C = (e(P_1, P_2)^s \cdot m,\ [s]P,\ [s]f(\mathrm{ID}_A)).$$

При расшифровании шифртекста $C = (c, Y, Z) \in \mathbb{G}_T \times \mathbb{G}_1^2$ используется ключ $(D_0, D_1) \in \mathbb{G}_1^2$ и формируется открытый текст

$$m = c \cdot e(Z, D_1)/e(Y, D_0).$$

Протокол извлечения ключа вслепую BlindExtract для обеих схем IBE (1) и (2) имеет следующий вид (A — пользователь, T — центр):

$$\begin{array}{ll}
A: & y \in_R \mathbb{Z}_q, \\
A \to T: & H' = [y]P + [\mathrm{ID}_A]P_1, \\
A \leftrightarrow T: & \mathrm{ZKproof}(y, \mathrm{ID}_A), \\
T: & r \in_R \mathbb{Z}_q, \\
T: & D_0' = [\alpha]P_2 + [r](H' + H), \\
T: & D_1' = [r]P, \\
A \leftarrow T: & (D_0', D_1'), \\
A: & \text{проверяет } e(P_1, P_2) \cdot e(D_1', H' + H) = e(D_0', P), \\
A: & z \in_R \mathbb{Z}_q, \\
A: & D_0 = D_0' - [y]D_1' + [z]f(\mathrm{ID}_A),\ D_1 = D_1' + [z]P.
\end{array}$$

Через $\mathrm{ZKproof}(y, a)$ здесь обозначен протокол доказательства с нулевым разглашением знания такой пары (y, a), что выполняется равенство $H' = [y]P + [a]P_1$. Этот протокол может быть сделан неинтерактивным, выполняемым однократной передачей сообщения $([r_1]P + [r_2]P_1, x, s_1, s_2)$, где $r_1, r_2 \in_R \mathbb{Z}_q$, $x = h([r_1]P + [r_2]P_1)$, $s_1 = r_1 + xr$, $s_2 = r_2 + xa$. Проверка правильности доказательства осуществляется путём рассмотрения равенства

$$[r_1]P + [r_2]P_1 = [s_1]P + [s_2]P_1 + [x]H'.$$

Различие между системами (1) и (2) заключается в выборе способа записи идентификаторов и конструкции функции $f : \{0,1\}^* \to \mathbb{G}_1$.

Для системы IBE (1) функция f определяется как

$$f(\mathrm{ID}_A) = H + [\mathrm{ID}_A]P_1.$$

Для системы IBE (2) идентификаторы $\mathrm{ID} = (a_1, \ldots, a_n)$, $a_j \in \{0,1\}^l$, выбираются из пространства битовых строк длины $N = nl$, а функция $f : \{0,1\}^N \to \mathbb{G}_T$ определяется как

$$f(\mathrm{ID}_A) = H + \sum_{j=1}^{n} [a_j] U_j,$$

где элементы $U_j \in \mathbb{G}_T$ выбираются центром случайно (Д. Неккаче предлагал использовать значения $N = 160$ и $l = 32$ [46]). Для протокола доказательства с нулевым разглашением ZKproof значение H' вычисляется по формуле $H' = [y]P + \sum_{j=1}^{n} [a_j] U_j$, $0 \leqslant a_i < 2^l$. Личный ключ пользователя, соответствующий идентификатору ID, где $r \in_R \mathbb{Z}_q$, имеет следующий вид:

$$(D_0, D_1) = ([\alpha]P_2 + [r] f(\mathrm{ID}_A), [r]P) = \left([\alpha]P_2 + [r] \left(H + \sum_{j=1}^{n} [a_j] U_j \right), [r]P \right).$$

CBC-системы шифрования

Системы шифрования на основе сертификатов (Certificate-Based Cryptosystem, CBC) помогают избавиться от недостатка обычных IDB-систем шифрования, где центр KGC имеет возможность вычислять личные ключи всех пользователей, а поэтому и знакомиться с содержанием шифрованной переписки каждого пользователя. В то же время они не требует наличия дорогостоящей инфраструктуры PKI, позволяющей постоянно проверять актуальность сертификата.

Сертификат используется как составная часть ключа расшифрования, который составлен непосредственно из сгенерированного пользователем личного ключа и полученного в центре сертификации CA сертификата. Хотя CA знает сертификат, он, в отличие от KGC, не может расшифровать ни одного шифртекста.

Пусть имеется операция симметричного билинейного спаривания для группы $\mathbb{G}_1 = \langle P \rangle$ и две хеш-функции $h_1 : \{0,1\}^* \to \mathbb{G}_1$ и $h_2 : \mathbb{G}_T \to \{0,1\}^n$, где n — длина открытого сообщения, подлежащего зашифрованию.

Центр CA выбирает закрытый ключ $s_C \in_R \mathbb{Z}_p^*$ и открытый ключ $Q_C = [s_C]P$.

Пользователь A обладает ключевой парой $(sk_A, pk_A) = (s_A, [s_A]P)$. Для получения сертификата он обращается в центр CA, отправляя туда сообщение infoA, содержащее значение открытого ключа $Q_A = [s_A]P$ и другую идентификационную информацию. Центр CA вычисляет $\mathcal{Q}_A = h_1(Q_A \,\|\, \mathrm{infoA})$ и возвращает сертификат $\mathrm{cert}_A = [s_C]\mathcal{Q}_A$. В качестве дополнительного аргумента для h_1 может быть включён период действия сертификата.

Теперь A подписывает infoA, формируя значение $[s_A]P_A$, где $P_A = h_1(\mathrm{infoA})$, и вычисляет личный ключ $S_A = \mathrm{cert}_A + [s_A]P_A$. Заметим, что это значение образовано из подписей центра CA и пользователя A под сообщениями $\mathcal{Q}_A$ и P_A соответственно.

Зашифрование сообщения m с использованием infoA выполняется следующим образом: вычисляется $\mathcal{Q}_A = h_1(Q_A \,\|\, \mathrm{infoA})$ и $P_A = h_1(\mathrm{infoA})$, выбирается случайное $t \in_R \mathbb{Z}_p^*$ и формируется шифртекст

$$(U, V) = ([t]P,\ M \oplus h_2(e(Q_C, \mathcal{Q}_A)^t e([s_A]P, P_A)^t)).$$

Расшифрование сообщения (U, V), зашифрованного с помощью infoA и открытого ключа $\mathcal{Q}_A$, осуществляется с помощью личного ключа S_A по формуле

$$M = V \oplus h_2(e(U, S_A)).$$

CLC-системы шифрования

Рассмотрим пример *системы шифрования без сертификатов* (Certificateless Cryptosystem), предложенный в работе [2]. Пусть, как и выше, имеется операция симметричного билинейного спаривания для группы $\mathbb{G}_1 = \langle P \rangle$. Центр выбирает закрытый ключ $s \in_R \mathbb{Z}_p^*$, и пусть открытый ключ $Q = [s]P$. Пусть $h_1 : \{0,1\}^* \to \mathbb{G}_1$ и $h_2 : \mathbb{G}_1^T \to \{0,1\}^n$ — хеш-функции; n — длина открытого сообщения, подлежащего зашифрованию.

Извлечение частичного личного ключа пользователя A осуществляется согласно алгоритму BF-IBE Боне — Франклина: для идентификатора ID_A алгоритм возвращает значение $[s]H_1(\mathrm{ID}_A)$.

Пользователь вырабатывает секретное значение $x \in_R \mathbb{Z}_p^*$ и формирует действующий личный ключ $sk_A = [xs]H_1(\mathrm{ID}_A)$ и открытый ключ $pk_A = (X, Y) = ([x]P, [x]Q)$. Корректность открытого ключа может быть проверена равенством $e(P, Y) = e(Q, X)$.

Зашифрование n-битового сообщения $m \in \mathbb{Z}_p$ с помощью идентификатора ID_A и открытого ключа (X, Y) выполняется следующим образом: проверяется корректность открытого ключа, вырабатывается случайное значение $t \in_R \mathbb{Z}_p^*$ и формируется шифртекст

$$(U, v) = ([t]P,\ m \oplus h_2(e(h_1(\mathrm{ID}_A), Y)^t)).$$

Для расшифрования шифртекста (U, V) используется личный ключ sk_A:

$$m = v \oplus h_2(e(sk_A, U)).$$

3.2. IBE-системы с процедурой отзыва ключей

Наиболее удобным способом обновления ключей является принудительная периодическая замена ключей путём привязки их к определённому периоду времени, например к одной неделе.

Центр KGC, выступающий в данном случае в роли центра управления ключами, должен регулярно производить обновление ключей всех пользователей. При невозможности непосредственных контактов для этого необходимо организовать рассылку зашифрованных ключей. Однако такой способ оказывается неудобным для систем с большим числом пользователей, поскольку центру KGC в этом случае приходится регулярно выполнять массовую рассылку зашифрованной ключевой информации.

В работе [6] предложена оригинальная схема, позволяющая избежать массовой рассылки, в которой пользователи обращаются к общедоступному серверу центра KGC только в случае необходимости. Для этого использована конструкция, построенная на основе нечёткой системы шифрования Fuzzy IBE и структуры бинарного дерева [38].

Преимуществом систем шифрования на основе конструкции Fuzzy IBE является то, что шифртексты могут храниться на общедоступном открытом сервере. В конструкции Fuzzy IBE из [56], описанной в п. 3.1, пользовательские ключи и ключи, использованные для зашифрования текстов, связаны с наборами описательных атрибутов. Ключ пользователя может расшифровать тот или иной зашифрованный текст только в том случае, если у ключа зашифрования и ключа пользователя совпадает определённое количество атрибутов (так называемая «устойчивость к ошибкам»). Количество атрибутов, используемых для шифрования, и степень устойчивости к ошибкам определяются заранее.

Для защищённости от сговора требуется, чтобы разные пользователи, объединив свои атрибуты вместе, не смогли расшифровать зашифрованный текст, который ни

один из них не смог расшифровать по отдельности. Чтобы предотвратить сговоры, алгоритм генерации ключей Fuzzy IBE для каждого пользователя генерирует случайный полином, степень которого на единицу меньше, чем степень устойчивости к ошибкам, для каждого пользователя. Этот многочлен используется для вычисления ключей, соответствующих набору атрибутов. Поскольку все ключи вычисляются на разных полиномах, они не могут быть объединены каким-либо значимым образом.

В схеме IBE сообщения шифруются по двум атрибутам: идентификационной информации получателя и периоду времени. Ключ расшифрования также вычисляется для идентификационной информации атрибутов и времени с помощью полинома первой степени, что означает, что оба атрибута ключа расшифрования должны совпадать с атрибутами зашифрованного текста.

Ключ расшифрования каждого пользователя, соответствующий идентификатору и времени, разделён на два компонента: личный ключ и обновление ключа. Чтобы иметь возможность расшифровать зашифрованный текст, пользователю требуется как личный ключ, так и обновление ключа. Личный ключ выдаётся каждому пользователю центром KGC так же, как и обычные личные ключи в IBE. Обновление ключа публикуется центром KGC и является общедоступным для всех пользователей.

Таким образом, когда центру управления ключами KGC необходимо отозвать ключ пользователя, он может просто прекратить публикацию обновлений ключей для этого пользователя.

Чтобы избежать необходимости вычисления обновлений ключей для каждого пользователя отдельно, в [6] используется структура двоичного симметричного дерева высотой h, в котором каждому пользователю соответствует уникальный конечный узел — лист дерева. Каждому узлу дерева присвоен случайный многочлен.

Каждый пользователь получает ключи, соответствующие его идентификационной информации и вычисленные по полиномам всех узлов на пути от листа, соответствующего этому пользователю, к корню дерева. Чтобы иметь возможность расшифровать текст, зашифрованный в период времени t, пользователю достаточно получить обновления ключа, соответствующие t, для всех полиномов вершин на этом пути. Таким образом, когда ни один пользователь не отозван, центру ключей достаточно опубликовать обновление ключа, вычисленное на полиноме корня дерева. Когда отзывается подмножество пользователей, центр сначала находит минимальный набор вершин в дереве, который содержит предка (или сам узел) среди всех листьев, соответствующих неотозванным пользователям. Затем центр публикует обновления ключей по полиномам для вершин из этого набора.

