УДК 681.3.06 (0.43)
DOI: 10.15587/1729-4061.2020.210683

Разработка модифицированного алгоритма UMAC на крипто-кодовых конструкциях

А. А. Гаврилова, И. Д. Волков, Ю. В. Кожедуб, Р. В. Королев, А. В. Лезик, В. К. Медведев, А. В. Милов, Б. П. Томашевский, А. В. Тристан, О. Н. Чекунова

Розвиток обчислювальної техніки визначив вектор напрямку розширення послуг на основі Інтернет-технологій та технологій "G". Основними вимогами, визначеними до сучасних послуг в банківському секторі, є безпека і надійність. Однак в умовах постквантового періоду фахівцями НІСТ США ставиться під сумнів стійкість сучасних засобів забезпечення основних послуг безпеки на основі криптоалгоритмів симетричної і несиметричної криптографії. Зростання обчислювальних ресурсів дозволяє зловмисникам використовувати сучасні загрози в комплексі. Таким чином вони отримують властивості синергізму і гібридності, що значно збільшує можливість їх реалізації. Тому виникає необхідність пошуку нових і/або модифікація відомих алгоритмів формування MAC-кодів (*message authentication code*)*. Крім того, зростання послуг в кіберпросторі збільшує обсяги інформації, автентичність якої необхідно забезпечити. Серед відомих геш-алгоритмів виділяються геш-функції універсального гешування, що дозволяють спочатку визначити кількість колізій і їх рівномірний розподіл по всій безлічі геш-кодів. Розглянуто можливості модифікації каскадного алгоритму гешування UMAC на основі використання крипто-кодової конструкції Мак-Еліса на алгеброгеометричних* (*еліптичних* (*EC*)*, модифікованих еліптичних* (*MEC*)) *кодах та збиткових кодах* (*DC*)*. Такий підхід дозволяє зберегти властивість унікальності на відміну від класичної схеми UMAC* (*message authentication code based on universal hashing, universal MAC*) *на основі блочного симетричного шифру* (*AES*)*. Представлені алгоритми оцінки властивостей універсальності і суворої універсальності геш-кодів, які дозволяють оцінити стійкість запропонованих конструкцій гешування на основі універсальних геш-функцій з урахуванням збереження властивості універсальності.*

Ключові слова: автентичність, алгоритм гешування, крипто-кодові конструкції, еліптичні коди, модифіковані еліптичні коди, збиткові коди, алгоритм UMAC, алгоритм MV2 (*універсальний механізм нанесення збитку*)*, постквантова криптографія.*

1. Введение

Развитие банковского сектора в последнее десятилетие позволило значительно расширить спектр своих услуг на основе использования вычислительных ресурсов Интернет-технологий и технологий X“G“–LTE (Long-Term Evolution).

Данные изменения способствуют развитию цифровой экономики, и в частности, электронного банкинга [1, 2]. Однако этому сопутствует рост количества

и разнообразия киберугроз. В [3–5] представлены результаты анализа киберугроз за последние три года на автоматизированные банковские системы (АБС) организаций банковского сектора (ОБС) (рис. 1).

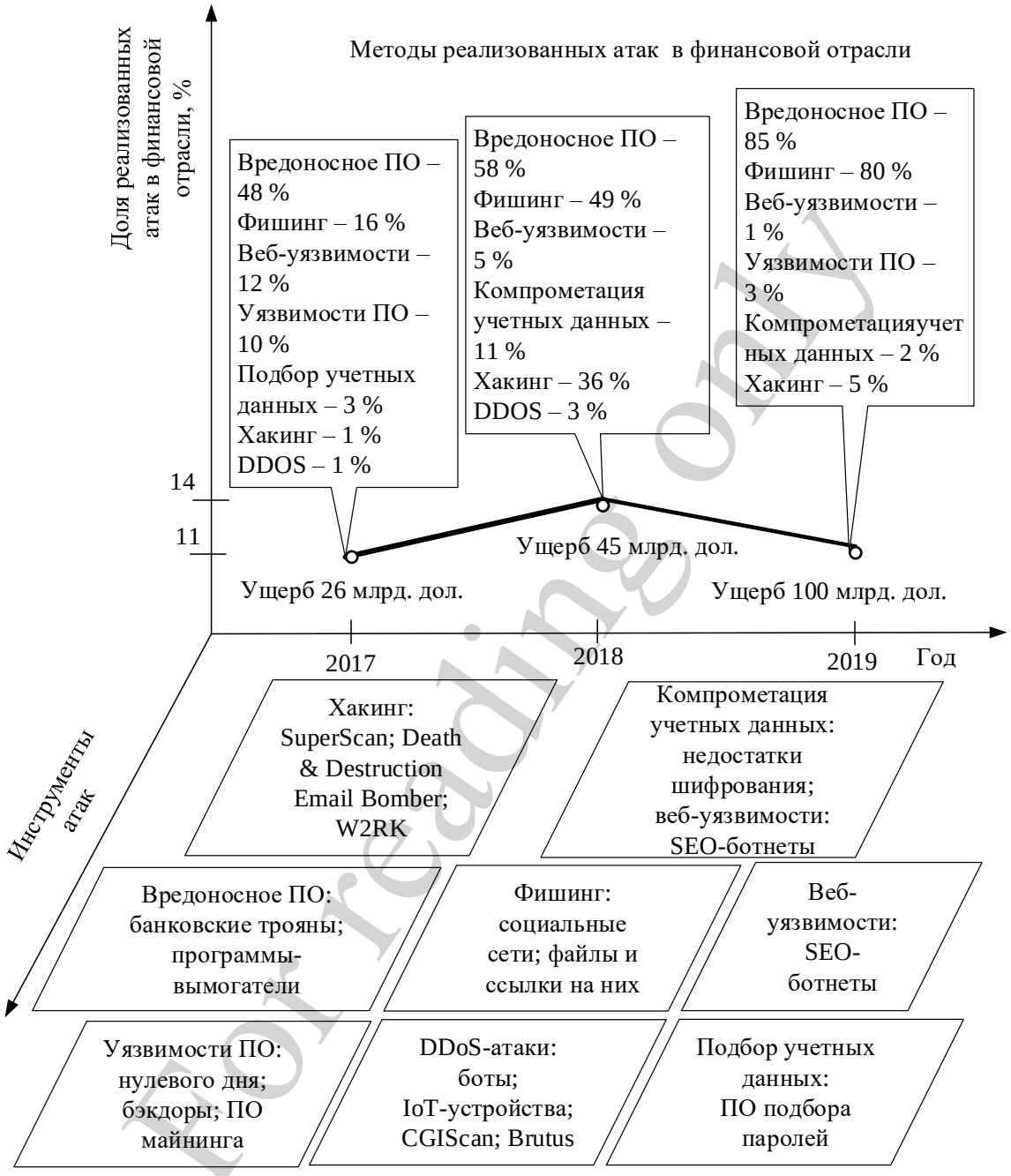


Рис. 1. Тенденции направленности кибератак на финансовый сектор: DDOS – Distributed Denial of Service – распределенный отказ в обслуживании; SuperScan - мощный сканер TCP портов; Death & Destruction – смертельный & всесокрушающий мейл-бомбер, программа организации спама на почтовый сервер; W2RK – Windows 2000 Resource Kit – комплект инструментов администратора Windows; SEO – Search Engine Optimization – комплекс мероприятий по увеличению видимости сайта в поисковых системах по целевым поисковым запросам; IoT – Internet of Things – Интернет вещей

Приведенный график показывает, что за 2017–2019 года на размеры ущерба от реализованных атак в финансовом секторе значительное влияние оказали такие методы реализованных атак как фишинг (2017 год – 16 %, 2018 год – 49 %, 2019 год – 80 %) и вредоносное программное обеспечение (ПО) (2017 год – 48 %, 2018 год – 58 %, 2019 год – 85 %). С точки зрения основных векторов современных атак на АБС проведенный анализ свидетельствует об их комплексировании с методами социальной инженерии. Это приводит к появлению у уже известных угроз свойств гибридности и синергизма [3–5].

Представленная статистка свидетельствует о том, что рост угроз, связанных с услугой аутентичности, неуклонно растет. Поэтому актуальными следует признать научные исследования, ориентированные на формирование новых подходов, обеспечивающих услугу аутентичности в условиях стремительного роста вычислительных ресурсов и современных угроз на основе полномасштабного квантового компьютера, на основе разработки модифицированного алгоритма UMAC с крипто-кодовыми конструкциями Мак-Элиса (в качестве формирования подкладки).

2. Анализ литературных данных и формулировка проблемы

В условиях роста вычислительных ресурсов и современных технологий увеличения объемов данных возникает интегрированная задача обеспечения не только безопасности, оперативности, но и аутентичности. Для выполнения предлагается использовать алгоритм каскадного хеширования UMAC (message authentication code based on universal hashing, universal MAC, код аутентификации сообщения на основе универсального хеширования), который одновременно позволяет обеспечить требуемый уровень стойкости и оперативности на основе использования универсальных функций хеширования. Однако классическая схема для обеспечения стойкости хеш-кода использует алгоритм блочно-симметричного шифра AES (Advanced Encryption Standard), что в итоге не позволяет обеспечить универсальность.

В работах [3–5] представлены результаты исследований векторов направленности киберугроз. Проведенный анализ свидетельствует, что использование угроз, направленных на взлом механизмов аутентификации, позволяет выполнить удаленный доступ к конфиденциальной информации и/или получить “привилегии”, обеспечивающие возможность “взлома” комплексной системы защиты информации (КСЗИ). В работах [6–8] рассмотрены возможности использования крипто-кодовых конструкций Мак-Элиса и Нидеррайтера, обеспечивающие услуги безопасности – конфиденциальность, целостность. Кроме этого интегрировано обеспечиваются оперативность криптопреобразований и достоверность передачи данных на основе использования алгеброгеометрических (циклических) кодов. Существенными недостатками их практического использования является возможность нахождения элементов матриц маскировки (личного ключа каждого пользователя системы) на основе атаки Сидельникова [9] и значительных энергетических затрат на практическую реализацию над конечным (вычислительным) полем размерностью GF (2^{10}–2^{13}) (Galois field – поле Галуа). Как известно, конечным полем называется конечное множество, на ко-

тором определены произвольные операции, называемые сложением, умножением, вычитанием и делением.

В работах [10–15] рассмотрены возможности полномасштабных квантовых компьютеров, обеспечивающих задачу взлома алгоритмов симметричной и несимметричной криптографии за полиномиальное время. Таким образом, реализация атаки на квантовом компьютере практически ставит под сомнение и стойкость алгоритмов хеширования на основе блочно-симметричных шифров в режимах CBC (Cipher Block Chaining – режим сцепления блоков шифротекста) и CFB (Cipher Feedback Mode – режим обратной связи по шифртексту).

Проведенный анализ возможностей криптоанализа в работе [16] подтверждает, что на основе квантовых алгоритмов Шора и Гровера, и полномасштабного квантового компьютера, алгоритмы симметричной и несимметричной криптографии подвержены взлому за полиномиальное время.

В работах [17, 18] рассмотрена возможность построения каскадного алгоритма UMAC на основе использования в качестве формирования псевдослучайной подкладки на третьем слое бесключевых алгоритмов MASH-1 и MASH-2 (MASH – Modular Arithmetic Secure Hash). Однако приведенные авторами результаты свидетельствуют о том, что алгоритм MASH-1 не обеспечивает требуемые параметры стойкости и универсальности и не может использоваться в модифицированном (усовершенствованном) алгоритме. Алгоритм MASH-2 обеспечивает требуемый уровень криптостойкости и сохраняет свойство универсальности сформированного хеш-кода в усовершенствованном UMAC, однако не обеспечивает его использование в режиме онлайн, из-за значительных вычислительных ресурсов. В работе [19] рассмотрена возможность формирования хеш-функций на основе использования циклических алгеброгеометрических помехоустойчивых кодов. Однако их использование в алгоритме UMAC не рассмотрено, не проведены исследования практической реализации. Такой подход обеспечивает универсальность и позволяет использовать крипто-кодовые конструкции в качестве псевдослучайной подкладки в каскадном алгоритме хеширования. В работах [7, 10, 20] рассмотрены гибридные крипто-кодовые конструкции (ГККК) на основе синтеза классических схем Мак-Элиса и Нидеррайтера. Однако использование двух схем увеличивает не только емкостные затраты на ее практическую реализацию, но и снижает оперативность криптопреобразований, что существенно для использования при формировании MAC-кода. В работе [21] авторы рассматривают использование циклических кодов Рида-Соломона, однако не проводят исследования стойкости такой крипто-кодовой конструкции к атаке Сидельникова, что не позволяет использовать ККК в качестве “гаранта” стойкости хеш-кода. В работе [22] рассмотрена возможность использования крипто-кодовой конструкции Нидеррайтера в постквантовой криптографии, однако данная схема использует два алгоритма (равновесного кодирования и схемы Мак-Элиса), что усложняет формирование псевдоподкладки и практическую реализацию. В работах [23, 24] рассмотрена стойкость универсальных хеш-функций. Авторы подтверждают, что использование универсального хеширования не позволяет однозначно формировать стойкие ко взлому хеш-коды, и предлагают использовать дополнительное шифрование для обеспечения стойкости хеш-кода. Такой подход увеличивает вычислительные за-

траты на их реализацию в режиме онлайн. В работе [25] предлагается использование новой схемы многоадресной аутентификации на основе симметричного алгоритма. Однако в условиях постквантового периода данный алгоритм может быть взломан, а использование модифицированного каскадного алгоритма хеширования обеспечит требуемый уровень стойкости и оперативности. Предложенные в работах [7, 10, 17–25] возможности формирования хеш-кода на основе алгоритма UMAC позволяет обеспечить свойство универсальности, однако требует значительных энергетических затрат на их практическую реализацию и не позволяют обеспечить использование алгоритма в режиме онлайн формирования дайджеста. Формирование модифицированного алгоритма UMAC на основе крипто-кодовых конструкций обеспечит решение задачи аутентификации в условиях постквантовой криптографии, требуемого уровня оперативности (формировании хеш-кода в онлайн режиме) при дальнейшем росте информационных массивов данных, и снижение энергозатрат при их практической реализации.