Начальная установка.

Пусть $\mathbb{G}_1$ — группа простого порядка p с образующим P и операцией билинейного спаривания. Определим функцию

$$F_{P,J,H_1,\ldots,H_J}(x) = [x^2]P + \sum_{i=1}^{J}[\delta_{i,J}(x)]H_i,$$

где $\delta_{i,J}(x)$ определяется равенством (2).

Пусть имеется бинарное древо T. Каждой вершине соответствует некоторая строка, определяемая путём $\mathrm{Path}(v)$ от v к корню root дерева T. Если вершина v не является листом дерева, то обозначим через v_l и v_r соответственно её левого и правого сына.

Обозначим через $rl = \{(v_i, t_i)\}$ список вершин, ключи которых отозваны. В этом списке для каждой вершины v_i указывается время t_i, когда произведён отзыв ключа.

Требуется расшифровать текст в период времени t так, чтобы ключи расшифрования не зависели от вершин, ключи которых были отозваны до этого момента времени.

Определим функцию `KUNodes`, которая вычисляет минимальное множество вершин, для которого требуется обновить ключи так, чтобы пользователи, ключи которых не отозваны до момента времени t, могли расшифровать шифртекст. В качестве входов функции `KUNodes` выступают бинарное дерево T, время t и список rl вершин, ключи которых отозваны. В этом списке для каждой вершины указывается время, когда произошел отзыв ключа. Выходом является минимальное множество вершин дерева T, для которого ни одна из вершин из списка rl, у которой время отзыва не превосходит t, не имеет предшественников в этом множестве и все остальные листья имеют в этом множестве в точности одного предшественника.

Алгоритм вычисления функции `KUNodes` состоит в следующем. Сначала помечаются как отзываемые все предшественники отозванных до момента времени t вершин, а затем и все дети отзываемых вершин. На рис. 2 показан пример работы алгоритма вычисления функции `KUNodes` при отзыве ключей пользователя, соответствующего вершине u_3. Символом $\otimes$ помечены вершины, ключи которых отзываются, а символом $\checkmark$ — вершины, ключи которых подлежат обновлению.

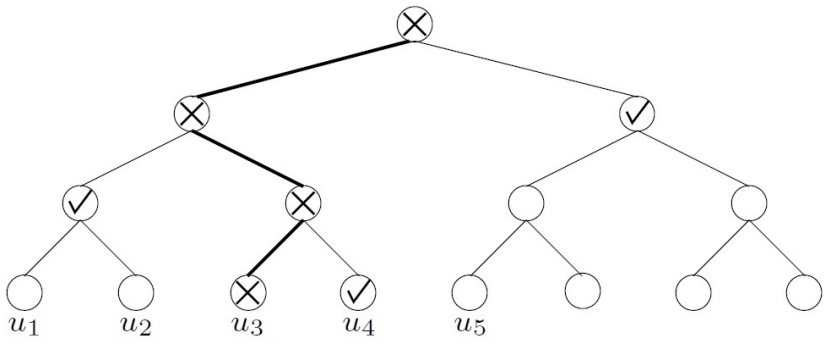

Рис. 2. Результат работы функции `KUNodes` при отзыве ключей пользователя u_3 [6]

Приведём описание алгоритма отзыва. Помимо дополнительных выходов rl и st, оно в точности повторяет процедуру из Fuzzy IBE.

Полагаем $rl = \varnothing$, и пусть T — бинарное дерево с n листьями. Пусть для примера $\mathbb{G}_1 = \langle P \rangle$, $|\mathbb{G}| = p$, $J \in \{1, 2, 3\}$, $a \in_R \mathbb{Z}_p$, $P_1 = [a]P$, $P_2, H_1, H_2, H_3 \in_R \mathbb{G}_1$.

Генерация закрытого ключа $\mathtt{SK}(pk, mk, \omega, st)$:

1) положить $pk = (P, P_1, P_2, H_1, H_2, H_3)$, $mk = a$, $st = T$;
2) взять непомеченную вершину-лист v из дерева T и приписать ω к этой вершине;
3) для всех $x \in Path(v)$:
   - если a_x не определено, то приписать вершине x значение $a_x \in_R \mathbb{Z}_p$;
   - $r_x \in_R \mathbb{Z}_p$;
   - $D_x = [a_x\omega + a]P_3 + [r_x]F_{P_2,J,H_1,H_2,H_3}(\omega)$;
   - $D'_x = [r_x]P$;
4) возвратить $sk_\omega = \{(x, D_x, D'_x)\}_{x \in Path(v)}$, st.

Заметим, что a_x фиксирует многочлен первой степени $q_x(y) = a_x y + a$ с условием $q_x(0) = a$, соответствующий вершине x. Алгоритм вычисляет ω-компоненты ключа расшифрования с помощью многочленов всех вершин на пути от листа, соответствующего ω, к корню дерева.

Генерация обновлённого ключа $\mathtt{KU}(pk, mk, t, rl, st)$:

1) положить $pk = (P, P_1, P_2, H_1, H_2, H_3)$, $mk = a$, $st = T$;
2) для всех $x \in \mathtt{KUNodes}(T, rl, t)$:
   - $r_x \in_R \mathbb{Z}_p$;
   - $E_x = [a_x t + a]P + [r_x]F_{P_2,J,H_1,H_2,H_3}(t)$;
   - $E'_x = [r_x]P$;
3) возвратить $ku_t = \{(x, E_x, E'_x) : x \in \mathtt{KUNodes}(T, rl, t)\}$.

Алгоритм сначала находит минимальное множество вершин, содержащее предшествующую вершину (или саму вершину) для всех неотзываемых вершин. Затем вычисляет t-компоненту ключа расшифрования с помощью многочленов всех вершин из этого множества.

Генерация ключа расшифрования (*Decryption Key Generation*) $DK(sk_\omega, ku_t)$:

1) положить $sk_\omega = \{(i, D_i, D'_i) : i \in I\}$, $ku_t = \{(j, E_j, E'_j) : j \in J\}$ для некоторых множеств вершин I, J;
2) для всех $(i, D_i, D'_i) \in sk_\omega, (j, E_j, E'_j) \in ku_t$:
   - если существуют $i \in I$, $j \in J$, такие, что $i = j$, то $dk_{\omega,t} = (D_i, E_j, D'_i, E'_j)$ (ключ создан);
   - в противном случае (т. е. sk_ω и ku_t не имеют ни одной общей вершины) $dk_{\omega,t} = \perp$ (символ $\perp$ указывает, что ключ отозван);
3) возвратить $dk_{\omega,t} = (D, E, D', E')$ (далее индексы i, j опускаем).

Алгоритм находит компоненты ключа sk_ω и ku_t, которые получены с использованием одинаковых многочленов.

Зашифрование $E(pk, \omega, t, m)$:

1) положить $pk = (P, P_1, P_2, H_1, H_2, H_3)$;
2) $z \in_R \mathbb{Z}_p$;
3) $c_1 = m \cdot e(P_1, P_2)^z$;
4) $C_2 = [z]P$;
5) $C_\omega = [z]F_{P_2,J,H_1,H_2,H_3}(\omega)$;
6) $C_t = [z]F_{P_2,J,H_1,H_2,H_3}(t)$;
7) возвратить $C = (\omega, t, C_\omega, C_t, c_1, C_2)$.

Алгоритм шифрования в точности повторяет IBE.

Расшифрование $D(dk_{\omega,t}, C)$:

1) положить $dk_{\omega,t} = (D, E, D', E')$, $C = (\omega, t, C_\omega, C_t, c_1, C_2)$;
2) $m = \left(\dfrac{e(D', C_\omega)}{e(D, C_2)}\right)^{t/(t-\omega)} \left(\dfrac{e(E', C_t)}{e(E, C_2)}\right)^{e(D,C_2)/(\omega-t)} \cdot c_1$;
3) возвратить m.

Алгоритм расшифрования такой же, как у Fuzzy IBE.

Отзыв (Revocation) $R(\omega, t, rl, st)$:

1) для любой вершины v, ассоциированной с идентификатором ω, добавить (v, t) к списку отозванных ключей rl;
2) возвратить rl.

Функция $\mathtt{KUNodes}$ должна вычисляться, только если изменяется содержание списка rl. Поэтому её значение должно быть сохранено и использоваться до следующего изменения списка rl.

Если число пользователей превышает ёмкость текущего дерева, то необходимо дополнить его непомеченным деревом такого же размера путём присоединения корневых

вершин обоих деревьев к новой корневой вершине. Тем самым число пользователей может быть увеличено в 2 раза. Каждый новый пользователь получает дополнительную компоненту личного ключа, определённую по старому или новому дереву, которая должна быть зашифрована с помощью соответствующей идентификационной информации с указанием времени и опубликована.

3.3. IDB-системы цифровой подписи

IDB-схема подписи Шамира на основе RSA

А. Шамир предложил схему цифровой IDB-подписи на основе RSA. Для подписи сообщения m пользователю A с открытым ключом $pk = h(\mathrm{ID}_A)$ и личным ключом $sk = h(\mathrm{ID}_A)^d \bmod n$ надо:

1) выбрать случайное число $r \in \mathbb{Z}_n$;
2) вычислить $t = r^e \bmod n$;
3) вычислить $f = h(t, m)$, где h — однонаправленная функция;
4) вычислить $s = sk \cdot r^f \bmod n$.

Значением подписи будет (s, t).

Для проверки подписи надо проверить выполнение равенства

$$s^e \stackrel{?}{=} pk \cdot t^{h(t,m)} \bmod n.$$

IDB-схема подписи GQ Гийу — Кискатера на основе RSA

L. C. Guillou и J.-J. Quisquater в 1999 г. [30] предложили модификацию схемы Шамира. Пусть центр KGC обладает закрытым мастер-ключом d и открытым ключом (n, e), $ed = 1 \,(\bmod n)$. Пользователь A получает в центре «тень» J_A своего идентификатора ID_A в качестве открытого ключа и RSA-подпись $S_A = J_A^{-d} \bmod n$ в качестве личного ключа. Ключ J_A формируется с учётом внесения избыточности в идентификационную информацию ID_A для того, чтобы не проходила (экзистенциональная) атака, позволяющая строить новые ключевые пары путём возведения в степень и умножения значений из известных противнику ключевых пар.

Алгоритм вычисления подписи $\sigma = (s, t)$ под сообщением m:

1) $r \in_R \mathbb{Z}_n$;
2) $u = r^e \bmod n$;
3) $t = J_A^m u^{e^k} \bmod n$, где k выбирается из условия $e^{k-1} \leqslant m \leqslant e^k$;
4) $s = rS_A^t$.

Для проверки подписи надо вычислить $u = J_A^t s^e$ и проверить выполнение равенства

$$t \stackrel{?}{=} J_A^m u^{e^k}.$$

Модифицированный вариант этой схемы [30] вошёл в международный стандарт ISO/IEC 14888-2: 1999.

IDB-схема подписи GQ1 на основе RSA из ISO/IEC 14888-2: 2008

Рассмотрим вариант GQ1 IDB-схемы цифровой подписи из последней версии этого стандарта. Пусть $n = p_1 \cdot \ldots \cdot p_f$ — произведение различных простых чисел; v — простое число (верификационная экспонента), $v < n$.

Формирование ключей владельца подписи.

Ключ проверки подписи формируется как элемент $G \in \mathbb{Z}_n$, каким-то образом сопоставленный идентификационной информации пользователя: $\mathrm{ID} \mapsto G$.

Ключ подписи $Q \in \mathbb{Z}_n$ можно вычислить двумя способами так, что полученные в результате числа G и Q удовлетворяют условию

$$G \cdot Q^v \bmod n = 1.$$

Способ 1 (с применением китайской теоремы об остатках (CRT, Chinese Remainder Theorem)):

1) для $i = 1, \ldots, f$:
   - найти число s_i как наименьшее положительное число, такое, что $v\,s_i - 1$ кратно $p_i - 1$;
   - вычислить $u_i = p_i - 1 - s_i$, $G_i = G \bmod p_i$, $Q_i = G_i^{u_i} \bmod p_i$;
2) найти $Q = \mathrm{CRT}(Q_1, \ldots, Q_f)$ с помощью китайской теоремы об остатках.

Способ 2 (без применения CRT):

1) найти число s как наименьшее положительное число, такое, что $v\,s - 1$ кратно наибольшему общему делителю $(p_1 - 1, \ldots, p_f - 1)$;
2) вычислить $u = (p_1 - 1, \ldots, p_f - 1) - s$;
3) вычислить $Q = G^u \bmod n$.

Алгоритм подписи:

1) выбрать случайные числа $r = (r_1, \ldots, r_t)$;
2) для $i = 1, \ldots, t$ вычислить $r_i^v \bmod n$ и сопоставить им битовую строку W;
3) вычислить $H = h(W||M)$, пусть R — начальный отрезок из $t(l(v) - 1)$ битов строки H, где $l(v)$ — битовая длина v;
4) для $i = 1, \ldots, t$ вычислить $r_i \cdot Q^{R_i} \bmod n$, сопоставить им битовую строку S.

Подписью является пара (R, S).

Алгоритм проверки подписи:

1) разделить битовые строки R и S на компоненты и выделить соответствующие числа r_i и s_i, $i = 1, \ldots, t$;
2) для $i = 1, \ldots, t$ вычислить $S_i^v \cdot G^{R_i} \bmod n$, сопоставить им битовую строку W^*;
3) вычислить $H^* = h(W^*||M)$, пусть R^* — начальный отрезок из $t(l(v) - 1)$ битов строки H^*;
4) проверить равенство $R^* \stackrel{?}{=} R$.

IBS-1-схема подписи на основе ECDLOG из ISO/IEC FDIS 14888-3: 2018

IDB-схема подписи IBS-1 строится на основе группы точек эллиптической кривой и операции билинейного спаривания. Этот механизм основан на алгоритме, разработанном в [35]. Пусть $\mathbb{G}_1 = \langle P \rangle$ — циклическая подгруппа порядка q группы точек эллиптической кривой E над полем $\mathrm{GF}(p^m)$.

Мастер-ключом центра KGC является ключевая пара (U, V), где открытый ключ U — случайное число в интервале $0 < U < q$, а закрытый ключ вычисляется по формуле $V = [U]P$.

Ключ формирования и проверки подписи пользователя — это ключевая пара (X, Y), где $Y = h_1(\mathrm{ID})$ — ключ проверки подписи, полученный из идентификационных данных владельца ID с помощью хеш-функции h_1, а X — личный ключ подписи, вычисляемый центром KGC: $X = [U]Y$.

Конкретные параметры схемы подписи — $\mathbb{G}_1, \mathbb{G}_2, P, q, e(,), h_1$ и h_2 — определены в [70]. Здесь $e(,)$ обозначает операцию билинейного спаривания $e : \mathbb{G}_1^2 \to \mathbb{G}_2$, которая предполагает наличие в мультипликативной подгруппе поля, являющегося некоторым расширением поля $\mathrm{GF}(p^m)$, циклической подгруппы $\mathbb{G}_2$ порядка q.

Алгоритм формирования подписи:

1) выбрать случайное целое k, $0 < k < q$, и сохранить его в секрете;
2) вычислить $\Pi = e(X, P)^k$.
   З а м е ч а н и е. Элемент Π принадлежит расширению поля $\mathrm{GF}(p^m)$ степени 4 для характеристики $p = 2$, степени 6 для характеристики $p = 3$ и степени 2 для характеристики $p > 3$;
3) при $p > 3$ вычислить
$$R = h_2(M \| \Pi_a \| \Pi_b) \bmod q,$$
   где $\Pi = (\Pi_a, \Pi_b) \in \mathrm{GF}(p^{2m})$. Если $R = 0$, то перейти к п. 1.
   Для полей более высокой степени расширения следует рассматривать больше компонент в строке Π. Например, для степени расширения 4: $\Pi = (\Pi_a, \Pi_b, \Pi_c, \Pi_d)$;
4) вычисление второй части подписи:
$$S = [k - R]X.$$

Подписью является $\Sigma = (R, S)$.

Алгоритм проверки подписи:

1) проверить $S \in \mathbb{G}_1$; если нет, то подпись неверна;
2) вычислить $\Pi' = e(S, P) * e(Y, V)^R$.
   З а м е ч а н и е . Значение спаривания $e(Y, V)$ может быть вычислено заранее;
3) вычислить
$$R' = h_2(M \| \Pi'_a \| \Pi'_b) \bmod q;$$
4) проверить свидетельство $R' \stackrel{?}{=} R$. Если да, то подпись верна, иначе неверна.

IDB-схема подписи BLMQ

Эта схема, основанная на операции билинейного спаривания и описанная в работе П. Баретто, Б. Либерта, Н. МакКулаха и Дж. Кискатера (P. Barreto, B. Libert, N. McCullagh и J. Quisquater) [5] в 2005 г., была опубликована, а затем предложена ими для включения в стандарт IEEE P1363.3 [4].

Стойкость схемы основана на трудности проблемы k-обращения Диффи — Хеллмана k-DHI (k-Diffie-Hellman Inversion) для групп $(\mathbb{G}_1, \mathbb{G}_2)$: для заданного $(k+2)$-набора $(P, Q, aQ, a^2Q, \ldots, a^kQ)$ найти $a^{-1}P$, где $P \in \mathbb{G}_1$; $Q \in \mathbb{G}_2$; $a \in_R \mathbb{Z}_q^*$; q — порядок этих групп.

Пусть $h_1, h_2 : \{0,1\}^* \to \mathbb{Z}_q^*$ — хеш-функции, $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ — операция билинейного спаривания. Открытый ключ центра: $P, Q, sQ, g = e(P, Q)$. Закрытый ключ центра: $s \in \mathbb{Z}_q$. Пользователь A получает в центре личный ключ

$$S_{\mathrm{ID}_A} = (h_1(\mathrm{ID}_A) + s)^{-1} P.$$

Подпись к сообщению m — это (H, S), где $H = h_2(m, r)$; $r = g^x$; $S = (x + H)S_{\mathrm{ID}_A}$; $x \in_R \mathbb{Z}_q^*$. Проверка подписи (H, S) заключается в проверке равенства

$$H \stackrel{?}{=} h_2(m, e(S, h_1(\mathrm{ID}_A)Q + sQ)g^{-h}).$$

IDB-схема подписи Уотерса

Схема IDB-шифрования, основанная на симметричном спаривании, предложенная в работе B. Waters в 2005 г. [66] и рассмотренная в п. 3.1, может быть преобразована в схему цифровой подписи следующим образом.

Пусть имеется операция билинейного спаривания $e\colon \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$, $\mathbb{G}_1 = \langle P_0 \rangle$. Закрытый ключ центра — случайный элемент $x \in_R \mathbb{Z}_p^*$.

Центр KGC вырабатывает случайные $P_2, U', U_1, \ldots, U_n \in_R \mathbb{G}_1$ и выдаёт пользователю личный ключ $[x]P_2$.

Для формирования подписи сообщение $m \in \{0,1\}^*$ сначала сжимается криптографической хеш-функцией $h\colon \{0,1\}^* \to \{0,1\}^n$: $h(m) = (m_1, \ldots, m_n)$, затем пользователь вырабатывает случайный элемент $r \in \mathbb{Z}_p$ и с помощью полученного в центре KGC личного ключа вычисляет

$$\mathrm{Sig}(m) = ([x]P_2 + [r]\left(U' + \sum_{i:m_i=1} U_i\right),\ [r]P_0).$$

Для проверки подписи $\mathrm{Sig}(m) = (\sigma_1, \sigma_2)$ к сообщению m с помощью открытого ключа $P_2, U', U_1, \ldots, U_n$ надо проверить равенство

$$e(\sigma_1, P_0)/e\left(\sigma_2, U' + \sum_{i:m_i=1} U_i\right) = e(P_0, P_2).$$

Описание других схем подписи на основе операции билинейного спаривания в группе точек эллиптической кривой, например предложенные Патерсоном (Paterson) и др., можно найти в обзоре [27].

IDB-система подписи с недоверенным центром KGC

В работе [18] Ч. Чен, Ф. Занг и К. Ким (X. Chen, F. Zhang и K. Kim) предложили вариант схемы подписи, позволяющий пользователю доказывать с нулевым разглашением арбитру, что центр KGC совершил атаку по (экзистенциональной) подмене подписи.

Пусть имеется операция билинейного спаривания $e\colon \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{Z}_q$, $\mathbb{G}_1 = \langle P \rangle$, две хеш-функции $h_1 : \{0,1\}^* \times \mathbb{G}_1 \to \mathbb{G}_1$ и $h_2 : \{0,1\}^* \times \mathbb{G}_1 \to \mathbb{Z}_q$.

Пользователь A выбирает $r \in_R \mathbb{Z}_q^*$ в качестве своего долговременного секрета и отправляет центру значение rP в качестве открытого ключа. Центр вычисляет $Q_A = h_1(\mathrm{ID}_A \| t, rP)$ и $S_A = sQ_A$, где t — срок действия, а затем передаёт их пользователю. Открытым ключом пользователя будет ID_A, а личным ключом — (S_A, s).

Алгоритм формирования подписи для сообщения m:

1) $a \in_R \mathbb{Z}_q$, $U = aQ_A$;
2) $V = rh_2(m, U)$;
3) $h = h_2(m, U + V)$;
4) $W = (a + h)S_A$.

Подписью является набор $\sigma = (U, V, W, t, rP) \in \mathbb{G}_1^3 \times \{0,1\}^* \times \mathbb{G}_1$.

Для проверки подписи надо вычислить $Q_A = h_1(\mathrm{ID}_A \| t, rP)$, $h_1(m, U)$ и $h = h_2(m, U + V)$, а затем проверить выполнимость равенств

$$e(W, P) = e(U + hQ_A, Q), \quad e(V, P) = e(h_1(m, U), rP).$$

Если центр выполнит атаку по подмене подписи для сообщения m следующим образом: вычислит $r' \in_R \mathbb{Z}_q$ и $Q'_A = h_2(\mathrm{ID}_A \| t, r'P)$, а затем сформирует подпись в соответствии с приведённым алгоритмом

$$\sigma' = (U', V', W', t, r'P),$$

то проверка подписи будет успешной, однако пользователь сможет доказать арбитру с нулевым разглашением на основе знания своего личного ключа, что это не его подпись.

IDB-системы одновременного вычисления цифровой подписи и шифрования (IBSigncryption)

Протокол Зенга. В 1997 г. Zheng в работе [68] предложил одну из первых схем одновременного вычисления цифровой подписи и шифрования, более быстрого, чем их последовательное вычисление.

Общими параметрами являются:

p — большое простое число;

q — делитель $(p-1)$;

$g \in_R \mathbb{Z}_p$, $g^q = 1 \bmod p$;

h — однонаправленная хеш-функция (с не менее чем 128-битовым выходом);

$\mathtt{KH}_k()$ — хеш-функция, зависящая от ключа;

(E, D) — алгоритмы симметричного зашифрования и расшифрования.

Пусть также:

— личным ключом A является $x_a \in_R \{1, \ldots, q-1\}$, открытым — $y_a = g^{x_a} \bmod p$;

— личным ключом B является $x_b \in_R \{1, \ldots, q-1\}$, открытым — $y_b = g^{x_b} \bmod p$.

Протокол передачи сообщения m имеет вид

$$A \to B: \ (c, r, s).$$

Алгоритм формирования подписи и зашифрования:

1) сторона A выбирает $x \in_R \{1, \ldots, q-1\}$, вычисляет $k = h(y_b^x \bmod p)$ и выделяет из k ключи $k = k_1 \| k_2$;
2) вычисляет $c = E_{k_1}(m)$, $r = \mathtt{KH}_{k_2}(m)$, $s = x/(r + x_a) \bmod q$;
3) отправляет стороне B подписанный шифртекст (c, r, s).