3. Цель и задачи исследования

Целью исследования является разработка модифицированного алгоритма UMAC на основе крипто-кодовых конструкций и алгоритмов оценки стойкости хеш-кода.

Для достижения поставленной цели необходимо решить следующие задачи:

– рассмотреть требования к алгоритмам универсального и строго универсального классов хеш-функций;

– проанализировать построение каскадного алгоритма UMAC с учетом обеспечения универсальности;

– разработать алгоритмы модифицированного алгоритма UMAC на основе крипто-кодовой конструкции Мак-Элиса на алгеброгеометрических и ущербных кодах;

– разработать алгоритмы оценки стойкости хеш-кодов модифицированного алгоритма UMAC на основе оценки критериев универсальности и строгой универсальности классов хеш-функций.

4. Основные требования к алгоритмам универсального и строго универсального классов хеш-функций

Универсальные классы хеш-функций были впервые предложены в работе [26]. Основные свойства универсальности и строгой универсальности классов хеш-функций изучены в работах [27–29].

Идея универсального хеширования заключается в определении такого набора элементов конечного множества H хеш-функций $h: A \to B$, $|A| = a$, $|B| = b$ (A – множество исходящих сообщений; B – множество MAC-кодов; $|A|$ – число исходящих сообщений; $|B|$ – число возможных состояний MAC-кодов; a – исходящее сообщение; b – состояние MAC-кодов), чтобы случайный выбор функции $h \in H$ беспечивал бы низкую вероятность коллизии, т. е. для любых различных входов x_1 и x_2 вероятность того, что $h(x_1)=h(x_2)$ (вероятность колли-

зии, столкновения) не может превосходить некоторой заранее заданной величины (фиксированной точности) ε:

$$P_{\text{кол}} = P\left(h\left(x_1\right) = h\left(x_2\right)\right) \le \varepsilon,$$

причем вероятность коллизии может быть рассчитана как

$$P_{\text{кол}} = \frac{\delta_H\left(x_1, x_2\right)}{\left|H\right|},$$

где $\delta_H\left(x_1, x_2\right)$ – количество таких хеш-функций в H, при которых значения $x_1, x_2 \in A$, $x_1 \neq x_2$ вызывают коллизию, т.е. $h(x_1)=h(x_2)$.

В работах [18, 30] приведены определения универсального хеширования:

1. Пусть $0<\varepsilon<1$. H является ε-универсальным хеш-классом (сокращенно ε–$U(H, A, B)$), если для двух различных элементов $x_1, x_2 \in A$ существует не больше чем ($|H|\times\varepsilon$) функций $f \in H$ таких, что $h(x_1)=h(x_2)$, если $\delta_H(x_1, x_2) \le \varepsilon\left|H\right|$ для всех $x_1, x_2 \in A$, $x_1 \neq x_2$.

2. Пусть $0<\varepsilon<1$. H является ε-строго универсальным хеш-классом (сокращенно ε–$SU(H, A, B)$) если выполняются следующие условия:

– для каждого $x_1 \in B$, и для каждого $y_1 \in B$,

$$\left|\left\{h \in H : h\left(x_1\right) = y_1\right\}\right| = \left|H\right| / \left|B\right|;$$

– для каждого $x_1, x_2 \in A$, и $x_1 \neq x_2$ для каждого $y_1, y_2 \in B$,

$$\left|\left\{h \in H : h\left(x_1\right) = y_1, h\left(x_2\right) = y_2\right\}\right| \le \varepsilon\left|H\right|.$$

Определение универсального класса хеш-функций эквивалентно определению алгоритма MAC, при котором число различных правил формирования хеш-кода (число ключей), при которых существует коллизия для двух произвольных входных последовательностей ограничено. Число таких ключей не может превосходить значение $\left(P_{\text{кол}} \times \left|H\right|\right)$, где $P_{\text{кол}}$ – вероятность коллизии, $|H|$ – число всех правил (ключей).

Универсальный класс хеш-функций удовлетворяет таким условиям стойкости, как стойкость к прообразу, к другому прообразу и к столкновению [29].

Практического применения схемы аутентификации, эквивалентные универсальным классам хеш-функций, не получили из-за низкой устойчивости к угрозам по навязыванию сообщений. Следовательно, в определении универсального класса хеш-функций нет условия, определяющего устойчивость его к дифференциальному анализу.

Идеи универсальной аутентификации получили развитие в теории безусловной аутентификации с использованием строго универсального хеширования [18, 31]. Так, в [19] приведено определение строго универсального класса хеш-функций, которое эквивалентно определению такого алгоритма формирования кодов контроля целостности и аутентичности данных, при котором будут выполняться следующие правила.

1. Число правил формирования MAC (число ключей), при которых для произвольной входной последовательности значение кода контроля целостности и аутентичности данных не изменяется, ограничено. Число таких ключей не может превосходить значения $|H|/|B|$.

2. Число правил формирования кода контроля целостности и аутентичности данных (число ключей), при которых для двух произвольных входных последовательностей соответствующие им значения MAC не изменяются, ограничено. Число таких ключей не может превосходить значения $P_{кол} \times |H|$.

Вероятность коллизии кодов контроля целостности и аутентичности данных в схеме со строго универсальным хешированием определяется как $P_{кол} \leq \varepsilon$.

Условия строго универсальных классов хеш-функций более "жесткие" по сравнению с требованиями универсальных классов. Схемы аутентификации, эквивалентные строго универсальным классам, обладают целым рядом преимуществ, которые определяются точным значением вероятности коллизий, несклонности к частотному и дифференциальному анализу и др. Однако построение таких схем весьма проблематично. Известны практические схемы, эквивалентные строго универсальным классам, обладающие существенным недостатком – большим объемом ключевых данных, превышающих объем информационных.

Таким образом, для оценки стойкости хеш-кодов, полученных на основе универсальных классов хеш-функций практически достаточно оценить выполнение критериев универсальности и строгой универсальности. Кроме этого знание количества коллизий и равносерное распределение хеш-кодов на всем множестве позволяет использовать данные коллизии в качестве идентификаторов больших объемов данных и значительно сократить время на поиск нужной информации.

5 Анализ построения каскадного алгоритма UMAC с учетом обеспечения универсальности

При дистанционном электронном взаимодействии важно обеспечить процесс распознавания объекта по предъявленным параметрам (индикаторам) и связанного с ним процесса аутентификации. Наиболее остро эта задача стоит в системах управления доступом и подтверждения подлинности и верификации при передаче сообщений [32–40].

Эффективным механизмом контроля целостности и аутентичности информации в современных телекоммуникационных сетях является хеширование информации [33–36]. На сегодняшний день используются коды обнаружения манипуляций (MDC – Manipulation Detection Code) для контроля целостности данных, и коды аутентификации сообщений (MAC – Message Authentication Code) [34, 40–45].

Код аутентификации сообщения UMAC был предложен в работе [35]. Алгоритм основан на семействах универсальных хеш-функций и обеспечивает доказуемую безопасность формируемого MAC [33–42]. При этом безопасность алгоритма обосновывается стойкостью применяемого в схеме UMAC блочного симметричного шифра (БСШ) AES в режиме CBC (сцепление блоков открытого текста) при формировании подкладки для третьего слоя функции UHASH-16 или UHASH-32 (UHASH – универсальная функция хеширования, с фиксированным хеш-кодов в 16 и 32 бита соответсвенно).

В работе [36] построение многослойных хеширующих функций на примере алгоритма UMAC представлено в виде симбиоза многоэтапного ключевого универсального хеширование и использования блочного симметричного шифра для формирования т.н. псевдослучайной подкладки. Применение универсального хеширования в многослойной конструкции UMAC позволяет обеспечить равновероятность формирования хеш-образов для всего множества используемых ключевых данных, на чем и базируется доказательство безопасности алгоритма [17, 18, 25, 37, 41–44]. Применение алгоритма шифрования AES обеспечивает высокую криптостойкость схемы UMAC [33–37, 45].

Схема формирования кодов подлинности сообщений UMAC использует многоуровневую конструкцию универсального хеширования *Hash*(*K,M,Taglen*) и процедуру формирования псевдослучайной подкладки *Pad*. Применение универсального хеширования позволяет обеспечить равновероятность формирования хеш-образов для всего множества используемых ключевых данных, на чем и базируется доказательство безопасности алгоритма [45]. Функция UHash сжимает сообщение, состоящее из трех разных слоев:

– сжатие: первый слой использует быстрое семейство хеш NH, для сжатия сообщения до установленного коэффициента;

– хеш установленной длины: второй слой использует семейство хеш RP, не такое быстрое как NH, но генерирует выход установленной длины с использованием ключа ключа установленной длины;

– усилить и сложить: третий слой использует семейство хеш IP, которое сводит длину своего входа к более соответствующему размеру.

На рис. 2 показан общий принцип работы алгоритма формирования MAC-кода с помощью класса универсальных хеш-функций UMAC-16 и UMAC-32.

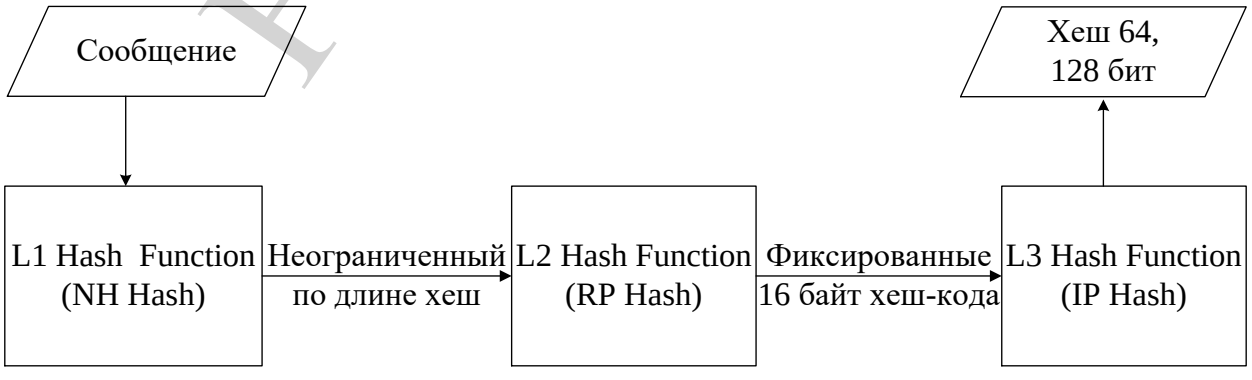


Рис. 2. Схема работы универсального класса UHASH

На первом слое L1 Hash-функции используется NH хеш-функция, результатом которой является разделение сообщения на блоки из 1024 байт. При этом получают сообщения в 128 раз меньше входного (рис. 3).

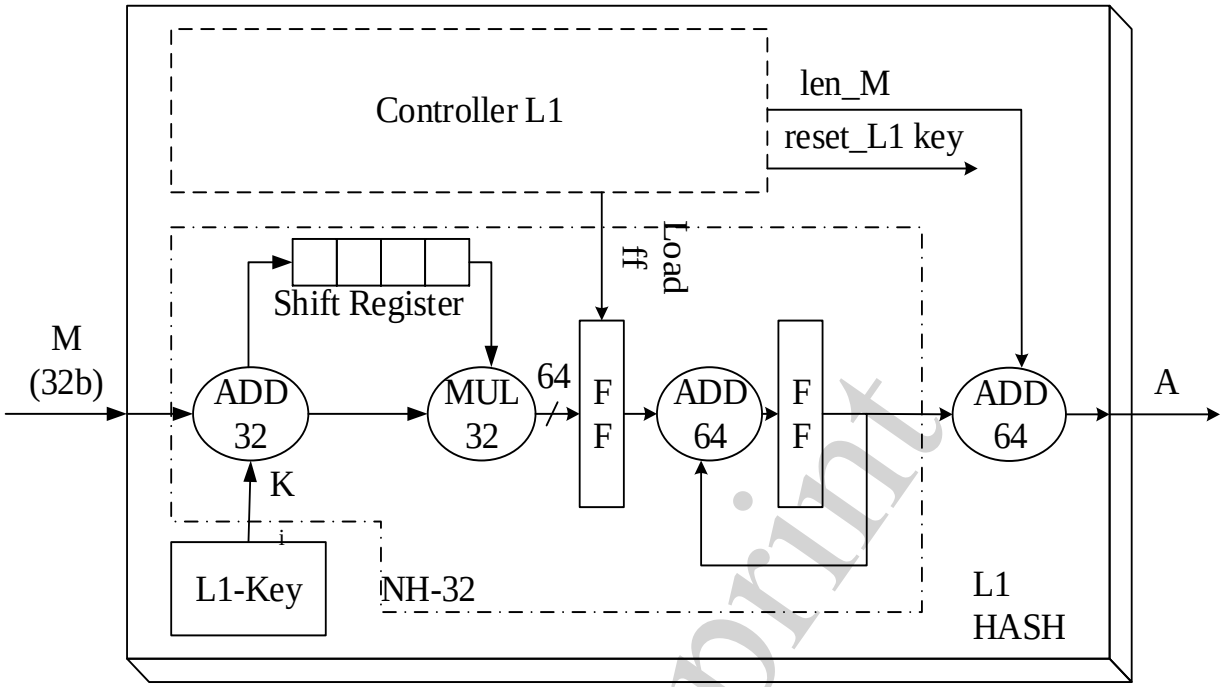


Рис. 3. Схема работы первого слоя L1 Hash-функции: M – входное сообщение, L1-Key – ключ, ADD 32(64) – конкатенация по модулю 32, Shift Register – регистр сдвига, MUL 32 – умножение по модулю 32, FF – данные, Load ff – загрузка данных, reset_L1 key – сброс ключа, len_M – длина сообщения, A – выходной блок

После формирования блока A, полученные данные передаются на второй слой L2 Hash-функции (рис. 4): A – входное сообщение, L2-Key – ключ, ADD – конкатенация по модулю 32, MOD p64 – операция расчета остатка от деления на простое число длинною 64 бит, MULT – мультиплексор, FF – данные, Reset – сброс, SEL – операция выбора, ZERO PAD – нулевой вектор, B – выходной блок, Compare 2^{64}-2^{32} – операция сравнения, K^2 mod p, K mod p – операции с ключами.