Алгоритм расшифрования и проверки подписи:

1) сторона B восстанавливает k из c, r, s, g, p, x_a и x_b:

$$k = h((y_a \cdot g^r)^{s x_b} \bmod p)$$

   и выделяет из k ключи k_1 и k_2;
2) вычисляет $m = D_{k_1}(c)$;
3) признаёт m подлинным сообщением от A в том и только в том случае, когда $r = \mathtt{KH}_{k_2}(m)$.

Этот протокол не защищён от чтения назад, так как если противник восстановит долговременный ключ x_a, то в силу равенства

$$h((y_a \cdot g^r)^{s x_b} \bmod p) = h((y_b^{x_a + r})^s \bmod p)$$

он сможет вычислить ключ $k = h((y_b^{x_a+r})^s \bmod p)$.

Протокол Нэйла — Ридди. На базе данной схемы D. Nalla и K. C. Reddy в работе [47] предложили IDB-схему одновременного вычисления цифровой подписи и шифрования, основанную на операции билинейного спаривания. Пусть $\mathbb{G}_1$ — подгруппа группы точек эллиптической кривой порядка q, для которой определена операция билинейного спаривания Вейля $\hat{e}\colon \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$; $\mathbb{G}_2$ — мультипликативная группа того же порядка q. Предположим также, что имеются хеш-функция $h'\colon \{0,1\}^* \to \mathbb{Z}_q^*$, функция вычисления ключа шифрования $h''\colon \mathbb{Z}_q^* \to \{0,1\}^*$ и псевдослучайная функция $h_1\colon \mathbb{G}_2 \to \{0,1\}^*$.

Стороны A и B получают в центре KGC ключевые пары (Q_A, S_A) и (Q_B, S_B), сформированные на основе своих идентификаторов, $S_A = [s]Q_A$, $S_B = [s]Q_B$, где $s \in_R \mathbb{Z}_q^*$ — закрытый мастер-ключ центра.

Для отправления стороне B подписанного сообщения $m \in \{0,1\}^*$ сторона A использует открытые ключи (Q_A, Q_B) и свой личный ключ S_A.

Алгоритм формирования подписи и зашифрования:

1) сторона A выбирает случайный элемент $a \in_R \mathbb{Z}_q^*$;
2) вычисляет

$$R = [a]S_A, \quad d = h'(R||h_1(\hat{e}(Q_B, S_A))||m), \quad S = [ad]Q_A,$$
$$k_A = h''(\hat{e}(Q_B, S_A)^{ad}), \quad c = k_A \oplus m;$$

3) отправляет (R, S, c) стороне B.

Алгоритм расшифрования и проверки подписи.

Для проверки полученного сообщения сторона B, используя полученные значения (R, S, c, Q_A, Q_B) и свой личный ключ S_B:

1) вычисляет $k_B = h''(\hat{e}(S_B, S))$ и $m = k_B \oplus c$;
2) вычисляет $d' = h'(R||h_1(\hat{e}(S_B, Q_A))||m)$ и принимает m, только если выполнено равенство $\hat{e}(S_B, S) = \hat{e}(Q_B, R)^{d'}$. В противном случае B прерывает протокол.

Данный протокол защищён от чтения назад и является вычислительно более эффективным.

IDB-системы одновременного вычисления цифровой подписи и шифрования с сокрытием идентификаторов

Рассмотрим протокол IBHigncryption (от Id-Based Higncryption). Термин «higncryption» означает identity-hiding signcryption, т. е. одновременное вычисление цифровой подписи и шифрования с сокрытием идентификаторов.

Пусть имеется операция симметричного билинейного спаривания $e\colon \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ для группы $\mathbb{G}_1 = \langle P \rangle$ порядка q и $h : \{0,1\}^* \to \mathbb{G}_1$ — криптографическая хеш-функция. Центр выбирает закрытый ключ $msk = s \in_R \mathbb{Z}_q^*$. Открытыми параметрами являются $(q, \mathbb{G}_1, \mathbb{G}_T, e, P, h)$.

Пользователь A применяет в качестве открытого ключа идентификатор $pk_A = \mathrm{ID}_A$ и получает в центре $sk_A = [s]h(\mathrm{ID}_A)$ в качестве личного ключа.

Пусть (k, E, D) — некоторая схема аутентифицированного шифрования с ассоциированными данными. Ключ k вырабатывается с использованием функции выработки производного ключа $\mathtt{KDF}\colon \mathbb{G}_T \times \{0,1\}^* \to K$.

Для зашифрования и подписи сообщения $n \in \{0,1\}^*$ с сокрытием идентификаторов отправитель A:

1) выбирает случайный $x \in_R \mathbb{Z}_q^*$ и вычисляет $X = [x]h(\mathrm{ID}_A) \in \mathbb{G}_1$;

2) вычисляет предварительный секрет $PS = e(sk_A, h(\mathrm{ID}_B))^x \in \mathbb{G}_T$;
3) вычисляет ключ для AE шифрования $k = \mathtt{KDF}(PS, X \| \mathrm{ID}_A)$;
4) вычисляет $c_{AE} = E_k(H, \mathrm{ID}_A \| n \| x)$ с ассоциированными данными $H \in \{0,1\}^*$;
5) отправляет получателю B сообщение $C = (H, X, c_{AE})$.

Для расшифрования полученного сообщения $C = (H, X, c_{AE})$ и проверки подписи получатель B:

1) вычисляет $PS = e(X, sk_B) \in \mathbb{G}_T$ и ключ $k = \mathtt{KDF}(PS, X \| \mathrm{ID}_B)$;
2) расшифровывает $D_k(H, c_{AE})$ с проверкой целостности;
3) получает ID_A, n, x и проверяет равенство $X = [x]h(\mathrm{ID}_A)$. Если всё правильно, то принимает сообщение m, в противном случае прерывает протокол.

Подписи на основе сертификатов

Цифровые подписи на основе сертификатов (Certificate-Based Signature, CBS) строятся аналогично CBE-системам. В них сертификат, построенный на основе открытого ключа и идентификационной информации пользователя, используется как составная часть ключа подписи, составленного непосредственно из сгенерированного пользователем личного ключа и полученного сертификата.

Впервые такая система предложена в [37], однако позднее в работе [41] найдена атака и построен исправленный вариант. В [3] предложена схема анонимной циклической подписи на основе сертификатов. Все указанные схемы построены с использованием операции билинейного спаривания.

Рассмотрим предложенную в работе [43] CBC-схему цифровой подписи, не использующую операцию билинейного спаривания. Пусть $\mathbb{G}$ — мультипликативная группа порядка q. Центр KGC выбирает случайный образующий элемент $g \in_R \mathbb{G}$ и случайное число $x \in_R \mathbb{Z}_q^*$, являющееся его мастер-ключом. Пусть $X = g^x$ и $h : \{0,1\}^* \to \mathbb{Z}_q^*$ — криптографическая хеш-функция. Параметрами схемы являются $(\mathbb{G}, q, g, X, h)$.

Пользователь A выбирает $u \in_R \mathbb{Z}_q^*$ в качестве своего личного ключа sk_A и вычисляет открытый ключ $pk_A = (g^u, X^u, \pi_u)$, где π_u — неинтерактивное доказательство знания значения дискретного логарифма $u = \log_g U_1 = \log_X U_2$ для элементов $U_1 = g^u$ и $U_2 = X^u$ (например, на основе протокола аутентификации Шнорра).

Идентификационной информацией пользователя служит значение $\tilde{H} = h(pk_A, \mathrm{ID}_A)$, где pk_A и ID_A — открытый ключ и идентификатор пользователя A. Сертификационный центр CA выбирает случайно $r \in_R \mathbb{Z}_q^*$ и вычисляет сертификат (R, s) пользователя A (фактически сертификат (R, s) является цифровой подписью Эль-Гамаля под сообщением $\tilde{H}$, связывающим значения pk_A и ID_A):

$$R = g^r, \qquad s = r^{-1}(\tilde{H} - xR) \bmod q.$$

Корректность сертификата проверяется равенством

$$R^s X^R = g^{\tilde{H}}.$$

Для подписания сообщения $m \in \{0,1\}^*$ пользователь использует сертификат (R, s) и личный ключ u, а также случайный элемент $y \in_R \mathbb{Z}_q^*$ и вычисляет значение подписи $\sigma = (Y, H, z)$:

$$Y = R^{-y}, \quad H = h(Y, R, m), \quad z = (y + Hsu) \bmod q.$$

Проверка подписи осуществляется следующим образом. Сначала проверяется, что π_u действительно является неинтерактивным доказательством знания пользователем с идентификатором ID_A и открытым ключом pk_A значения ключа u. Если оно

прошло проверку, то вычисляются $H = h(Y, R, m)$ и $\tilde{H} = h(pk_A, ID_A)$, а затем проверяется равенство

$$(g^u)^{\tilde{H}} \stackrel{?}{=} R^z Y (X^u)^{HR}.$$

Если оно выполнено, то подпись признаётся истинной.

3.4. IDB-системы аутентификации сторон

Протоколы аутентификации сторон обычно строятся на основе протоколов цифровой подписи, использующих технику доказательства знания без разглашения секретов. Рассмотрим пример такого протокола.

IDB-система аутентификации сторон GQ

В 1988 г. Л. Гийу и Ж. Кискатер [29] предложили протокол аутентификации сторон на основе схемы RSA, вариант GQ1 которого [31] позже вошёл в международный стандарт ISO/IEC 9798-5:2009, а соответствующий ему IDB-протокол цифровой подписи — в рассмотренный выше стандарт ISO/IEC 14888-2:2008.

Пусть $n = pq$ и числа e, d удовлетворяют условию $ed = 1 \pmod{\varphi(n)}$. Число d известно только центру, выдающему ключи. Пусть h — хеш-функция, известная всем участникам. Каждый участник A получает в центре личный ключ $u = h(ID_A)^{-d} \in \mathbb{Z}_n$ и открытый ключ $v = h(\mathrm{ID}_A)$. Тем самым выполняется равенство $v = (u^{-1})^e \bmod n$, и мы можем использовать ранее рассмотренный протокол:

$$\begin{aligned} A \to B: &\quad A, \gamma = r^e \bmod n, \\ A \leftarrow B: &\quad x, \\ A \to B: &\quad y = ru^x \bmod n. \end{aligned}$$

Для проверки правильности B использует равенство

$$\gamma = v^x y^b \bmod n.$$

3.5. IDB-протоколы выработки общего ключа

IBKE-протокол Окамото на основе RSA

Первый IDB-протокол выработки общего ключа (Identity-Based Key Exchange) на основе RSA предложен Е. Окамото (E. Okamoto) в 1987 г. [48]. KGC имеет открытый ключ (e, n), а число d, такое, что $ed = 1 \pmod{\varphi(n)}$, KGC хранит в секрете. Пусть g_0 — образующий элемент мультипликативной группы $\mathbb{Z}_n^*$.

Пользователь $X \in \{A, B\}$ имеет открытый ключ ID_X и получает в центре KGC личный ключ s_X, удовлетворяющий условию $(s_X)^e \mathrm{ID}_X = 1 \bmod n$. Значение s_X центр KGC вычисляет по формуле $s_X = (1/\mathrm{ID}_X)^d \bmod n$, где ID_X — идентификационные данные пользователя X.