На данном слое используется RP. Универсальное семейство хэш-функций RP основывается на полиномиальных вычислениях. Строка, состоящая из n слов длиной n бит, может рассматриваться как полином степени n в области значений, где каждое слово строки служит коэффициентом. Для вычисления хеша необходимо вычислить полином для произвольно выбранной точки (ключа).

Назначение данного слоя заключается в преобразовании результата, полученного на первом слое, с помощью полиномиальной функции POLY. Если длина входного вектора длиннее, чем 1024 бит, то полиномиальная функция использует дополнительные параметры при формировании остатка.

На третьем слое L3 Hash-функции используется функция IP (рис. 5).

Слой с универсальным семейством хэш-функций IP снижает длину своего входа, так как слой хеша RP генерирует выходы значительно большие по сравнению с предполагаемой вероятностью ошибки. Основан на обработке

внутренних данных в области значений (умножая входные слова с ключевыми словами и составляя результаты). Вероятность ошибки с предшествующего слоя перешла на данный и сохраняется.

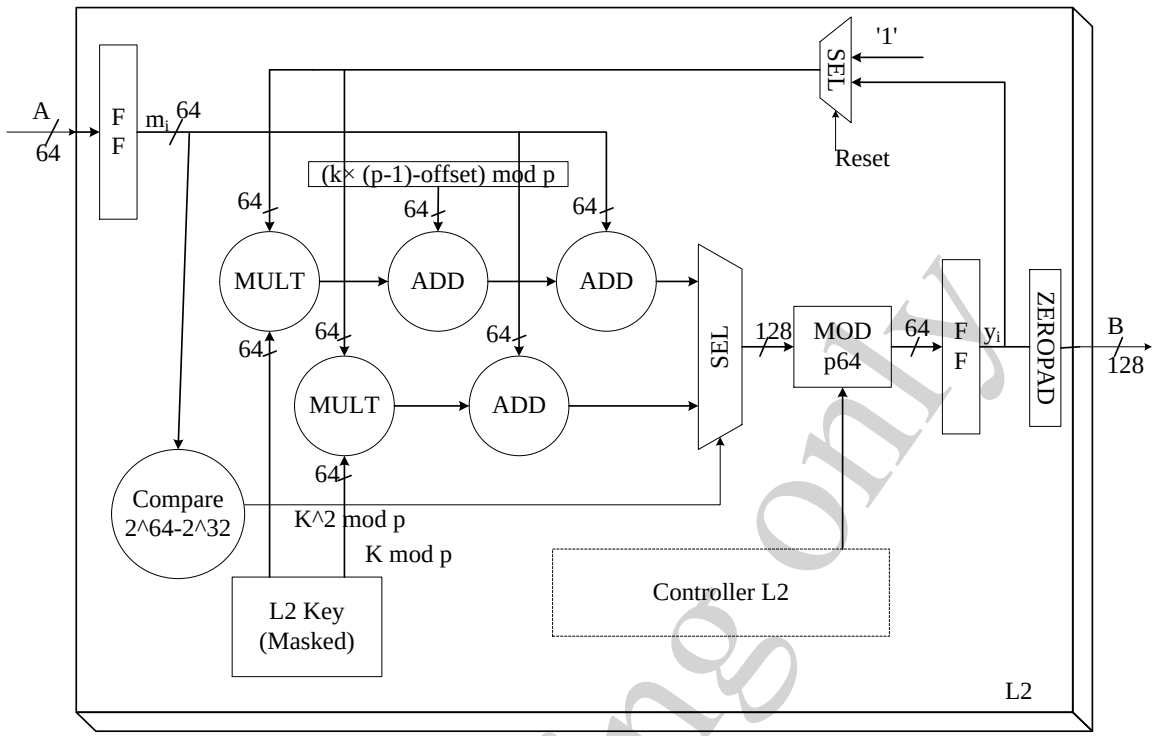


Рис. 4. Схема работы второго слоя L2 Hash-функция

Назначение третьего слоя состоит в преобразовании входного вектора B длиной 16 байт в строку равной 4 байт.

Формирование псевдослучайной подкладки криптографически стойким алгоритмом блочного симметричного шифра AES обеспечивает криптостойкость алгоритма UMAC на уровне стойкости применяемого криптоалгоритма [14]. Данная схема формирования UMAC обладает потенциально высокими показателями эффективности, что обеспечивается наложением на формируемые хеш-коды *Y=Hash*(*K, M, Taglen*) псевдослучайных подкладок *Pad=Hash*(*K, Nonce, Taglen*). Процедура наложения эквивалентна операции поразрядного сложения. Данный подход является классическим методом реализации алгоритма UMAC.

Рассмотренная многослойная конструкция при формировании UMAC обладает потенциально высокими показателями эффективности. Но после применения на последнем слое наложения псевдослучайных подкладок, алгоритм формирования UMAC теряет свойство “универсальности” хеширования и его коллизионные свойства существенно ухудшаются. Это объясняется использованием блочного симметричного шифрования, которое не гарантирует сохранения свойства универсальности результирующего MAC-кода [18, 19, 32].

Для обеспечения повышения не только криптостойкости алгоритмов шифрования сообщений для передачи по каналам связи, но сохранения универсальности, в работах [19, 32] предлагается использовать применение универсального хеширования на основе модулярных преобразований. Обеспечение

высокой криптостойкости переданных сообщений происходит за счет невозможности за вычислительное время провести расшифрование этих сообщений. Но данный метод устойчив только при существующих вычислительных мощностях, а при появлении высокопроизводительных квантовых компьютеров увеличится и риск его взлома.

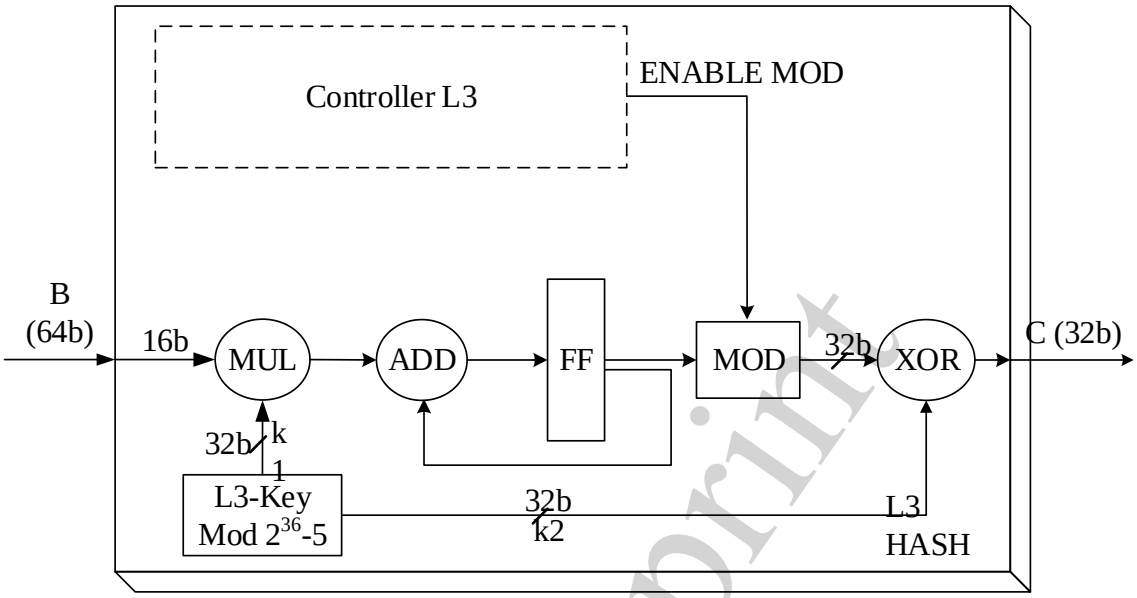


Рис. 5. Схема работы третьего слоя L3 Hash-функции: B – входное сообщение, L3-Key – ключ, ADD – конкатенация, MOD – операция расчета остатка от деления, MUL – умножение, FF – данные, Reset – сброс, SEL – операция выбора, Enable Mod – команда, позволяющая выполнять MOD, C – выходной блок

В работе [19] рассмотрено применение универсального хеширования на основе модулярных преобразований с помощью алгоритма RSA (Rivest, Shamir, Adleman), который основан на эллиптических кривых и вычислительной сложности задачи факторизации больших чисел. На заключительном этапе хеширования данный подход обеспечивает обработку криптографически сильной функцией строго универсального хеширования на основе модулярных преобразований с использованием цикловых функций $f\left(x_i, H_{i-1}\right) = \left(x_i \oplus H_{i-1}\right)^e \bmod\left(N\right)$ и/или $f\left(x_i, H_{i-1}\right) = \left(\alpha^{x_i \oplus H_{i-1}}\right) \bmod\left(p\right)$. При этом основная часть информационных данных обрабатывается первыми слоями универсального хеширования. А псевдослучайная подкладка обрабатывается криптографически сильной функцией строго универсального хеширования на основе модулярных преобразований с помощью алгоритма RSA. Однако такой подход не обеспечивает оперативности криптопреобразований, и его стойкость в условиях постквантовой криптографии не удовлетворяет требованиям.

В работе [32] предлагается усовершенствованный метод формирования кодов контроля целостности и аутентичности данных на основе использования алгоритма UMAC. При этом первые два слоя являются высокоскоростными, но криптографически слабыми схемами универсального хеширования, последний

слой предлагается реализовать с использованием разработанной безопасной (криптографически сильной) схемы строго универсального хеширования на основе модулярных преобразований.

Формирование кодов аутентификации сообщений с использованием ключевого хеширования, построенного на основе бесключевого алгоритма MASH-2 с изменяемыми векторами инициализации, в определенных случаях позволяет строить универсальные и строго универсальные классы хеш-функций. Однако не для всех значений начальных параметров (простых чисел p и q) это условие выполнимо.

Формально предлагаемая схема каскадного формирования кодов контроля целостности и аутентичности данных с использованием модулярных преобразований представлена на рис. 6. Такой подход позволяет обеспечить универсальность и стойкость хеш-кодов, но практически не применим в онлайн режиме формирования хеш-кода из-за низкой скорости формирования псевдослучайной подкладки на основе бесключевого алгоритма MASH-2.

Таким образом, для устранения выявленного недостатка предлагается в качестве механизма формирования псевдоподложки для третьего слоя каскадного алгоритма хеширования UMAC использовать крипто-кодовые конструкции на алгеброгеометрических кодах и ущербных кодах.