Протокол:

$$\begin{aligned} A: &\quad r_A \in_R \mathbb{Z}_n^*, \\ A \to B: &\quad t_A = s_A g^{r_A} \bmod n, \\ B: &\quad r_B \in_R \mathbb{Z}_n^*, \\ A \leftarrow B: &\quad t_B = s_B g^{r_B} \bmod n. \end{aligned}$$

Теперь A и B вычисляют общий ключ $k = g^{er_A r_B}$ соответственно по формулам

$$k_A = (t_B^e \mathrm{ID}_B)^{r_A}, \qquad k_B = (t_A^e \mathrm{ID}_A)^{r_B}.$$

IBAKE-протокол с обеспечением частичной аутентификации ключа

Протокол, предложенный Е. Окамото и К. Танака (E. Okamoto, K. Tanaka) в 1989 г. [50], является вариантом предыдущего. В нём используется хеш-функция $h : \{0,1\}^* \to \mathbb{Z}_{n-1}^*$ для обеспечения частичной аутентификации ключа. Обозначим через T_A, T_B метки времени.

Протокол:

$$\begin{array}{rl}
A: & r_A \in_R \mathbb{Z}_n^*, u_A = g^{er_A} \bmod n,\\
A: & c_A = h(u_A, \mathrm{ID}_A, \mathrm{ID}_B, T_A),\\
A \to B: & u_A, v_A = s_A g^{c_A r_A} \bmod n,\\
B: & c_A = h(u_A, \mathrm{ID}_A, \mathrm{ID}_B, T_A),\\
B: & \text{проверяет } \mathrm{ID}_A \stackrel{?}{=} u_A^{c_A}/v_A^e,\\
B: & r_B \in_R \mathbb{Z}_n^*, u_B = h^{er_B} \bmod n,\\
B: & c_B = h(u_B, \mathrm{ID}_B, \mathrm{ID}_A, T_B),\\
A \leftarrow B: & u_B, v_B = s_B g^{c_B r_B} \bmod n,\\
A: & c_B = h(u_B, \mathrm{ID}_B, \mathrm{ID}_A, T_B),\\
A: & \text{проверяет } \mathrm{ID}_B \stackrel{?}{=} u_B^{c_B}/v_B^e.
\end{array}$$

Теперь A и B вычисляют общий ключ $k = g^{er_A r_B}$ соответственно по формулам

$$k_A = u_B^{r_A}, \qquad k_B = u_A^{r_B}.$$

Два предыдущих протокола не защищены от нечестного центра KGC, так как он имеет возможность восстановить личные ключи всех пользователей, а также не обеспечивают защиты от чтения назад при компрометации мастер-ключа центра KGC. Поэтому необходимо обеспечить защиту пользователя от центра, предоставив ему возможность самому формировать личные ключи.

IBKE-протокол с защитой личного ключа пользователя от центра KGC

М. Гиро и Дж. Пале (M. Girault, J. Paillès) в [25] предложили модификацию протокола Окамото, в которой предусмотрена защита личного ключа пользователя от центра KGC. Пользователь A передаёт в центру KGC значение $g^{x_A} \bmod n$, где $sk_A = x_A \in \mathbb{Z}_n^*$ — выбранный им самостоятельно и сохраняемый в секрете личный ключ, и получает открытый ключ $pk_A = y_A = \mathrm{ID}_A^{-d} g^{-x_A} \bmod n$, корректность которого он может проверить из условия $(y_A)^e \mathrm{ID}_A = g^{-ex_A} \bmod n$ (заметим, что A может вычислить y_A самостоятельно по значению s_A из протокола Окамото).

Протокол:

$$\begin{array}{rl}
A: & r_A \in_R \mathbb{Z}_n^*,\\
A \to B: & t_A = y_A g^{x_A - r_A} \bmod n,\\
B: & r_B \in_R \mathbb{Z}_n^*,\\
A \leftarrow B: & t_B = y_B g^{x_B - r_B} \bmod n.
\end{array}$$

Теперь A и B вычисляют общий ключ $k = g^{-er_A r_B}$ аналогично протоколу Окамото по формулам

$$k_A = (t_B^e \mathrm{ID}_B)^{r_A}, \qquad k_B = (t_A^e \mathrm{ID}_A)^{r_B}.$$

В результате выполнения протокола получается ключ, отличающийся знаком от того, который вычисляется в протоколе Окамото. Однако так как ключ не зависит от личных ключей участников, данный протокол не имеет преимуществ по сравнению с протоколом Окамото.

IBKE-протокол с самосертифицируемыми открытыми ключами

В 1991 г. М. Гиролт (M. Girault) предложил следующий протокол [26]. Пользователь A передаёт центру KGC значение $g^{x_A} \bmod n$, где $sk_A = x_A \in \mathbb{Z}_n^*$ — выбранный им самостоятельно и сохраняемый в секрете личный ключ, и получает открытый ключ $pk_A = y_A$, удовлетворяющий условию $y_A = (g^{x_A} - \mathrm{ID}_A)^d \bmod n$.

Центр KGC также не может узнать значение x_A, но теперь с помощью проверки равенства $g^{x_A} = y_A^e + \mathrm{ID}_A \pmod n$ каждый может убедиться в правильности открытого ключа.

Протокол:

$$\begin{array}{rl} A: & r_A \in_R \mathbb{Z}_n^*, \\ A \to B: & t_A = g^{r_A} \bmod n, \\ B: & r_B \in_R \mathbb{Z}_n^*, \\ A \leftarrow B: & t_B = g^{r_B} \bmod n. \end{array}$$

Теперь A и B вычисляют общий ключ $k = g^{r_A x_B + r_B x_B}$ аналогично протоколу MTI/A0 [44] по формулам

$$k_A = t_B^{x_A}(y_B^e + \mathrm{ID}_B)^{r_A}, \qquad k_B = t_A^e(y_A^e + \mathrm{ID}_A)^{r_B}.$$

IBKE-протокол выработки общего ключа на основе цифровой подписи

К. Гюнтер (C. G. Günther) в [32] предложил протокол выработки общего ключа на основе цифровой подписи, сформированной центром KGC. Закрытым и открытым ключами центра являются элементы $x_T \in \{1, \ldots, p-1\}$ и $y_T = g^{x_T} \in \mathbb{Z}_p^*$, где p — большое простое число; g — образующий элемент группы $\mathbb{Z}_p^*$.

Пользователь A получает в KGC цифровую подпись (u_A, v_A) по схеме Эль-Гамаля для своего идентификатора $\mathrm{ID}_A \in \mathbb{Z}_p^*$:

$$u_A = g^{k_A}, \quad v_A = (\mathrm{ID}_A - x_T u_A) k_A^{-1} \bmod (p-1),$$

где $k_A \in_R \mathbb{Z}_p^*$; $(k_A, p-1) = 1$. Проверка подписи проводится с помощью уравнения

$$u_A^{v_A} = g^{\mathrm{ID}_A} y_T^{-u_A}.$$

В роли открытого ключа пользователя A выступает его идентификатор ID_A и первая половина подписи u_A, а в роли личного ключа — вторая половина подписи $v_A = \log_{u_A}(g^{\mathrm{ID}_A} y_t^{-u_A})$. Для выработки общего ключа стороны A и B выбирают случайные элементы $r_A, r_B \in_R \mathbb{Z}_p^*$ и выполняют следующий протокол:

$$\begin{array}{rl} A \to B: & \mathrm{ID}_A, u_A, \\ A \leftarrow B: & \mathrm{ID}_B, u_B, \\ A \to B: & w_A = u_B^{r_A}, \\ A \leftarrow B: & w_B = u_A^{r_B}. \end{array}$$

Теперь A и B могут вычислить общий ключ соответственно по формулам

$$k_A = w_B^{v_A}(g^{\mathrm{ID}_B} y_T^{-u_B})^{r_A}, \quad k_B = w_A^{v_B}(g^{\mathrm{ID}_A} y_T^{-u_A})^{r_B}.$$

В результате формируется общий ключ $k = w_B^{v_A} w_A^{v_A} = g^{k_B r_A v_B + k_A v_A r_B}$.

Данный протокол также нестоек к атаке чтения назад, так как при компрометации долговременных ключей v_A, v_B имеется возможность определения разовых общих ключей по передаваемым сообщениям w_B, w_A:

$$k = w_B^{v_A} w_A^{v_B}.$$

В качестве улучшения этого протокола с одновременным сокращением числа передаваемых сообщений С. Саедниа (S. Saeednia) в 2000 г. [52] предложил следующую модификацию:

$$\begin{aligned} A \to B: &\quad \mathrm{ID}_A, u_A, t_A = g^{r_A}, \\ A \leftarrow B: &\quad \mathrm{ID}_B, u_B, t_B = g^{r_B}. \end{aligned}$$

Улучшение достигнуто за счёт изменения формулы для вычисления второй части подписи Эль-Гамаля

$$v_A = \mathrm{ID}_A k_A - x_T u_A \bmod (p-1)$$

и проверочного соотношения

$$g^{v_A} = u_A^{\mathrm{ID}_A} y_T^{u_A}.$$

Стороны A и B могут вычислить общий ключ соответственно по формулам

$$k_A = t_B^{v_B} (u_B^{\mathrm{ID}_B} y_S^{u_B})^{r_A}, \quad k_B = t_A^{v_B} (u_A^{\mathrm{ID}_A} y_S^{u_A})^{r_B}.$$

В результате получается $k = g^{v_B r_A + v_A r_B}$.

Протоколы выработки общего ключа на основе операции билинейного спаривания

IBKE-протокол Сакаи — Огиши — Казахара

Протокол типа SK предложен R. Sakai, K. Ohgishi и M. Kasahara в 2001 г. [54]. В нём используется операция билинейного спаривания с *разными* группами $\mathbb{G}_1$ и $\mathbb{G}_2$, для которых имеются хеш-функции $h_1\colon \{0,1\}^* \to \mathbb{G}_1$ и $h_2\colon \{0,1\}^* \to \mathbb{G}_2$.

Центр KGC обладает ключевой парой (s, Q_{pub}), $Q_{\mathrm{pub}} = [s]P$, $P \in \mathbb{G}_2$.

Протокол является неинтерактивным и не предполагает обмена сообщениями, аналогично статичному протоколу Диффи — Хеллмана.

Первый вариант протокола использует только одну хеш-функцию h_2 и предполагает наличие гомоморфизма $\psi\colon \mathbb{G}_2 \to \mathbb{G}_1$. Сторона A получает открытый ключ $Q_A = h_2(\mathrm{ID}_A)$ и личный ключ $S_A = [s]Q_A$.

Общий ключ $k = e(\psi(Q_A), Q_B)^s$ вычисляется сторонами по формулам

$$k = e(\psi(S_A), Q_B) = e(\psi(S_B), Q_A).$$

Второй вариант протокола не использует гомоморфизм. Сторона A получает открытый ключ $(Q_A, Q'_A) = (h_1(\mathrm{ID}_A), h_2(\mathrm{ID}_A))$ и личный ключ $(S_A, S'_A) = ([s]Q_A, [s]Q'_A)$.

Общий ключ $k = e(Q_A, Q'_B)^s \cdot e(Q_B, Q'_A)^s$ вычисляется сторонами по формулам

$$k_A = e(S_A, Q'_B) \cdot e(Q_B, S'_A), \qquad k_B = e(S_B, Q'_A) \cdot e(Q_A, S'_B).$$

IBKE-протокол Смарта

В 2002 г. Н. Смарт (N. P. Smart) [61] предложил первый интерактивный протокол выработки общего ключа на основе операции билинейного спаривания, аналогичный протоколу MTI/A0 [44]. Это протокол типа SOK, в котором центр обладает ключевой парой (s, Q_{pub}), $Q_{\text{pub}} = [s]P$, $s \in_R \mathbb{Z}_q^*$, $\langle P \rangle = \mathbb{G}$, $|\mathbb{G}| = q$, а участник A получает ключевую пару $(Q_A = h_1(\text{ID}_A), S_A = [s]Q_A)$.