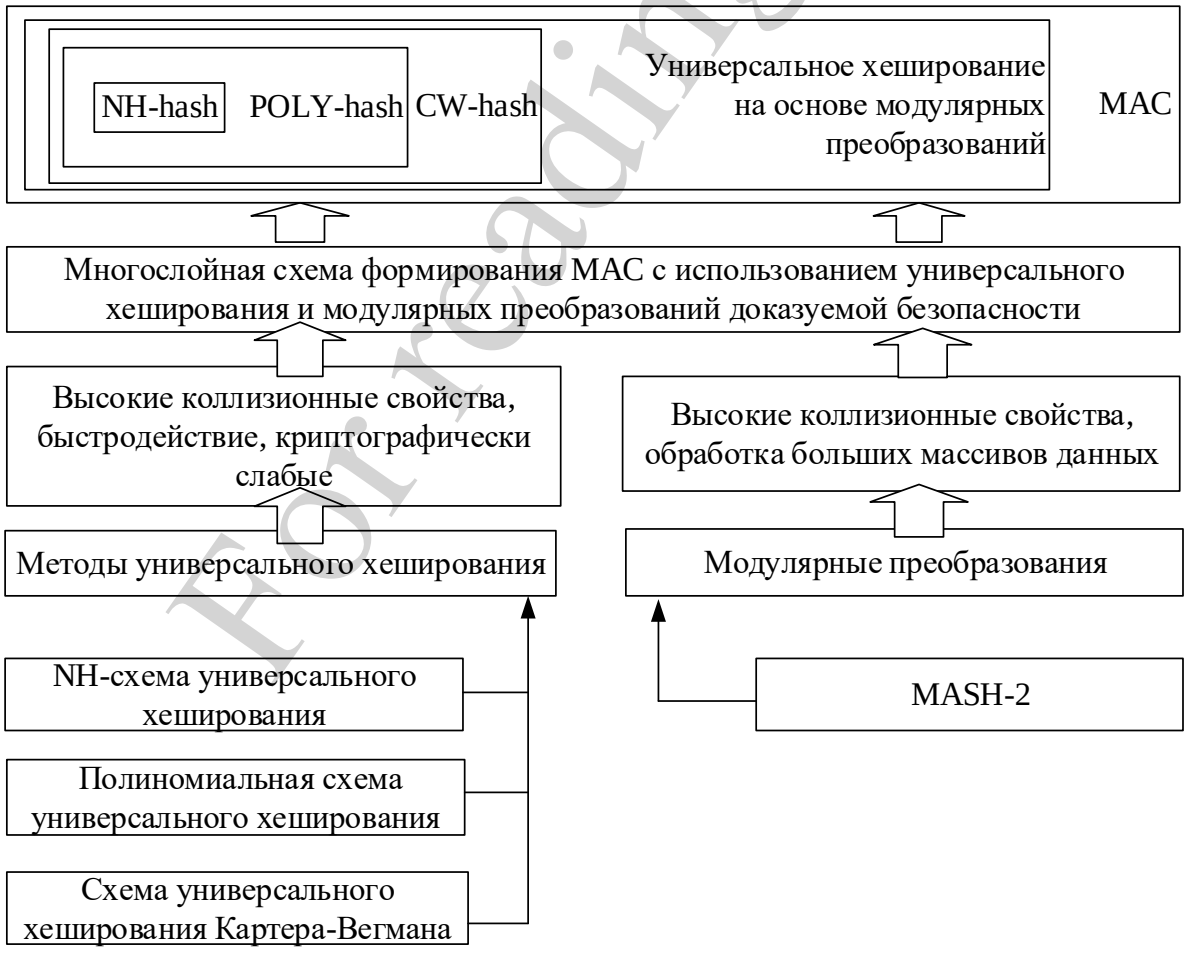


Рис. 6. Схема каскадного формирования кодов контроля целостности и аутентичности данных с использованием бесключевого алгоритма хеширования MASH-2

В работах [14, 46, 47] рассмотрены основные подходы к обеспечению снижения энергетических затрат на практическую реализацию ККК на ЕС, МЕС, DC. Такой подход позволяет формировать модифицированный алгоритм UMAC в зависимости от требований и вычислительные возможностей.

6. Разработка алгоритмов модифицированного UMAC на основе ККК Мак-Элиса на ЕС, МЕС и DC

В работах [14, 47–50] рассмотрены алгоритмы построения ККК Мак-Элиса на ЕС (МЕС), DC, которые могут использоваться для формирования псевдослучайной подкладки.

Так для построения подкладки на ККК Мак-Элиса на ЕС используется следующий алгоритм:

– шаг 1. Ввод информации, подлежащей кодированию (используем открытый текст сообщения). Ввод открытого ключа G_X^{EC};

– шаг 2. Кодирование информации эллиптическим кодом. Формирование кодового слова c_X эллиптического кода, заданного матрицей G_X^{EC} (формирование первой части подкладки);

– шаг 3. Формирование вектора ошибок e, вес которого не превышает $\leq t$ – исправляющую способность эллиптического кода (формирование второй части подкладки);

– шаг 4. Формирование кодограммы (подкладки) $c_X^* = c_X + e$.

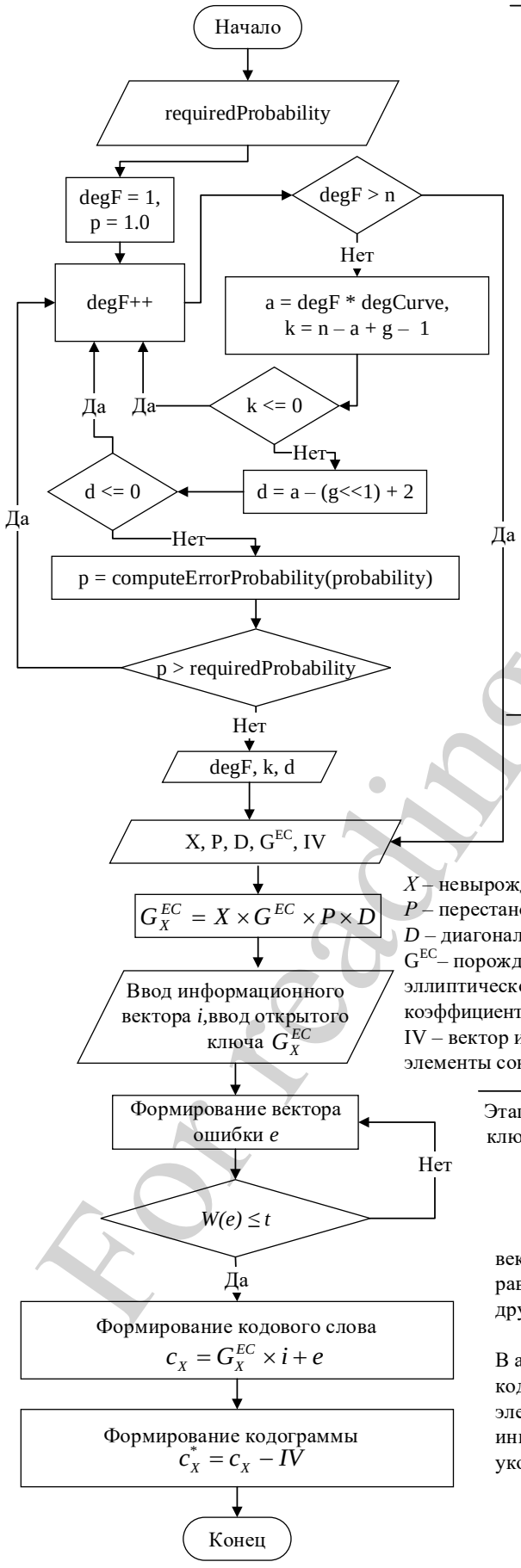
Таким образом, сформированный на ЕС код является подкладкой, которая используется на третьем слое модифицированного алгоритма UMAC на ККК Мак-Элиса. Такой подход обеспечивает сохранение свойства универсальности полученного хеш-кода и требуемый уровень оперативности. Однако требует значительных энергетических затрат для обеспечения требуемого уровня стойкости (необходимо построение ККК над GF (2^{10}–2^{13}). Для снижения энергетических затрат предлагается использовать ККК Мак-Элиса на МЕС, что позволяет снизить энергетические затраты (построение ККК над GF (2^6–2^8) и обеспечить требуемый уровень стойкости.

В работе [51] (рис. 7), и в работе [14] (рис. 8) представлены алгоритмы формирования подкладки для формирования хеш-кода в модифицированном алгоритме UMAC на ККК Мак-Элиса с укороченными или удлиненными МЕС.

Результаты исследования показали, что применение на практике данного подхода способствует как повышению криптостойкости реализации формирования кода аутентификации, так и поднимает вопрос быстродействия проведения связанных с данным процессом операций.

Для построения крипто-кодовых конструкций на основе модифицированных (укороченных или удлиненных) циклических кодах на эллиптических кривых используются параметры, представленные в табл. 1. В табл. 2 приведены параметры несимметричных криптосистем на соответствующих ККК.

В работах [14, 49–51] рассмотрены практические алгоритмы построения модифицированного алгоритма UMAC на КККК с МЕС.



Этап 1. Установка параметров кода

requiredProbability – заданная вероятность искажения блока,
n – общее число символов в коде (длина кода),
k – число информационных символов,
d – минимальное расстояние кодовых комбинаций по Хеммингу,
g – род кривой,
degF – степень генераторной функции,
degCurve – степень кривой.

Этап 2. Формирование личного и открытого ключей несимметричной криптосистемы, ввод информационной посылки

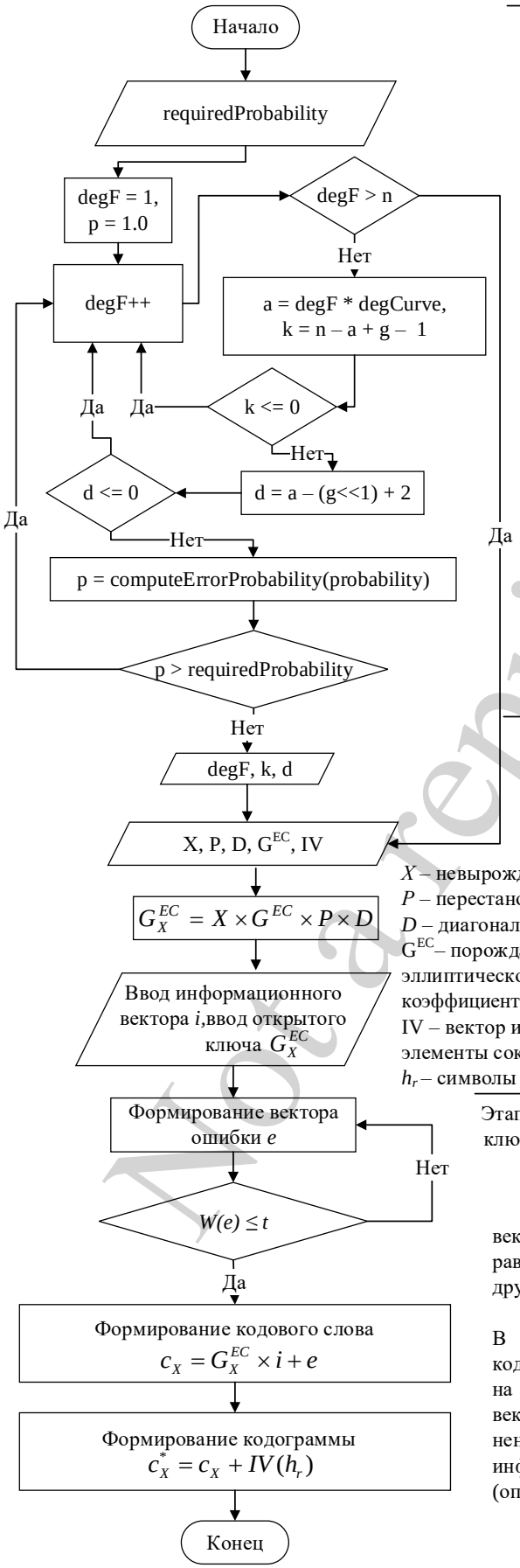
X – невырожденная $k \times k$ матрица над *GF(q)*,
P – перестановочная $n \times n$ матрица над *GF(q)*,
D – диагональная $n \times n$ матрица над *GF(q)*,
G^{EC} – порождающая $k \times n$ матрица эллиптического кода над *GF(q)*,a_i – набор коэффициентов многочлена кривой $a_1 \ldots a_6$,
IV – вектор инициализации, IV= $|h| = ½\ k$ – элементы сокращения

Этап 3. Формирование сеансового ключа и кодограммы (подкладки)

вектор *e* формируется случайно, равновероятно и независимо от других закрытых текстов

В алгоритм UMAC поступает кодовое слово *c* без нулевых элементов вектора инициализации (операция укорочения)

Рис. 7. Алгоритм формирования кодограммы в модифицированной ККК Мак-Элиса с укороченным модифицированным кодом (МЕС)



Этап 1. Установка параметров кода

requiredProbability – заданная вероятность искажения блока,
n – общее число символов в коде (длина кода),
k – число информационных символов,
d – минимальное расстояние кодовых комбинаций по Хеммингу,
g – род кривой,
degF – степень генераторной функции,
degCurve – степень кривой.

Этап 2. Формирование личного и открытого ключей несимметричной криптосистемы, ввод информационной посылки

X – невырожденная $k \times k$ матрица над *GF(q)*,
P – перестановочная $n \times n$ матрица над *GF(q)*,
D – диагональная $n \times n$ матрица над *GF(q)*,
G^{EC} – порождающая $k \times n$ матрица эллиптического кода над *GF(q)*, a_i – набор коэффициентов многочлена кривой $a_1 \ldots a_6$,
IV – вектор инициализации, $IV = |h| = ½\ k$ – элементы сокращения
h_r – символы удлинения, $|h_r| = ½\ k$

Этап 3. Формирование сеансового ключа и кодограммы (подкладки)

вектор *e* формируется случайно, равновероятно и независимо от других закрытых текстов

В алгоритм UMAC поступает кодовое слово *c* со вставленными на места нулевых элементов вектора инициализации ненулевых элементов информационных символов (операция удлинения)

Рис. 8. Алгоритм формирования кодограммы в модифицированной ККК Мак-Элиса с удлиненным модифицированным кодом (МЕС)

Таблица 1

Основные (*n, k, d*)-параметры *МЕС*

Параметры	укороченные *МЕС*	удлиненные *МЕС*
(*n, k, d*) параметры кода, построенного через отображение вида $\varphi: X \rightarrow P^{k-1}$	$n = 2\sqrt{q} + q + 1 - x$, $k \geq \alpha - x$, $d \geq n - \alpha$, $\alpha = 3 \times \deg F$, $k + d \geq n$	$n = 2\sqrt{q} + q + 1 - x + x_1$, $k \geq \alpha - x + x_1$, $d \geq n - \alpha$, $\alpha = 3 \times \deg F$
(*n, k, d*) параметры кода, построенного через отображение вида $\varphi: X \rightarrow P^{r-1}$	$n = 2\sqrt{q} + q + 1 - x$, $k \geq n - \alpha$, $d \geq \alpha$, $\alpha = 3 \times \deg F$, $k + d \geq n$	$n = 2\sqrt{q} + q + 1 - x + x_1$, $k \geq n - \alpha$, $d \geq \alpha$, $\alpha = 3 \times \deg F$

Таблица 2

Основные параметры модифицированных криптосистем на основе ККК Мак-Элиса на *МЕС*

Параметры	укороченные *МЕС*	удлиненные *МЕС*
размерность секретного ключа	$l_{K+} = x \times \left\lceil \log_2 \left(2\sqrt{q} + q + 1 \right) \right\rceil$	$l_{K+} = (x - x_1) \times \log_2 \left(2\sqrt{q} + q + 1 \right)$
размерность информационного вектора	$l_I = (\alpha - x) \times m$	$l_I = (\alpha - x + x_1) \times m$
размерность криптограммы	$l_S = \left(2\sqrt{q} + q + 1 - x \right) \times m$	$l_S = \left(2\sqrt{q} + q + 1 - x + x_1 \right) \times m$
относительная скорость кодирования	$R = (\alpha - x) / \left(2\sqrt{q} + q + 1 - x \right)$	$R = (\alpha - x + x_1) / \left(2\sqrt{q} + q + 1 - x + x_1 \right)$

Предложенный подход позволит формировать псевдослучайную подкладку с дальнейшим сокращением вычислительных ресурсов за счет уменьшения вычислительного поля эллиптических кривых на основании нанесения ущерба (сокращения или удлинения кодовой последовательности).

Схематично процесс подтверждения целостности информации при передаче от отправляющей к принимающей стороне на основании проверки верификаций кодограмм и хеш-кодов с использованием ККК Мак-Элиса на МЕС представлено на рис. 7.

Применение алгоритма UMAC на предложенных крипто-кодовых конструкциях позволит выявлять модификации открытого текста при передаче через открытый канал. При разработке математической модели формирования хеш-кода в алгоритме UMAC, используется псевдослучайная последовательность, которая обеспечивает криптостойкость данного хеш-кода. В качестве алгоритма формирования подложки выступает крипто-кодовая конструкция Мак-Элиса на МЕС.

Нанесение модификационных изменений на эллиптические коды позволяет снизить нагрузку на вычислительные ресурсы и приводит к повышению оперативности формирования МАС-кодов в режиме реального времени.

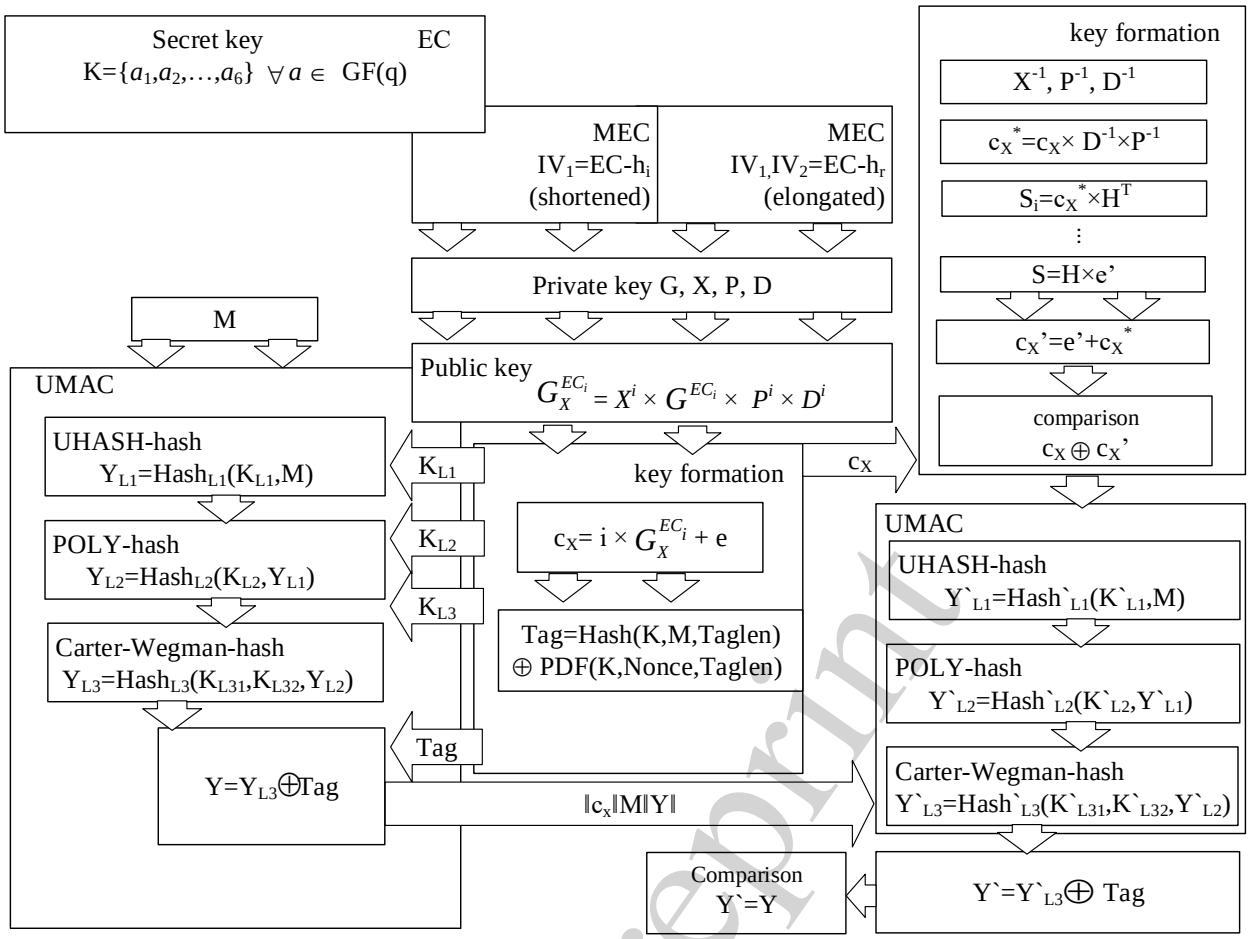
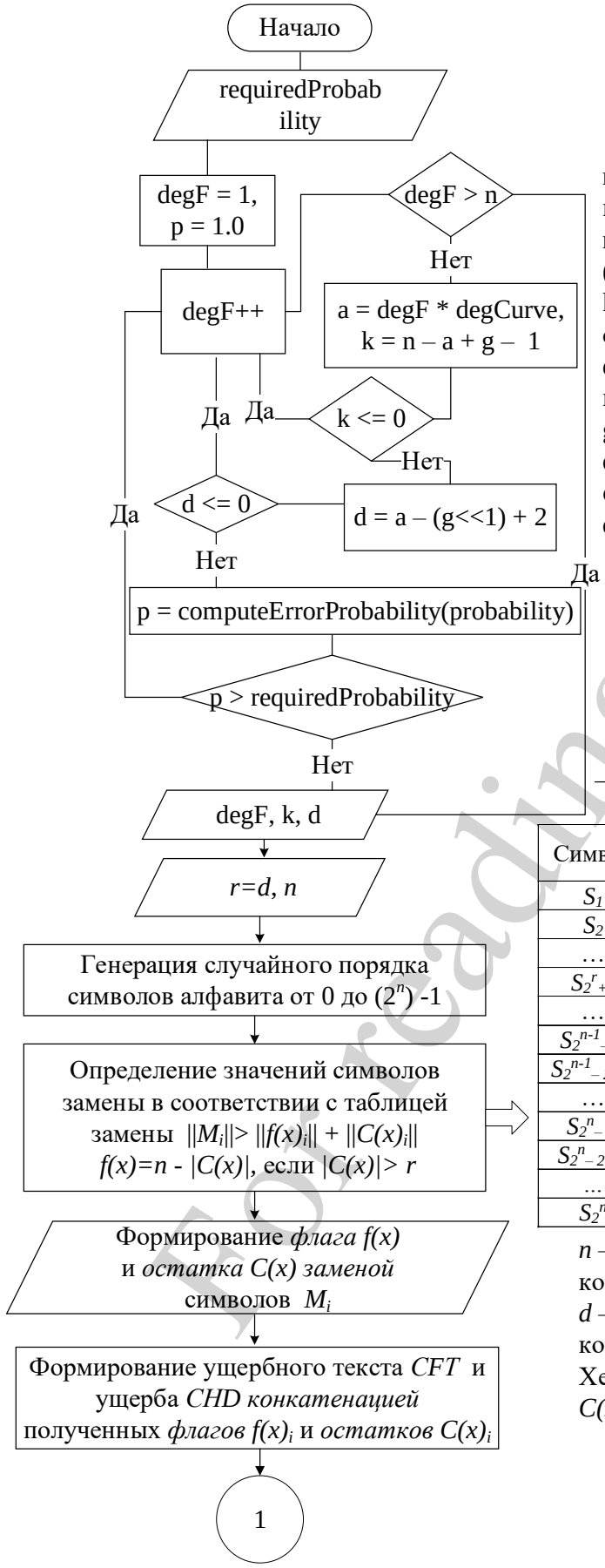


Рис. 7. Схема передачи сообщения от отправителя к получателю и проверки целостности полученного через сравнение кодограмм и хеш-кодов с использованием ККК Мак-Элиса на МЕС (secret key – секретный ключ; key formation – формирование ключа; private key – личный ключ; shortened - укороченный; elongated - удлиненный)

В работе [50] предложено в качестве механизма формирования псевдослучайной подкладки третьего слоя UMAC использовать крипто-кодовую конструкцию Мак-Элиса с использованием ущербных кодов. Основные идеи ущербной криптографии предложены в работах [52, 53]. Такой подход позволяет использовать гибридную конструкцию ККК (ГККК) с многоканальной криптографией и обеспечивает максимальную скорость формирования подложки с требуемым уровнем стойкости. Кроме того, позволяет сократить энергетические затраты (построение ГККК над GF (2^4–2^6) с сохранением требуемого уровня стойкости хеш-кода. На рис. 9, 10 представлен алгоритм использования ГККК Мак-Элиса для формирования подкладки.

Предложенный подход позволит формировать псевдослучайную подкладку с дальнейшим сокращением вычислительных ресурсов за счет уменьшения вычислительного поля эллиптических кривых на основании нанесения ущерба (сокращения или удлинения кодовой последовательности).



Этап 1. Установка параметров кода

requiredProbability – заданная вероятность искажения блока,
n – общее число символов в коде (длина кода),
k – число информационных символов,
d – минимальное расстояние кодовых комбинаций по Хеммингу,
g – род кривой,
degF – степень генераторной функции,
degCurve – степень кривой.

Этап 2. Нанесение

Символ	Длина остатка	Остаток $C(x)$	Флаг $f(x)$
S_1	r	0^r	$0^{n-r-1}1$
S_2	r	$0^{r-1}1$	$0^{n-r-1}1$
…	…	…	…
$S_2^{r}{}_{+1}$	$r+1$	0^{r+1}	$0^{n-r-2}1$
…	…	…	…
$S_2^{n-1}{}_{-2^r}$	$n-2$	1^{n-2}	01
$S_2^{n-1}{}_{-2^r+1}$	$n-1$	0^{n-1}	1
…	…	…	…
$S_2^{n}{}_{-2^r}$	$n-1$	1^{n-1}	1
$S_2^{n}{}_{-2^r+1}$	r	0^r	0^{n-r}
…	…	…	…
S_2^n	r	1^r	0^{n-r}

n – общее число символов в коде (длина кода),
d – минимальное расстояние кодовых комбинаций по Хеммингу, $f(x)$ – флаг,
$C(x)$ – остаток