Для выработки общего ключа стороны выбирают случайно $r_A, r_B \in_R \mathbb{Z}_q^*$ и выполняют следующий протокол:

$$\begin{array}{ll} A \to B: & T_A = [r_A]P, \\ A \leftarrow B: & T_B = [r_B]P. \end{array} \tag{3}$$

Теперь A и B могут вычислить общий ключ соответственно по формулам

$$k_A = e([r_A]Q_B, Q_{\text{pub}})\, e(S_A, T_B), \quad k_B = e([r_B]Q_A, Q_{\text{pub}})\, e(S_B, T_A).$$

В результате получается ключ

$$\begin{gathered} k = e([r_A]Q_B, Q_{\text{pub}})\, e([s]Q_A, [r_B]P) = e([r_A]Q_B, Q_{\text{pub}})\, e([r_B]Q_A, Q_{\text{pub}}) = \\ = e([r_A]Q_B + [r_B]Q_A, Q_{\text{pub}}). \end{gathered}$$

Данный протокол, подобно протоколу MTI, защищён от атаки «противник-в-середине», но он не стоек к атаке чтения назад, так как при компрометации долговременных ключей S_A и S_B пользователей противник имеет возможность вычислять действующие ключи для любого сеанса

$$k = e(S_A, T_B)e(S_B, T_A).$$

IBKE-протокол Скотта

В 2002 г. М. Скотт (M. Scott) в [59] предложил другой способ обмена сообщениями, зависящими от идентификаторов сторон, аналогичный протоколу MTI/C1:

$$\begin{array}{ll} A \to B: & p_A = e(S_A, Q_B)^{r_A}, \\ A \leftarrow B: & p_B = e(S_B, Q_A)^{r_B}. \end{array}$$

Теперь A и B могут вычислить общий ключ $k = e(Q_A, Q_B)^{sr_Ar_B}$ соответственно по формулам

$$k_A = p_B^{r_A}, \quad k_B = p_A^{r_B}.$$

IBKE-протокол Шима

В 2003 г. К. Шим (K. Shim) [62] предложил защищённый от чтения назад вариант протокола. Он также отличается только способом вычисления общего ключа $k = e(T_A + Q_A, T_B + Q_B)^s$ по формулам

$$k_A = e([r_A]Q_{\text{pub}} + S_A, T_B + Q_B), \quad k_B = e([r_B]Q_{\text{pub}} + S_B, T_A + Q_A).$$

Однако в [63] найдена атака «противник-в-середине» на этот протокол. Противник C выбирает случайные числа a' и b' и подменяет сообщения в протоколе (3):

$$\begin{array}{lll} A \to & C(B): & T_A = [r_A]P, \\ & C(A) \to B: & T'_A = [a']P - Q_A, \\ & C(A) \leftarrow B: & T_B = [r_B]P, \\ A \leftarrow & C(B): & T'_B = [b']P - Q_B. \end{array}$$

Теперь A и B вычисляют различные ключи по формулам

$$k_A = e([r_A]Q_{pub} + S_A, T'_B + Q_B) = e([r_A]Q_{\text{pub}} + S_A, [a']P) = e(P,P)^{r_A s b'} e(Q_A, P)^{s b'},$$
$$k_B = e([r_B]Q_{\text{pub}} + S_B, T'_A + Q_A) = e([r_B]Q_{\text{pub}} + S_B, [b']P) = e(P,P)^{a' s r_B} e(Q_A, P)^{s a'}.$$

При этом противник C может вычислить эти значения по формулам

$$k'_A = e(T_A, b'Q_{\text{pub}}) e(Q_A, b'Q_{\text{pub}}) = e(P,P)^{r_A s b'} e(Q_A, P)^{s b'} = k_A,$$
$$k'_B = e(T_B, a'Q_{\text{pub}}) e(Q_B, a'Q_{\text{pub}}) = e(P,P)^{a' s r_B} e(Q_A, P)^{s a'} = k_B.$$

IBKE-протокол Рюи — Юн — Юу

Защищённый от этих атак вариант протокола Смарта типа SOK предложили в 2004 г. Е. Рюи, Е. Юн и К. Юу (E. K. Ryu, E. J. Yoon и K. Y. Yoo) [51]. Он отличается только способом вычисления общего ключа $k = ([r_A r_B]P, e(Q_A, Q_B)^s)$ по формулам

$$k_A = ([r_B]T_A, e(S_A, Q_B)), \quad k_B = ([r_A]T_B, e(S_B, Q_A)).$$

В 2009 г. С. Ванг и др. [64] предложили новый способ вычисления общего ключа, основанный на применении хеш-функции

$$k = h(\text{ID}_A, \text{ID}_B, [r_A r_B]P, e(Q_A, Q_B)^s, T_A, T_B).$$

IBKE-протокол Ванга

В 2013 г. Ю. Ванг (Y. Wang) [65] предложил новый аутентифицированный протокол вида

$$\begin{array}{ll} A \to B: & T_A = [r_A]Q_A, \\ A \leftarrow B: & T_B = [r_B]Q_B \end{array}$$

типа SOK, но отличающийся способом выработки общего ключа

$$k = e(Q_A, Q_B)^{s(t_A + s_A)(t_B + s_B)},$$

где $s_A = h(T_A, T_B)$; $s_B = h(T_B, T_A)$; $h : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$. Пользователи вычисляют общий ключ по формулам

$$K_A = e((t_A + s_A)S_A, s_B Q_B + T_B)), \quad K_B = e((t_B + s_B)S_B, s_A Q_A + T_A)).$$

IBKE-протокол МакКалаха — Барето

В 2005 г. Н. МакКалах и П. Барето (N. McCullagh и P. S. L. M. Barreto) [45] предложили протокол выработки общего ключа на основе идентификаторов, имеющий аналогии с протоколом цифровой подписи BLMQ. Пусть P — точка эллиптической кривой над полем $\mathbb{Z}_p$, $\langle P\rangle = \mathbb{G}$, $h : \{0,1\}^* \to \mathbb{Z}_p^*$ — хеш-функция. Центр обладает закрытым ключом s и открытым ключом $[s]P$. Пользователь A получает в удостоверяющем центре открытый ключ $Q_A = [h(\mathrm{ID}_A)]P + [s]P = [h(\mathrm{ID}_A) + s]P$ и личный ключ $S_A = [(h(\mathrm{ID}_A) + s)^{-1}]P$. Пользователи A и B выбирают случайные элементы поля r_A и r_B соответственно и выполняют протокол

$$\begin{aligned} A \to B: &\quad N_A = [r_A]Q_B, \\ A \leftarrow B: &\quad N_B = [r_B]Q_A. \end{aligned}$$

Теперь A и B могут вычислить общий ключ $k = e(P,P)^{s(r_A+r_B)}$ по формулам

$$k_A = e(S_A, N_B)^{r_A}, \qquad k_B = e(S_B, N_A)^{r_B}.$$

IBKE-протоколы выработки общего ключа на основе эллиптической кривой без операции билинейного спаривания

IBKE-протокол Као — Коу — Ду

Протокол предложен X. Cao, W. Kou, X. Du. в 2010 г. [11]. Центр KGC генерирует для пользователя с идентификатором $\mathtt{ID}_A$ случайное число s_A, выступающее в роли случайного параметра для текущей ключевой пары, и вычисляет открытый ключ $Q_A = s_A P$ и личный ключ $\sigma_A = s_A + h_A s \bmod q$, где $h_A = h_1(\mathtt{ID}_A \| s_A)$, $h_1 : \{0,1\}^* \times \mathbb{G}_T \to \mathbb{Z}_q$.

Для выработки общего ключа пользователи A и B выбирают случайные числа r_A и r_B и выполняют протокол

$$\begin{aligned} A \to B: &\quad T_A = [r_A]P, Q_A, \\ A \leftarrow B: &\quad T_B = [r_B]P, Q_B. \end{aligned}$$

Теперь A и B могут вычислить общий ключ $K = [r_A r_B]P$ по формулам

$$K_A = Q_A + [r_A](T_B + [h_B]Q_{\mathrm{pub}}), \qquad K_B = Q_B + [r_B](T_A + [h_A]Q_{\mathrm{pub}}).$$

IBKE-протокол Ислама — Бисваса

Протокол предложили H. Islam и G. P. Biswas в 2010 г. [36]. Он отличается от предыдущего видом пересылаемых сообщений и формулой для общего ключа:

$$\begin{aligned} A \to B: &\quad T_A = [r_A](Q_A + [h_A]Q_{\mathrm{pub}}), Q_A, \\ A \leftarrow B: &\quad T_B = [r_B](Q_B + [h_B]Q_{\mathrm{pub}}), Q_B. \end{aligned}$$

Теперь A и B могут вычислить общий ключ $K = [(r_A + r_B)\sigma_A\sigma_B]P$ по формулам

$$K_A = [\sigma_A](T_B + [r_A](Q_B + [h_B]Q_{\mathrm{pub}})), \qquad K_B = [\sigma_B](T_A + [r_B](Q_A + [h_B]Q_{\mathrm{pub}})).$$

IBKE-протокол Горейши и др.

Протокол предложен в 2015 г. [24]. Пусть $h_1 : \{0,1\}^* \times \mathbb{G}_T \to \mathbb{Z}_q$, $h_2 : \{0,1\}^* \to \mathbb{Z}_q^*$. Помимо ключевой пары (Q_A, σ_A), пользователь A выбирает случайное число $x_A \in_R \mathbb{Z}_q^*$ и вычисляет $X_A = x_A P$, $y_A = \sigma_A + h_2(\mathtt{ID}_A)x_A \bmod q$, $Y_A = [y_A]P$. Протокол предполагает выполнение предварительного этапа обмена значениями (Q_A, P_A) и (Q_B, P_B):

$$\begin{aligned} A \to B: &\quad Q_A, X_A, \\ A \leftarrow B: &\quad Q_B, X_B, \\ A \to B: &\quad T_A = [r_A \sigma_A y_A] Y_B, \\ A \leftarrow B: &\quad T_B = [r_B \sigma_B y_B] Y_A. \end{aligned}$$

Пользователи A и B вычисляют по формулам $K_A = [r_A \sigma_A] T_B$ и $K_B = [r_B \sigma_B] T_A$ общий ключ

$$K = [r_A r_B \sigma_A \sigma_B y_A y_B] P.$$

Данный протокол более быстрый, так как в нём требуется только два раза вычислять кратные точки вместо трёх.

Выводы

В работе проанализированы основные положительные и отрицательные свойства криптографических систем с открытыми ключами, вычисляемыми на основе идентификационной информации. Несмотря на очевидные достоинства, связанные с упрощенной процедурой распределения ключей, такие системы обладают целым рядом ограничений, вытекающих из способа их построения. К их числу относятся: трудность масштабирования на распределённые системы с большим числом пользователей, необходимость наличия защищённого канала для получения ключей пользователями, высокая степень доверия к центру генерации ключей, имеющему возможность в любой момент восстанавливать все ранее им выданные личные ключи пользователей, сложность процедур отзыва и обновления ключей и др. Описаны способы защиты от возможных уязвимостей и перечислены применяемые при этом математические конструкции.

ЛИТЕРАТУРА

1. *Abdalla M., Bellare M., Catalano D., et al.* Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions // LNCS. 2005. V. 3621. P. 205–222.
2. *Al-Riyami S. S. and Paterson K. G.* Certificateless public key cryptography // LNCS. 2003. V. 2894. P. 452–473.
3. *Au M., Liu J., Susilo W., and Yuen T.* Certificate based (linkable) ring signature // LNCS. 2007. V. 4464. P. 79–92.
4. *Barreto P. S. L. M., Libert B., McCullagh N., and Quisquater J-J.* Efficient and Secure Identity-Based Signatures and Signcryption from Bilinear Maps. `https://www.slideserve.com/connie/efficient-and-secure-identity-based-signatures-and-signcryption-from-bilinear-maps`.
5. *Barreto P. S. L. M., Libert B., McCullagh N., and Quisquater J-J.* Efficient and provably-secure identity-based signatures and signcryption from bilinear maps // LNCS. 2005. V. 3788. P. 515–532.
6. *Boldyreva A., Goyal V., and Kumar V.* Identity-based encryption with efficient revocation // Proc. CCS'08. N.Y.: ACM, 2008. P. 417–426.