Рис. 9. Формирование подкладки на основе ГККК Мак-Элиса с ущербными кодами

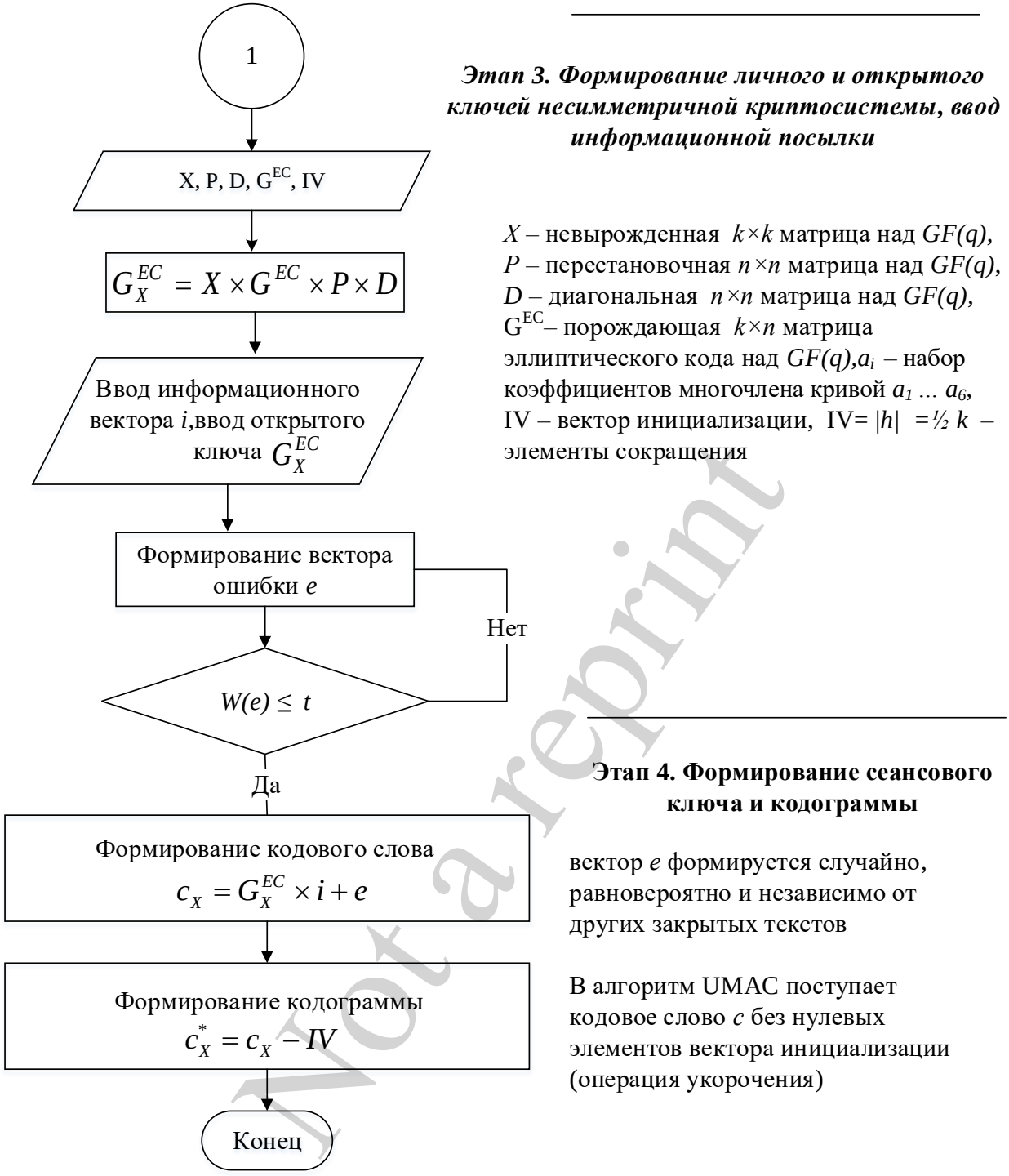


Рис. 10. Формирование подкладки на основе ГККК Мак-Элиса с ущербными кодами

7. Разработка алгоритмов оценки стойкости хеш-кодов модифицированного алгоритма UMAC на основе оценки критериев универсальности и строгой универсальности классов хеш-функций

В основе предлагаемых алгоритмов оценки стойкости хеш-кодов модифицированного алгоритма **UMAC** (статистического исследования коллизионных свойств формируемых элементов $h(x)$) лежит эмпирическая оценка максимумов числа ключей (правил хеширования) при которых:

1) для произвольных x_1, $x_2 \in A$, $x_1 \neq x_2$, выполняется равенство:

$$h(x_1)=h(x_2); \quad (1)$$

2) для произвольных $x_1 \in A$ и $y_1 \in B$ выполняется равенство:

$$h(x_1)=y_1; \quad (2)$$

3) для произвольных x_1, $x_2 \in A$, $x_1 \neq x_2$ и y_1, $y_2 \in B$ выполняются равенства:

$$h(x_1)=y_1,\ h(x_2)=y_2. \quad (3)$$

Оценка по первому критерию соответствует проверке выполнимости условия для универсального класса хеш-функций, оценка по второму и третьему критерию – условий для строго универсального класса хеш-функций.

Для оценки приведенных равенств вводят следующие обозначения [54]:

$$n_1(x_1, x_2) = \left|\left\{h \in H : h(x_1) = h(x_2)\right\}\right|,$$

$$x_1, x_2 \in A,\ \ x_1 \neq x_2,$$

$$n_2(x_1, y_1) = \left|\left\{h \in H : h(x_1) = y_1\right\}\right|,\ \ x_1 \in A,\ \ y_1 \in B;$$

$$n_3(x_1, x_2, y_1, y_2) = \left|\left\{h \in H : h(x_1) = y_1, h(x_2) = y_2\right\}\right|,$$

$$x_1, x_2 \in A,\ \ x_1 \neq x_2,$$

$$y_1, y_2 \in B.$$

Первый показатель $n_1(x_1, x_2)$ характеризует число правил хеширования (правил формирования MAC), при которых для заданных x_1, $x_2 \in A$, $x_1 \neq x_2$, выполняется равенство (1), т. е. число ключей, при которых существует коллизия (совпадение MAC) для двух входных последовательностей x_1 и x_2.

Второй показатель $n_2(x_1, y_1)$ характеризует число правил хеширования (правил формирования MAC), при которых для заданных $x_1 \in A$, $y_1 \in B$ выполняется равенство (2), т. е. число ключей, при которых для входной последовательности x_1 значение хеш-кода (MAC) y_1 не изменяется.

Третий показатель $n_3(x_1, x_2, y_1, y_2)$ характеризует число правил хеширования (правил формирования MAC). Для заданных $x_1, x_2 \in A,\ \ x_1 \neq x_2,\ \ y_1, y_2 \in B$ выполняется равенство (3), т. е. число ключей, при которых для двух входных последовательностей x_1 и x_2 соответствующие им значения хеш-кодов (MAC) y_1 и y_2 не изменяются.

Поскольку число ключей не должно превосходить соответствующих им значений $P_{\text{кол}} \cdot |H|$, $|H|/|B|$ и $P_{\text{кол}} \cdot |H|$, то представляет интерес максимальное число таких ключей для каждого из рассматриваемого набора элементов.

Для проведения исследований необходимо определить максимумы этих величин, а затем сравнить полученные результаты с числом $\left(P_{\text{кол}} \cdot |H|\right)$ (для первого критерия), с числом $\left(|H|/|B|\right)$(для второго критерия) и числом $\left(P_{\text{кол}} \cdot |H|\right)$ (для третьего критерия).

Таким образом, в качестве статистических показателей оценки коллизионных свойств, используют [54]:

математические ожидания $m(n_1)$, $m(n_2)$ и $m(n_3)$ максимумов числа правил хеширования (правил формирования МАС), при которых выполняются равенства (1), (2) и (3), соответственно;

дисперсии $D(n_1)$, $D(n_2)$ и $D(n_3)$, характеризующие рассеивание значений числа правил хеширования (правил формирования МАС), при которых выполняются равенства (1), (2) и (3), относительно их математических ожиданий $m(n_1)$, $m(n_2)$ и $m(n_3)$, соответственно.

Оценку коллизионных свойств по приведенным критериям универсальности и строгой универсальности производят в среднестатистическом смысле. Так, при постановке эксперимента используют ограниченный набор элементов $x_1, x_2 \in A$, $x_1 \neq x_2$ и соответствующих им хеш-образов (МАС) $y_1, y_2 \in B$, рассматривая соответствующие результаты как выборку из генеральной совокупности.

Естественной оценкой $\tilde{m}$ для математического ожидания m случайной величины X является среднее арифметическое ее наблюдаемых значений X_i (или статистическое среднее) [19]:

$$\tilde{m} = \frac{1}{N}\sum_{i=1}^{N} X_i,$$

где N – количество реализаций случайной величины X.

Оценка дисперсии $\tilde{D}$ случайной величины X определяется выражением:

$$\tilde{D} = \frac{1}{N-1}\sum_{i=1}^{N}\left(X_i - \tilde{m}\right)^2.$$

В силу центральной предельной теоремы теории вероятностей при больших значениях количества реализаций N среднее арифметическое будет иметь распределение, близкое к нормальному, с математическим ожиданием [19]:

$$m\left[\tilde{m}\right] \approx \tilde{m}$$

и средним квадратическим отклонением

$$\sigma[\tilde{m}] \approx \frac{\sigma}{\sqrt{N}},$$

где σ – среднее квадратическое отклонение оцениваемого параметра.

При этом вероятность того, что оценка $\tilde{m}$ отклонится от своего математического ожидания меньше чем на ε (доверительная вероятность), равна [19]:

$$P\left(|\tilde{m} - m| < \varepsilon\right) \approx 2\Phi\left(\frac{\varepsilon}{\sigma[\tilde{m}]}\right), \quad (4)$$

где Φ(x) – функция Лапласа, определяется выражением:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-\frac{t^2}{2}} \mathrm{d}t. \quad (5)$$

При проведении экспериментальных исследований коллизионных свойств предложено использовать методы статистической проверки гипотез и математической статистики [18, 19]:

1. Из генеральной совокупности случайной величины X формируют выборку следующим образом:

– для среднестатистической оценки математического ожидания $m(n_1)$ и дисперсии $D(n_1)$ в качестве случайной величины выступает максимум $n_1(x_1, x_2)$ при которых выполняется равенство $h(x_1)=h(x_2)$, следовательно, выборку объема N: $X_1, X_2, ..., X_N$ сформируем отбором N множеств, в каждом из которых содержится M пар элементов $x_1, x_2 \in A$, $x_1 \neq x_2$ и оценивается $n_1(x_1, x_2)$, т. е. общий объем формируемых пар элементов $x_1, x_2 \in A$, $x_1 \neq x_2$ составит NM;

– для среднестатистической оценки $m(n_2)$ и $D(n_2)$ в качестве случайной величины выступает максимум $n_2(x_1,y_1)$ при которых выполняется равенство $y_1=h(x_1)$, следовательно, выборку объема N: $X_1, X_2, ..., X_N$ сформируем отбором N множеств, в каждом из которых содержится M пар элементов $x_1 \in A$, $y_1 \in B$, и оценивается $n_2(x_1,y_1)$. Общий объем формируемых пар элементов $x_1 \in A$, $y_1 \in B$, составит NM;

– для среднестатистической оценки $m(n_3)$ и $D(n_3)$ в качестве случайной величины выступает максимум $n_3(x_1, x_2, y_1, y_2)$ при которых выполняются равенства $y_1=h(x_1)$ и $y_2=h(x_2)$, следовательно, выборку объема N: $X_1, X_2, ..., X_N$ сформируем отбором N множеств, в каждом из которых содержится M четверок элементов $x_1, x_2 \in A$, $x_1 \neq x_2$, $y_1, y_2 \in B$ и оценивается $n_3(x_1, x_2, y_1, y_2)$, общий объем формируемых четверок составит NM.

2. При экспериментальных исследованиях коллизионных свойств хеширования оценивают среднее арифметическое $\tilde{m}(n_i)$ наблюдаемых значений максимумов n_i и дисперсию $\tilde{D}(n_i)$, $i = 1,2,3$.

3. Достоверность полученных среднестатистических оценок обосновывают следующим образом. Фиксируют точность ε и рассчитывают значения функции

Лапласа, которые, в соответствии с доверительной вероятностью, дадут соответствующие доверительные вероятности:

$$P\left(\left|\tilde{m}\left(n_i\right)-m\left(n_i\right)\right|<\varepsilon\right)\approx 2\Phi\left(\frac{\varepsilon}{\sigma\left[\tilde{m}\left(n_i\right)\right]}\right),$$

$$\sigma\left[\tilde{m}\left(n_i\right)\right]\approx\frac{\sqrt{\tilde{D}\left(n_i\right)}}{\sqrt{N}},\ i=1,2,3.$$

При обратной постановке задачи, т. е. для фиксированной доверительной вероятности P_σ при объеме выборки N доверительный интервал определяют следующим образом:

$$\tilde{m}\left(n_i\right)-t_\rho\cdot\sigma\left[\tilde{m}\left(n_i\right)\right]<m\left(n_i\right)<\tilde{m}\left(n_i\right)+t_\rho\cdot\sigma\left[\tilde{m}\left(n_i\right)\right],$$

$$i=1,2,3,$$

где t_ρ – корень уравнения $2\Phi\left(t_\rho\right)=P_д$.

Алгоритм проверки хеш-кодов на выполнение правил универсального класса хеш-функций приведен на рис. 8.

Реализация алгоритма можно описать следующими шагами его выполнения.

Шаг 1. Формируются входные сообщения.

Шаг 2. Формируются ключи.

Шаг 3. По каждому входному сообщению I_i с применением ключей K_j формируют их хеш-коды H_{ij}.

Шаг 4. Проводим последовательное сравнение полученных хеш-кодов H_{ij} по одинаковому ключу K_j по всем входным сообщениям между собой на основании такого условия:

если значения хешей совпадают (H_{ij}=H_{ij+1}), что свидетельствует о возникновении коллизии (L_j), то к счетчику коллизий прибавляется 1:

$$L_j=\sum_{j=1}^{N}L_{j-1}+1,$$

где L_{j-1} – предыдущее значение счетчика коллизий по j-ому ключу.

Шаг 5. В рамках одного сообщения по всем ключам выбираем максимальное значение коллизий L_{maxi}.