7. *Baek J., Newmarch J., Safavi-Naini R., and Susilo W.* A survey of identity-based cryptography // Proc. Australian Unix Users Group Annual Conf. 2004. P. 95–102.
8. *Boneh D and Boyen X.* Efficient selective-ID secure Identity-Based Encryption without random oracles // LNCS. 2004. V. 3027. P. 223–238.
9. *Boneh D. and Franklin M.* Identity based encryption from the Weil pairing // LNCS. 2001. V. 2139. P. 213–229; SIAM J. Comput. 2003. V. 32. No. 3. P. 586–615.
10. *Boyd C., Mathura A., and Stebila D.* Protocols for Authentication and Key Establishment. 2nd ed. Berlin; Heidelberg: Springer, 2020. 521 p.
11. *Cao X., Kou W., and Du X.* A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges // Inform. Sci. 2010. V. 180. Iss. 15. P. 2895–2903.
12. *Cha J. C. and Cheon J. H.* An identity-based signature from gap Diffie — Hellman groups // LNCS. 2003. V. 2567. P. 18–30.
13. *Chatterjee S. and Sarkar P.* Trading time for space: Towards an efficient IBE scheme with short(er) public parameters in the standard model // LNCS. 2005. V. 3935. P. 424–440.
14. *Chen L. and Cheng Z.* Security proof of Sakai — Kasahar's identity-based encryption scheme // LNCS. 2005. V. 3796. P. 442–459.
15. *Chen L.* Identity-based Cryptography. Hewlett-Packard Laboratories. September 2006. `http://www.sti.uniurb.it/events/fosad06/papers/Chen-fosad06.pdf`.
16. *Chatterjee S. and Sarkar P.* Identity-Based Encryption. N.Y.: Springer, 2011. 180 p.
17. *Chen L., Cheng Z., and Smart N. P.* Identity-based Key Agreement Protocols from Pairings. Cryptology ePrint Archive. Report 2006/199. 2006. `https://eprint.iacr.org/2006/199.pdf`.
18. *Chen X., Zhang F., and Kim K.* A new ID-based group signature scheme from bilinear pairings // LNCS. 2003. V. 2908. P. 585–592.
19. *Cocks C.* An identity-based encryption scheme based on quadratic residues // LNCS. 2001. V. 2260. P. 360–363.
20. *Gagné M.* Identity-Based Encryption: a Survey // RSA Laboratories Cryptobytes. 2003. V. 6. No. 1. P. 10–19.
21. *Galindo D.* Boneh-Franklin identity based encryption revisited // Proc. ICALP 2005. Lisbon, Portugal, 2005. P. 791–802
22. *Gentry C.* Certificate-based encryption and the certificate revocation problem // LNCS. 2003. V. 2656. P. 272–293.
23. *Gentry C. and Silverberg A.* Hierarchical ID-based cryptography // LNCS. 2002. V. 2501. P. 548–566.
24. *Ghoreishi S.-M., Isnin I. F., Razak S. A., and Chizari H.* Secure and authenticated key agreement protocol with minimal complexity of operations in the context of identity-based cryptosystems // Proc. I4CT. Kuching, Malaysia, 2015. P. 299–303.
25. *Girault M. and Paillès J. C.* An identity-based scheme providing zero-knowledge authentication and authenticated key exchange // Proc. ESORICS. AFCET, Toulouse, 1990. P. 173–184.
26. *Girault M.* Self-certified public keys // LNCS. 1991. V. 547. P. 490–497.
27. *Gorantla M. C., Gangishetti R., and Saxena A.* A Survey on ID-Based Cryptographic Primitives. `http://eprint.iacr.org/2005/094`.
28. *Green M. and Hohenberger S.* Blind identity-based encryption and simulatable oblivious transfer // LNCS. 2007. V. 4833. P. 265–282.
29. *Guillou L. and Quisquater J.-J.* A practical zero knowledge protocol fitted to security microprocessor minimizing both transmission and memory // LNCS. 1988. V. 330. P. 123–128.

30. *Guillou L. C. and Quisquater J.-J.* A "paradoxical" identity-based signature scheme resulting from zero-knowledge // LNCS. 1990. V. 403. P. 216–231.
31. *Guillou L. C., Ugon M., and Quisquater J.-J.* Cryptographic authentication protocols for smart cards // Computer Networks Magazine. 2002. V. 36. P. 437–451.
32. *Günther C. G.* An identity-based key-exchange protocol // LNCS. 1990. V. 434. P. 29–37.
33. *Grumăzescu C. and Patriciu V-V.* A comprehensive survey on ID-based cryptography for wireless sensor networks // J. Military Technology. 2018. V. 1. No. 1. P. 57–70.
34. *Horwitz J. and Lynn B.* Toward hierarchical identity-based encryption // LNCS. 2002. V. 2332. P. 466–481.
35. *Hess F.* Efficient identity based signature schemes based on pairings // LNCS. 2003. V. 2595. P. 310–324.
36. *Islam H. and Biswas G. P.* An improved pairing-free identity-based authenticated key agreement protocol based on ECC // Procedia Engineering. 2012. V. 30. P. 499–507.
37. *Kang B. G., Park J. H., and Hahn S. G.* A certificate-based signature scheme // LNCS. 2004. V. 2964. P. 99–111.
38. *Katz J.* Binary tree encryption: Constructions and applications // LNCS. 2004. V. 2971. P. 1–11.
39. *Lee K., Lee D. H., and Park J. H.* Efficient revocable identity-based encryption via subset difference methods // Des. Codes Cryptogr. 2017. V. 85. P. 39–76.
40. *Tseng Y. and Tsai T.* Efficient revocable ID-based encryption with a public channel // Computer J. 2012. V. 55. No. 4. P. 475–486.
41. *Li J., Huang X., Mu Y., et al.* Certificate-based signature: Security model and efficient construction // LNCS. 2007. V. 4582. P. 110–125.
42. *Libert B. and Vergnaud D.* Adaptive-ID Secure revocable identity-based encryption // LNCS. 2009. V. 5473. P. 1–15.
43. *Liu J. K., Baek J., Susilo W., and Zhou J.* Certificate-based signature schemes without pairings or random oracles // LNCS. 2008. V. 5222. P. 285–297.
44. *Matsumoto T., Takashima Y., and Imai H.* On seeking smart public-key distribution systems // Trans. IECE. Japan. Sec. E. 1986. V. 69. Iss. 2. P. 99–106.
45. *McCullagh N. and Barreto P. S. L. M.* A new two-party identity-based authenticated key agreement // LNCS. 2005. V. 3376. P. 262–274.
46. *Naccache D.* Secure and Practical Identity-Based Encryption. Cryptology ePrint Archive. Report 2005/369. 2005. `https://eprint.iacr.org/2005/369`.
47. *Nalla D. and Reddy K. C.* Signcryption Scheme for Identity-based Cryptosystems. `https://eprint.iacr.org/2003/066.pdf`.
48. *Okamoto E.* Key distribution systems based on identification information // LNCS. 1987. V. 293. P. 194–202.
49. *Okamoto T.* Efficient blind and partially blind signatures without random oracles // LNCS. 2006. V. 3876. P. 80–99.
50. *Okamoto E. and Tanaka K.* Key distribution system based on identification information // IEEE J. Selected Areas Communications. 1989. V. 7. No. 4. P. 481–485.
51. *Ryu E. K., Yoon E. J., and Yoo K. Y.* An efficient ID-based authenticated key agreement protocol from pairings // LNCS. 2004. V. 3042. P. 1464–1469.
52. *Saeednia S.* Improvement of Gunther's identity-based key exchange protocol // Electronics Lett. 2000. V. 36. No. 18. P. 1535–1536.
53. *Sakai R., Ohgishi K., and Kasahara M.* Cryptosystems based on pairing // Proc. SCIS'00. Okinawa, Japan, 2000. P. 26–28.

54. *Sakai R., Ohgishi K., and Kasahara M.* Cryptosystems based on pairing over elliptic curve // Proc. Symp. on Cryptography and Information Security. Oiso, Japan, January 2001. (in Japanese)
55. *Sakai R. and Kasahara M.* ID Based Cryptosystems with Pairing on Elliptic Curve. Cryptology ePrint Archive. Report 2003/054. `https://eprint.iacr.org/2003/054.pdf`. 2003.
56. *Sahai A. and Waters B.* Fuzzy identity-based encryption // LNCS. 2005. V. 3494. P. 457–473.
57. *Sayid J., Sayid I., and Kar J.* Certificateless public key cryptography: A research survey // Intern. J. Security Appl. 2016. V. 10. No. 7. P. 103–118.
58. *Seo J. H. and Emura K.* Revocable hierarchical identity-based encryption // Theor. Comput. Sci. 2014. V. 542. P. 44–62.
59. *Scott M.* Authenticated ID-Based Key Exchange and Remote Log-in with Simple Token and PIN Number. Cryptology ePrint Archive. 2002. Report 2002/164. `https://eprint.iacr.org/2002/164`.
60. *Shamir A.* Identity-based cryptosystems and signature schemes // LNCS. 1984. V. 196. P. 47–53.
61. *Smart N. P.* An identity based authenticated key agreement protocol based on the Weil pairing // Electronics Lett. 2002. V. 38. No. 13. P. 630–632.
62. *Shim K.* Efficient ID-based authenticated key agreement protocol based on Weil pairing // Electronics Lett. 2003. V. 39. No. 8. P. 653–654.
63. *Sun H.-M. and Hsieh B.-T.* Security Analysis of Shim's Authenticated Key Agreement Protocols from Pairings. Cryptology ePrint Archive. 2003. Report 2003/113. `http://eprint.iacr.org/2003/113`.
64. *Wang S., Cao Z., Choo K. K. R., and Wang L.* An improved identity-based key agreement protocol and its security proof // Inf. Sci. 2009. V. 179. No. 3. P. 307–318.
65. *Wang Y.* Efficient identity-based and authenticated key agreement protocols // LNCS. 2013. V. 7420. P. 172–197.
66. *Waters B.* Efficient identity-based encryption without random oracles // Proc. EUROCRYPT'05. Aarhus, Denmark, 2005. P. 114–127.
67. *Yao D., Fazio N., Dodis Y., and Lysyanskaya A.* Id-based encryption for complex hierarchies with applications to forward security and broadcast encryption // Proc. CCS'04. Washington: ACM, 2004. P. 354–363.
68. *Zheng Y.* Digital signcryption or how to achieve cost(signature&encryption) $\ll$ cost(signature) + cost(encryption) // LNCS. 1997. V. 1294. P. 165–179.
69. ISO/IEC 14888-2. Information Technology — Security Techniques — Digital Signatures with Appendix — P. 2: Integer Factorization Based Mechanisms. ISO/IEC, 1999.
70. ISO/IEC 14888-3. Information Technology — Security Techniques — Digital Signatures with Appendix — P. 3: Discrete Logarithm Based Mechanisms. ISO/IEC, 1998.
71. ISO/IEC 11770-3. Information Technology — Security Techniques — Key Management — P. 3: Mechanisms Using Asymmetric Techniques. ISO/IEC, 1999.
72. IEEE P1363.3. Identity-Based Public Key Cryptography Using Pairings. `https://standards.ieee.org/ieee/1363.3/3822/`. 2013.
73. GM/T 0044.2-2016. Identity-Based Cryptographic Algorithm using Bilinear Pairings — P. 2: Digital Signature Algorithm. 2016. (in Chinese).

REFERENCES

1. *Abdalla M., Bellare M., Catalano D., et al.* Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. LNCS, 2005, vol. 3621, pp. 205–222.

2. *Al-Riyami S. S. and Paterson K. G.* Certificateless public key cryptography. LNCS, 2003, vol. 2894, pp. 452–473.
3. *Au M., Liu J., Susilo W., and Yuen T.* Certificate based (linkable) ring signature. LNCS, 2007, vol. 4464, pp. 79–92.
4. *Barreto P. S. L. M., Libert B., McCullagh N., and Quisquater J-J.* Efficient and Secure Identity-Based Signatures and Signcryption from Bilinear Maps. `https://www.slideserve.com/connie/efficient-and-secure-identity-based-signatures-and-signcryption-from-bilinear-maps`.
5. *Barreto P. S. L. M., Libert B., McCullagh N., and Quisquater J-J.* Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. LNCS, 2005, vol. 3788, pp. 515–532.
6. *Boldyreva A., Goyal V., and Kumar V.* Identity-based encryption with efficient revocation. Proc. CCS'08, N.Y., ACM, 2008, pp. 417–426.
7. *Baek J., Newmarch J., Safavi-Naini R., and Susilo W.* A survey of identity-based cryptography. Proc. Australian Unix Users Group Annual Conf., 2004, pp. 95–102.
8. *Boneh D and Boyen X.* Efficient selective-ID secure Identity-Based Encryption without random oracles. LNCS, 2004, vol. 3027, pp. 223–238.
9. *Boneh D. and Franklin M.* Identity based encryption from the Weil pairing. LNCS, 2001, vol. 2139, pp. 213–229; SIAM J. Comput., 2003, vol. 32, no. 3, pp. 586–615.
10. *Boyd C., Mathura A., and Stebila D.* Protocols for Authentication and Key Establishment. 2nd ed. Berlin; Heidelberg, Springer, 2020. 521 p.
11. *Cao X., Kou W., and Du X.* A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges. Inform. Sci., 2010, vol. 180, iss. 15, pp. 2895–2903.
12. *Cha J. C. and Cheon J. H.* An identity-based signature from gap Diffie — Hellman groups. LNCS, 2003, vol. 2567, pp. 18–30.
13. *Chatterjee S. and Sarkar P.* Trading time for space: Towards an efficient IBE scheme with short(er) public parameters in the standard model. LNCS, 2005, vol. 3935, pp. 424–440.
14. *Chen L. and Cheng Z.* Security proof of Sakai — Kasahar's identity-based encryption scheme. LNCS, 2005, vol. 3796, pp. 442–459.
15. *Chen L.* Identity-based Cryptography. Hewlett-Packard Laboratories. September 2006. `http://www.sti.uniurb.it/events/fosad06/papers/Chen-fosad06.pdf`.
16. *Chatterjee S. and Sarkar P.* Identity-Based Encryption. N.Y., Springer, 2011. 180 p.
17. *Chen L., Cheng Z., and Smart N. P.* Identity-based Key Agreement Protocols from Pairings. Cryptology ePrint Archive, Report 2006/199, 2006. `https://eprint.iacr.org/2006/199.pdf`.
18. *Chen X., Zhang F., and Kim K.* A new ID-based group signature scheme from bilinear pairings. LNCS, 2003, vol. 2908, pp. 585–592.
19. *Cocks C.* An identity-based encryption scheme based on quadratic residues. LNCS, 2001, vol. 2260, pp. 360–363.
20. *Gagné M.* Identity-Based Encryption: a Survey. RSA Laboratories Cryptobytes, 2003, vol. 6, no. 1, pp. 10–19.
21. *Galindo D.* Boneh-Franklin identity based encryption revisited. Proc. ICALP 2005, Lisbon, Portugal, 2005, pp. 791–802
22. *Gentry C.* Certificate-based encryption and the certificate revocation problem. LNCS, 2003, vol. 2656, pp. 272–293.
23. *Gentry C. and Silverberg A.* Hierarchical ID-based cryptography. LNCS, 2002, vol. 2501, pp. 548–566.

24. *Ghoreishi S.-M., Isnin I. F., Razak S. A., and Chizari H.* Secure and authenticated key agreement protocol with minimal complexity of operations in the context of identity-based cryptosystems. Proc. I4CT. Kuching, Malaysia, 2015. pp. 299–303.
25. *Girault M. and Paillès J. C.* An identity-based scheme providing zero-knowledge authentication and authenticated key exchange. Proc. ESORICS, AFCET, Toulouse, 1990, pp. 173–184.
26. *Girault M.* Self-certified public keys. LNCS, 1991, vol. 547, pp. 490–497.
27. *Gorantla M. C., Gangishetti R., and Saxena A.* A Survey on ID-Based Cryptographic Primitives. `http://eprint.iacr.org/2005/094`.
28. *Green M. and Hohenberger S.* Blind identity-based encryption and simulatable oblivious transfer. LNCS, 2007, vol. 4833, pp. 265–282.
29. *Guillou L. and Quisquater J.-J.* A practical zero knowledge protocol fitted to security microprocessor minimizing both transmission and memory. LNCS, 1988, vol. 330, pp. 123–128.
30. *Guillou L. C. and Quisquater J.-J.* A "paradoxical" identity-based signature scheme resulting from zero-knowledge. LNCS, 1990, vol. 403, pp. 216–231.
31. *Guillou L. C., Ugon M., and Quisquater J.-J.* Cryptographic authentication protocols for smart cards. Computer Networks Magazine, 2002, vol. 36, pp. 437–451.
32. *Günther C. G.* An identity-based key-exchange protocol. LNCS, 1990, vol. 434, pp. 29–37.
33. *Grumăzescu C. and Patriciu V-V.* A comprehensive survey on ID-based cryptography for wireless sensor networks. J. Military Technology, 2018, vol. 1, no. 1, pp. 57–70.
34. *Horwitz J. and Lynn B.* Toward hierarchical identity-based encryption. LNCS, 2002, vol. 2332, pp. 466–481.
35. *Hess F.* Efficient identity based signature schemes based on pairings. LNCS, 2003, vol. 2595, pp. 310–324.
36. *Islam H. and Biswas G. P.* An improved pairing-free identity-based authenticated key agreement protocol based on ECC. Procedia Engineering, 2012, vol. 30, pp. 499–507.
37. *Kang B. G., Park J. H., and Hahn S. G.* A certificate-based signature scheme. LNCS, 2004. vol. 2964. pp. 99–111.
38. *Katz J.* Binary tree encryption: Constructions and applications. LNCS, 2004, vol. 2971, pp. 1–11.
39. *Lee K., Lee D. H., and Park J. H.* Efficient revocable identity-based encryption via subset difference methods. Des. Codes Cryptogr., 2017, vol. 85, pp. 39–76.
40. *Tseng Y. and Tsai T.* Efficient revocable ID-based encryption with a public channel. Computer J., 2012, vol. 55, no. 4, pp. 475–486.
41. *Li J., Huang X., Mu Y., et al.* Certificate-based signature: Security model and efficient construction. LNCS, 2007, vol. 4582, pp. 110–125.
42. *Libert B. and Vergnaud D.* Adaptive-ID Secure revocable identity-based encryption. LNCS, 2009, vol. 5473, pp. 1–15.
43. *Liu J. K., Baek J., Susilo W., and Zhou J.* Certificate-based signature schemes without pairings or random oracles. LNCS, 2008, vol. 5222, pp. 285–297.
44. *Matsumoto T., Takashima Y., and Imai H.* On seeking smart public-key distribution systems. Trans. IECE, Japan, Sec. E, 1986, vol. 69, iss. 2, pp. 99–106.
45. *McCullagh N. and Barreto P. S. L. M.* A new two-party identity-based authenticated key agreement. LNCS, 2005, vol. 3376, pp. 262–274.
46. *Naccache D.* Secure and Practical Identity-Based Encryption. Cryptology ePrint Archive. Report 2005/369. 2005. `https://eprint.iacr.org/2005/369`.

47. *Nalla D. and Reddy K. C.* Signcryption Scheme for Identity-based Cryptosystems. https://eprint.iacr.org/2003/066.pdf.
48. *Okamoto E.* Key distribution systems based on identification information. LNCS, 1987, vol. 293, pp. 194–202.
49. *Okamoto T.* Efficient blind and partially blind signatures without random oracles. LNCS, 2006, vol. 3876, pp. 80–99.
50. *Okamoto E. and Tanaka K.* Key distribution system based on identification information. IEEE J. Selected Areas Communications. 1989, vol. 7, no. 4, pp. 481–485.
51. *Ryu E. K., Yoon E. J., and Yoo K. Y.* An efficient ID-based authenticated key agreement protocol from pairings. LNCS, 2004, vol. 3042, pp. 1464–1469.
52. *Saeednia S.* Improvement of Gunther's identity-based key exchange protocol. Electronics Lett., 2000, vol. 36, no. 18, pp. 1535–1536.
53. *Sakai R., Ohgishi K., and Kasahara M.* Cryptosystems based on pairing. Proc. SCIS'00. Okinawa, Japan, 2000, pp. 26–28.
54. *Sakai R., Ohgishi K., and Kasahara M.* Cryptosystems based on pairing over elliptic curve. Proc. Symp. on Cryptography and Information Security, Oiso, Japan, January 2001. (in Japanese)
55. *Sakai R. and Kasahara M.* ID Based Cryptosystems with Pairing on Elliptic Curve. Cryptology ePrint Archive, Report 2003/054. https://eprint.iacr.org/2003/054.pdf. 2003.
56. *Sahai A. and Waters B.* Fuzzy identity-based encryption. LNCS, 2005, vol. 3494, pp. 457–473.
57. *Sayid J., Sayid I., and Kar J.* Certificateless public key cryptography: A research survey. Intern. J. Security Appl., 2016. vol. 10, no. 7, pp. 103–118.
58. *Seo J. H. and Emura K.* Revocable hierarchical identity-based encryption. Theor. Comput. Sci., 2014. vol. 542, pp. 44–62.
59. *Scott M.* Authenticated ID-Based Key Exchange and Remote Log-in with Simple Token and PIN Number. Cryptology ePrint Archive, 2002, Report 2002/164. https://eprint.iacr.org/2002/164.
60. *Shamir A.* Identity-based cryptosystems and signature schemes. LNCS, 1984, vol. 196, pp. 47–53.
61. *Smart N. P.* An identity based authenticated key agreement protocol based on the Weil pairing. Electronics Lett., 2002, vol. 38, no. 13, pp. 630–632.
62. *Shim K.* Efficient ID-based authenticated key agreement protocol based on Weil pairing. Electronics Lett., 2003, vol. 39, no. 8, pp. 653–654.
63. *Sun H.-M. and Hsieh B.-T.* Security Analysis of Shim's Authenticated Key Agreement Protocols from Pairings. Cryptology ePrint Archive, 2003, Report 2003/113. http://eprint.iacr.org/2003/113.
64. *Wang S., Cao Z., Choo K. K. R., and Wang L.* An improved identity-based key agreement protocol and its security proof. Inf. Sci., 2009, vol. 179, no. 3, pp. 307–318.
65. *Wang Y.* Efficient identity-based and authenticated key agreement protocols. LNCS, 2013, vol. 7420, pp. 172–197.
66. *Waters B.* Efficient identity-based encryption without random oracles. Proc. EUROCRYPT'05, Aarhus, Denmark, 2005, pp. 114–127.
67. *Yao D., Fazio N., Dodis Y., and Lysyanskaya A.* Id-based encryption for complex hierarchies with applications to forward security and broadcast encryption. Proc. CCS'04, Washington, ACM, 2004, pp. 354–363.
68. *Zheng Y.* Digital signcryption or how to achieve cost(signature&encryption) $\ll$ cost(signature) + cost(encryption). LNCS, 1997, vol. 1294, pp. 165–179.

69. ISO/IEC 14888-2. Information Technology — Security Techniques — Digital Signatures with Appendix — P. 2: Integer Factorization Based Mechanisms. ISO/IEC, 1999.
70. ISO/IEC 14888-3. Information Technology — Security Techniques — Digital Signatures with Appendix — P. 3: Discrete Logarithm Based Mechanisms. ISO/IEC, 1998.
71. ISO/IEC 11770-3. Information Technology — Security Techniques — Key Management — P. 3: Mechanisms Using Asymmetric Techniques. ISO/IEC, 1999.
72. IEEE P1363.3. Identity-Based Public Key Cryptography Using Pairings. `https://standards.ieee.org/ieee/1363.3/3822/`. 2013.
73. GM/T 0044.2-2016. Identity-Based Cryptographic Algorithm using Bilinear Pairings — P. 2: Digital Signature Algorithm. 2016. (in Chinese).