Шаг 6. Рассчитываем среднее арифметическое или центральное значение по максимумам количества коллизий путем нахождения их математического ожидания $M(L_{maxi})$:

$$M\left(L\max_i\right) = \sum_{i=1}^{M} x_i \times p_i$$

Шаг 7. Рассчитываем значение разброса максимальных значений коллизий вокруг их центрального значения путем нахождения дисперсии ($D(L_{maxi})$) по максимумам количества коллизий:

$$D\left(L_{\max i}\right) = M\left(x - M\left(x\right)\right)^2.$$

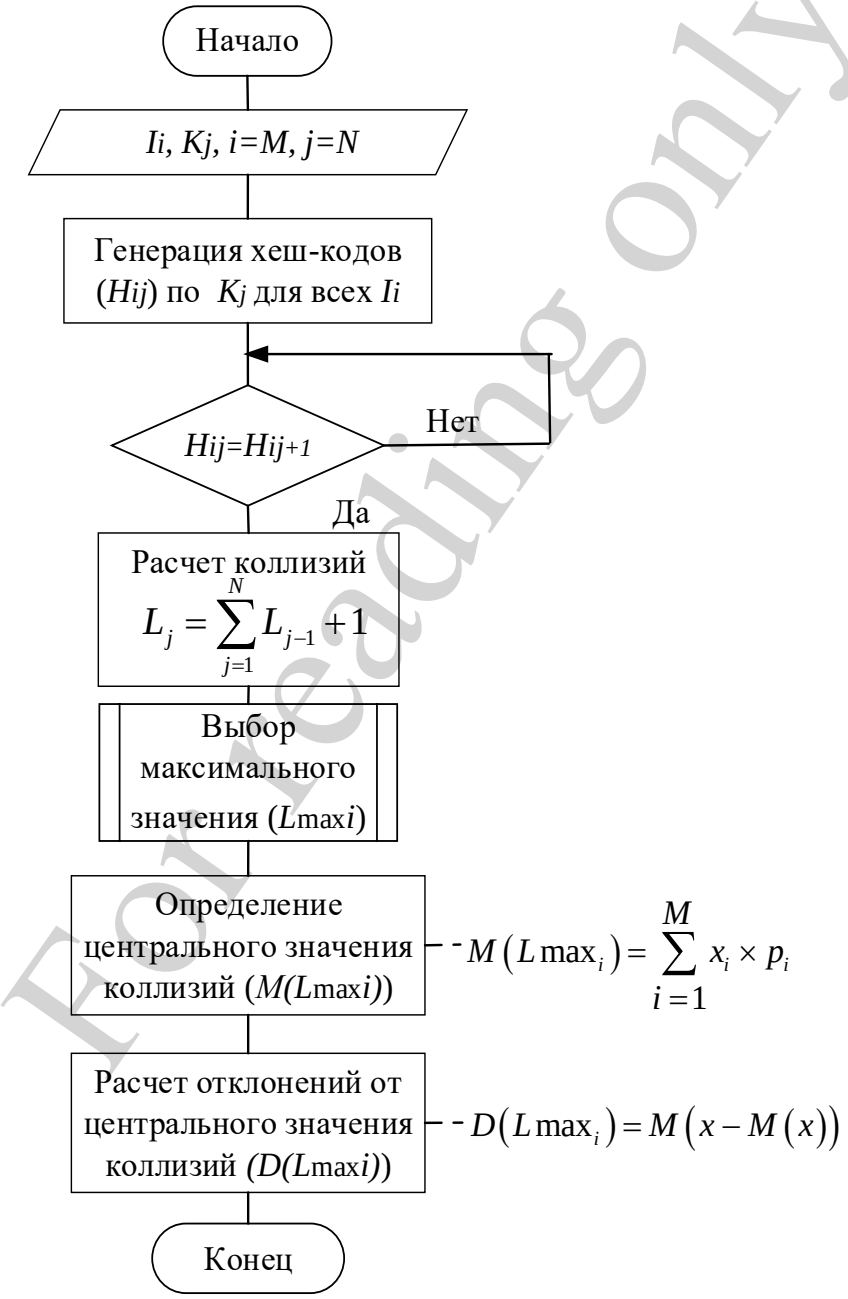


Рис. 8. Алгоритм проверки хеш-кодов на выполнение требований универсального класса хеш-функций

Алгоритм проверки хеш-кодов на выполнение требований строго универсального класса хеш-функций по первому критерию приведен на рис. 9.

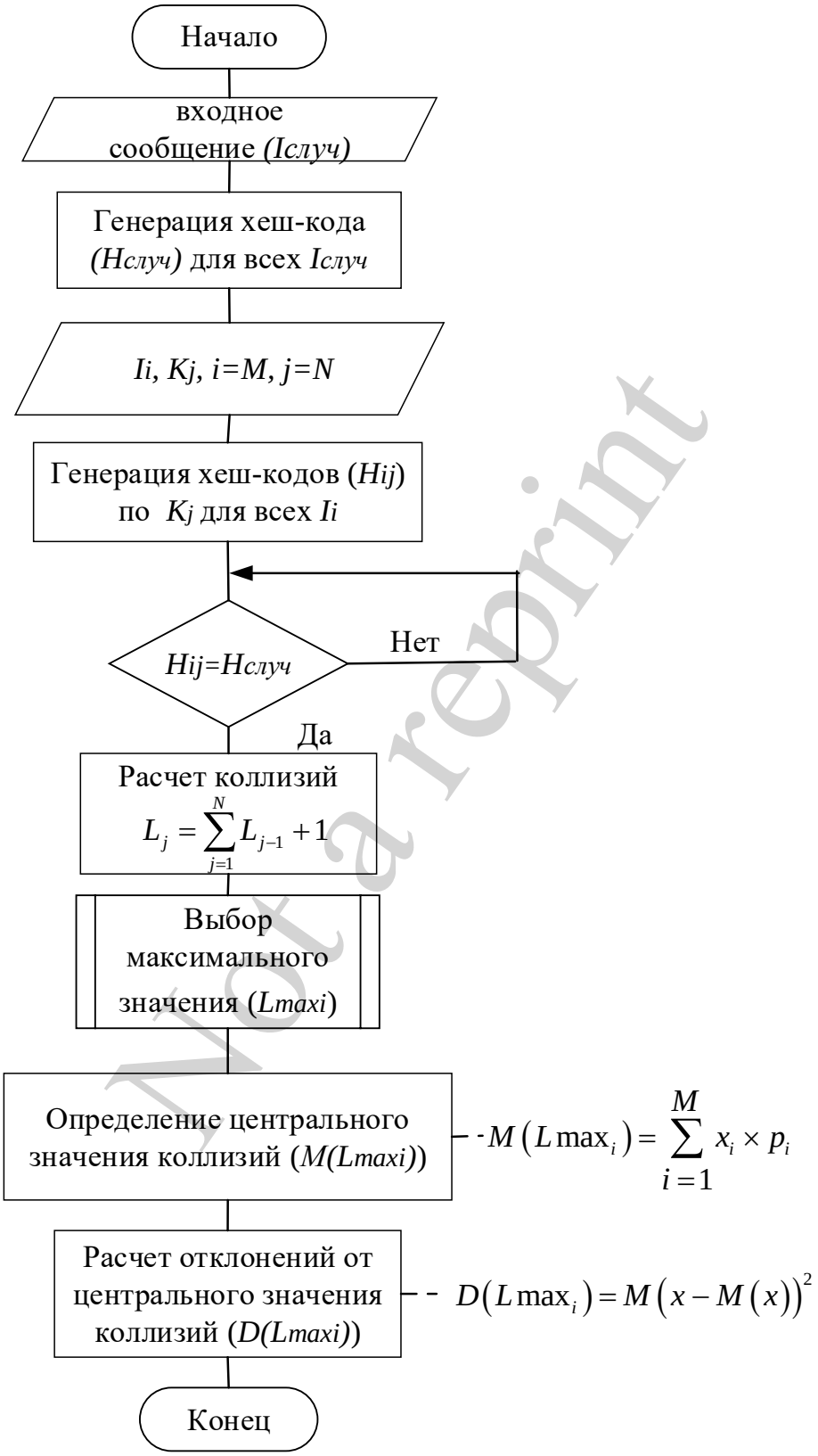


Рис. 9. Алгоритм проверки хеш-кодов на выполнение требований строго универсального класса хеш-функций по первому критерию

Реализация алгоритма можно описать следующими шагами его выполнения.

Шаг 1. Формируем одно случайное входное сообщение $I_{случ.}$

Шаг 2. Формируем хеш-код $H_{случ}$ случайного сообщения $I_{случ}$.

Шаг 3. Формируются входные сообщения

Шаг 4. Формируются ключи

Шаг 5. По каждому входному сообщению I_i с применением ключей K_j формируем их хеш-коды H_{ij}

Шаг 6. Проводим последовательное сравнение полученных хеш-кодов H_{ij} по одинаковому ключу K_j по всем входным сообщениям с хеш-кодом $H_{случ}$ случайного сообщения $I_{случ}$ на основании такого условия:

– если значения хешей совпадают (H_{ij}=$H_{случ}$), что свидетельствует о возникновении коллизии (L_j), то к счетчику коллизий прибавляется 1:

$$L_j = \sum_{j=1}^{N} L_{j-1} + 1.$$

Шаг 7. В рамках одного сообщения по всем ключам, выбираем максимальное значение коллизии L_{maxi}.

Шаг 8. Рассчитываем среднее арифметическое или центральное значение по максимумам количества коллизий путем нахождения их математического ожидания $M(L_{maxi})$:

$$M\left(L\max_i\right) = \sum_{i=1}^{M} x_i \times p_i.$$

Шаг 9. Рассчитываем значение разброса максимальных значений коллизий вокруг их центрального значения путем нахождения дисперсии ($D(L_{maxi})$) по максимумам количества коллизий:

$$D\left(L\max_i\right) = M\left(x - M\left(x\right)\right)^2.$$

Алгоритм проверки хеш-кодов на выполнение требований строго универсального класса хеш-функций по второму критерию приведен на рис. 10.

Реализация алгоритма можно описать следующими шагами его выполнения.

Шаг 1. Формируем два разных случайных входных сообщения $I_{случ1}$ и $I_{случ2.}$

Шаг 2. Формируем хеш-коды $H_{случ1}$ и $H_{случ2}$ для каждого из сообщений $I_{случ1}$ и $I_{случ2}$.

Шаг 3. Формируются входные сообщения.

Шаг 4. Формируются ключи.

Шаг 5. по каждому входному сообщению I_i с применением ключей K_j формируем их хеш-коды H_{ij}.

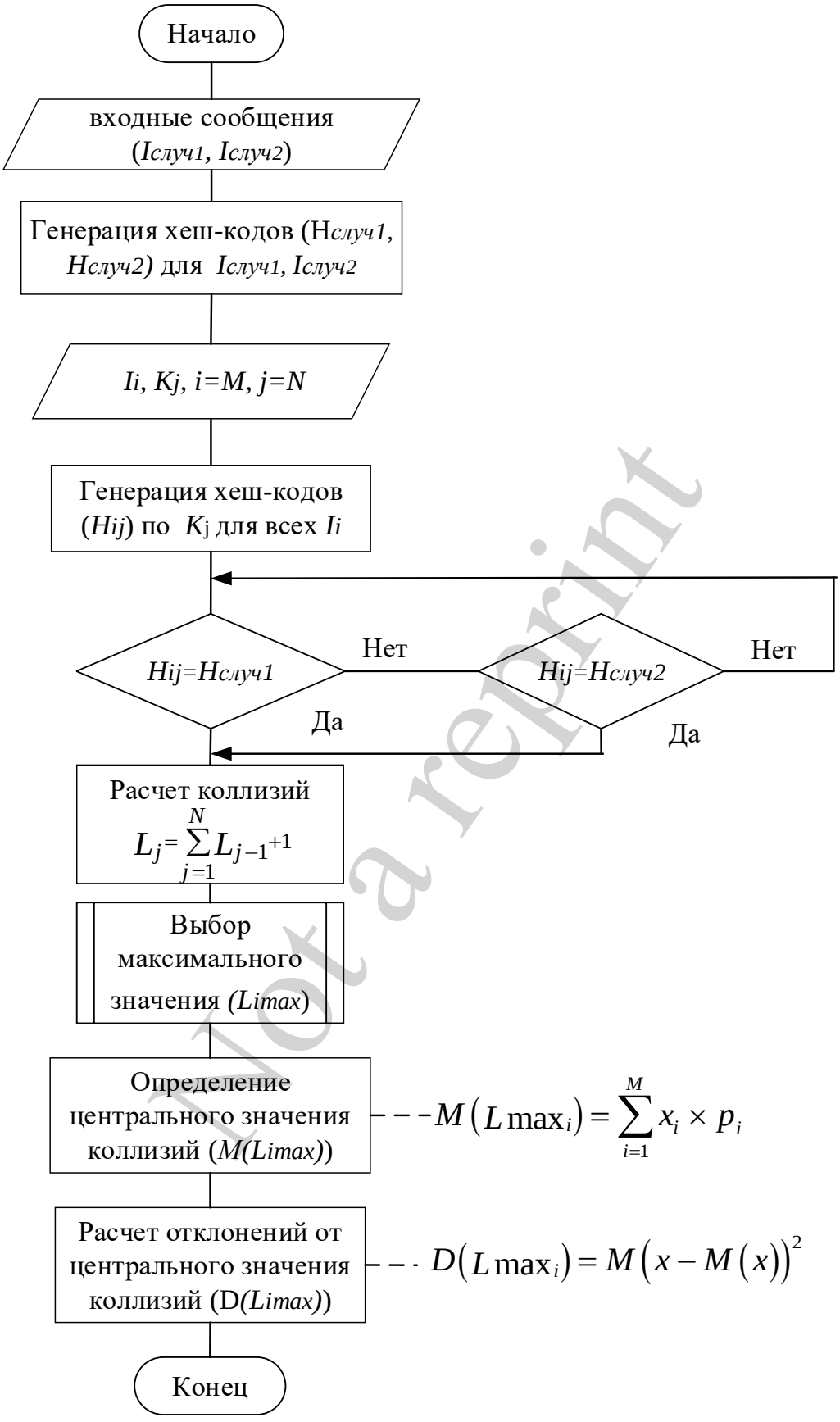


Рис. 10. Алгоритм проверки хеш-кодов на выполнение требований строго универсального класса хеш-функций по второму критерию

Шаг 6. Проводим последовательное сравнение полученных хеш-кодов $H_{случ1}$ и $H_{случ2}$ по одинаковому ключу K_j по всем входным сообщениям с хеш-кодами двух случайных сообщений:

– если значения хешей совпадают ($H_{ij}=H_{случ1}$ или $H_{ij}=H_{случ2}$), что свидетельствует о возникновении коллизии (L_j), то к счетчику коллизий прибавляется 1:

$$L_j = \sum_{j=1}^{N} L_{j-1} + 1.$$

Шаг 7. В рамках одного сообщения по всем ключам, выбираем максимальное значение коллизии L_{maxi}.

Шаг 8. Рассчитываем среднее арифметическое или центральное значение по максимумам количества коллизий путем нахождения их математического ожидания $M(L_{maxi})$*:*

$$M\left(L\max_i\right) = \sum_{i=1}^{M} x_i \times p_i.$$

Шаг 9. Рассчитываем значение разброса максимальных значений коллизий вокруг их центрального значения путем нахождения дисперсии ($D(L_{maxi})$) по максимумам количества коллизий:

$$D\left(L\max_i\right) = M\left(x - M\left(x\right)\right)^2.$$

Таким образом, предложенный алгоритм позволяет провести оценку не только выполнение критериев универсальности и строгой универсальности полученного МАС-кода, но и уровня стойкости.

9. Обсуждение результатов практической реализации модифицированного UMAC на ККК

Для обеспечения достоверности предлагаемого подхода рассмотрим практический пример реализации модифицированного каскадного алгоритма UMAC на ККК с ЕС, МЕС и DC.

Для моделирования реализации модифицированного алгоритма UMAC на алгеброгеометических (ЕС, МЕС) и ущербных кодах на основе ККК (ГККК) Мак-Элиса будет использовать следующие входные данные:

– длина блока данных (байт) – 32;

– длина секретного ключа – 32 байта;

– длина массива передаваемого открытого текста I – 3 байта;

– длина псевдослучайной ключевой последовательности (количество подключей) –1027;

– передаваемый открытый текст (*k*-разрядный информационный вектор над *GF*(*q*)) – 11;

– секретный вектор ошибок *e*=00000200 (сеансовый ключ);

– матрицы маскировки (личный ключ пользователя – KR_i):

– невырожденная $k \times k$ матрица – *X*;

– перестановочная матрица *P* размера $n \times n$;

– диагональная матрица *D*;

– порождающая матрица *G*;

Порождающая матрица (открытый ключ пользователя – KU_i) – G_X^{EC}, G_X^{MEC}, в зависимости от ККК (ГККК) Мак-Элиса.

Преобразования значений кодограммы по алгоритму *MV*2 приведены в табл. 3.

Таблица 3

Преобразования значений кодограммы по алгоритму *MV*2

Двоичное представление символа	Длина остатка	Остаток $C(x)$	Флаг $F(x)$
000	2	00	1
001	2	01	1
010	2	10	1
011	2	11	1
100	2	00	0
101	2	01	0
110	2	10	0
111	2	11	0

Сформируем хеш-код на основе модифицированного алгоритма UMAC с различными механизмами формирования псведоподложки на основе ККК с EC, MEC, DC.

I. Создание хеш-кода на основе алгоритма UMAC.

Шаг 1. Формирование первого слоя.

Значение хеш-функции первого уровня UHASH-hash Y_{L1I} рассчитаем по формуле:

$$Y_{L1I} = Hash_{L1}\left(K_{L1I}, I\right).$$

Для формирования K_{L1I} представим его как последовательность ключей из четырехбайтовых блоков:

$$K_{L1I} = K_{1I} \,||\, K_{2I} \,||\, ... \,||\, K_{nI},$$

где || – конкатенация (соединение) строк, соответствующих подключей.

Количество данных подключа:

$$n = \frac{1072}{32} = 33{,}5 \approx 33 \Rightarrow i = 1, 2, ..., 33.$$

Так как $T_i = \text{Index}\|i$, то для первого уровня *Index*=1, => T_i= $K_{i\text{I}}$:

$K_{1\text{I}}$=00000001 000000001
K_{2I}=00000001 000000010
K_{3I}=00000001 000000011
K_{4I}=00000001 000000100
K_{5I}=00000001 000000101
K_{6I}=00000001 000000110
K_{7I}=00000001 000000111
K_{8I}=00000001 000001000
K_{9I}=00000001 000001001
K_{10I}=00000001 00001010
K_{11I}=00000001 00001011
$K_{12I=}$00000001 00001100
$K_{13\text{I}}$=00000001 00001101
K_{14I}=00000001 00001110
$K_{15\text{I}=}$00000001 00001111
K_{16I}=00000001 00010000
K_{17I}=00000001 00010001
$K_{18I=}$00000001 00010010
K_{19I}=00000001 00010011
K_{20I}=00000001 00010100
K_{21I}=00000001 00010101
K_{22I}=00000001 00010110
K_{23I}=00000001 00010111
K_{24I}=00000001 00011000
K_{25I}=00000001 00011001
K_{26I}=00000001 00011010
K_{27I}=00000001 00011011
K_{28I}=00000001 00011100
K_{29I}=00000001 00011101
K_{30I}=00000001 00011110
K_{31I}=00000001 00011111
K_{32I}=00000001 00100000
K_{33I}=00000001 00100001

На основании длины M входного сообщения количество блоков равно T=1, поэтому количество подключей на этом уровне одинаково, следовательно $K_{L1I} = T_1 = 0000000100000001$.

Значения хеш-функции этого слоя рассчитываются по следующей формуле:

$$Y_{L1I} = (I + K_{L1I}) \bmod 32 == \left(0100110 + 10000001\right) \bmod 32 = 111.$$

Шаг 2. Формирование второго слоя.

Поскольку длина *M* меньше 1024 байта, этот уровень хеширования не будет выполняться, и мы будем выполнять вычисления с использованием хеш-кода третьего уровня.

Шаг 3. Формирование третьего слоя.

Количество подключей для K_{L31}:

$$n = \frac{64 \times 4}{32} = 8 \Rightarrow i = 1, 2, 3, 4, 5, 6, 7, 8.$$

Поэтому для формирования K_{L31I} представьте его как последовательность ключей из восьми четырехбайтовых блоков:

$$K_{L31I} = K_{1I} \| K_{2I} \| K_{3I} \| K_{4I} \| K_{5I} \| K_{6I} \| K_{7I} \| K_{8I}.$$

Для третьего уровня *Index* =3, => T_i=K_{iI}:

K_{1I}=00000011 00000001	K_{5I}=00000011 00000101
K_{2I}=00000011 00000010	K_{6I}=00000011 00000110
K_{3I}=00000011 00000011	K_{7I}=00000011 00000111
K_{4I}=00000011 00000100	K_{8I}=00000011 00001000

Количество подключей для K_{L32I}:

$$n = \frac{4 \times 4}{32} = 0,5 \approx 1 => i = 1.$$

Для формирования K_{L32I} представьте его как последовательность ключей из 1 четырехбайтового подблока:

$$K_{L32I} = K_{1I}.$$

Для третьего уровня *Index*=4, => $T_i = K_{1I}$:

K_{1I}=00000100 00000001.

Значение хеш-функции третьего слоя рассчитывается по следующей формуле:

$$Y_{L3I} = \left(\left(Y_{L1I} \bmod\left(2^{36} - 5\right)\right) \bmod 2^{32}\right) \text{xor} Y_{L32I} =$$

$$((I + K_{1I}) \bmod 32) \bmod(2^{36} - 5)) \bmod 2^{32}) \text{xor} Y_{L32I}.$$

$$Y_{L3I} = \left(\left(11 \bmod\left(2^{36} - 5\right)\right) \bmod 2^{32}\right) \text{xor} 00000100\ 00000001 = 10000000010.$$

II. Формирование ключевых данных модифицированных крипто-кодовых конструкций Мак-Элиса с EC, MEC, DC [7, 8, 19, 37].

Шаг 1. Открытый ключ для ККК Мак-Элиса на EC, формируется за выражением:

$$G_X^{EC} = X \times G \times P \times D.$$

Открытый ключ для ККК Мак-Элиса на MEC, формируется за выражением:

$$G_X^{MEC} = X \times G^{EC} \times P \times D.$$

Открытый ключ для ГККК Мак-Элиса на DC, формируется за выражением:

$$G_X^{HMEC} = X \times G^{MEC} \times P \times D.$$

Результатом формирование открытого ключа является порождающая матрица алгеброгеометрического кода:

$$G_X^{EC} = G_X^{MEC} = G_X^{HMEC} = \begin{bmatrix} 2 & 1 & 3 & 0 & 1 & 1 & 1 & 0 \\ 0 & 2 & 2 & 2 & 2 & 0 & 3 & 2 \end{bmatrix}.$$

Шаг 2. Криптограмма (кодограмма), сформированная из информационного сообщения *I*, представляет собой вектор длины *n*, который рассчитывается по следующей формуле:

– для ККК Мак-Элиса на ЕС:

$$C_X^* = I \times G_X^{EC} \oplus e.$$

– для ККК Мак-Элиса на МЕС:

$$C_X^* = I \times G_X^{MEC} \oplus e.$$

– для ГККК Мак-Элиса на DC:

$$C_X^* = I \times G_X^{HMEC} \oplus e.$$

Результатом сформированного вектора является следующее его значение: $C_X^* = 23023322$.

Шаг 3. Формирование вектора инициализации IV=00100000, который определяет местоположение символов сокращения/удлинения кодовой последовательности:

– для укороченных МЕС – $C_x^* = 2323322$;

– для удлиненных МЕС – $C_x^* = 23123322$.

Шаг 4. Формирование ущерба и ущербного кода, через нанесение ущерба на основе алгоритма *MV*2 кодового слова:

– исходный текст (слово):

– для укорочения – $C_x^* = 2323322_{10} = 010011\ 000010011011\ 010\ 010_2$.

Шаг 5. Передача ущерба (флага) одним из каналов получателю и отправку ущербного кода (остатка) другим каналом получателю:

– для укороченных МЕС:

– отправка ущерба в канал – $C_x^* = 010011\ 000010011011\ 010\ 010_2 = 45818_{10}$;

– отправка ущербного кода в канал – $F(x)=11111111_2=255_{10}$;

– для удлиненных МЕС:

– отправка ущерба в канал – $C_x^* = 010011\ 000010011011\ 010\ 010_2 = 2495337_{10}$;

– отправка ущербного кода в канал – $F(x)=111111111_2=511_{10}$.

III. Формирование псевдослучайной подкладки проводится на основе предложенных модифицированных крипто-кодовых конструкций Мак-Элиса с ЕС, МЕС, DC:

$$Pad = PDF(K, Nonce, Taglen) = PDF(0106, 8, 4) = 1101010.$$

IV. Генерация хеш-кода:

$$Tag = UMAC\big(K, I, Nonce, Taglen\big) = Hash(K, I, Taglen) \oplus$$
$$\oplus PDF(K, Nonce, Taglen) = Y_{L3M} \oplus Pad = 10000000010 \oplus 1101010 = 10001101100.$$

$$Y = Y_{L3M} \oplus Tag.$$

$$Y = 10000000010 \oplus 10001101100 = 1101110 = 110_{10}.$$

VI. Верификация на основе DSA (Digital Signature Algorithm).

При использовании модифицированного алгоритма UMAC с ККК (ГККК) Мак-Элиса в стандарте DSA отправитель использует личный ключ) KR_i, получатель использует открытый ключ KU_i для верификации цифровой подписи.

Предложенный подход позволяет обеспечить требуемый уровень оперативности формирования хеш-кода (в режиме онлайн) с учетом роста вычислительных ресурсов и объемов, передаваемых данных. В табл. 4 приведены результаты исследований энергетических затрат на практическую реализацию предлагаемых алгоритмов формирования подкладки на основе использования ККК (ГККК) Мак-Элиса на алгеброгеометрических (ЕС, МЕС) и ущербных кодах.

Таблица 4

Зависимость программной реализации ККК (ГККК) Мак-Элиса от мощности поля (количество групповых операций)

Криптосистемы	2^5	2^6	2^7	2^8	2^9	2^{10}
ККК на ЕС	10018042	18048068	32847145	47489784	63215578	**82467897**
ККК на укороченных МЕС	10007947	**17787431**	28595014	44079433	61974253	79554764
ККК на удлиненных МЕС	11156138	**18561228**	33210708	48297112	65171690	84051337
ГККК на укороченных МЕС	**7900315**	14892945	25565274	42279183	58963778	76564173
ГККК на удлиненных МЕС	**7905257**	14682411	25595014	42116327	58468143	75474764

Анализ табл. 4 подтверждает теоретические расчеты снижения затрат на практическую реализацию ККК (ГККК) Мак-Элиса на ЕС (МЕС), DC с сохранением требуемых показателей уровня стойкости хеш-кода в модифицированном алгоритме UMAC. В табл. 5 представлены результаты исследований статистических свойств предложенных методов на основе пакета *NIST STS* 822, которые подтверждают стойкость предложенных ККК (ГККК) Мак-Элиса на ЕС (МЕС), DC [55].

Таблица 5
Результаты исследования статистической безопасности

Крипто-системы	Количество тестов, в которых тестирование прошли более 99 % последовательностей	Количество тестов, в которых тестирование прошли более 96 % последовательностей	Количество тестов, в которых тестирование прошли менее 96 % последовательностей
ККК *McEliece EC*	149 (78,83 %)	189 (100 %)	0 (0 %)
МККК *McEliece* на укороченных *MEC*	151 (79,89 %)	189 (100 %)	0 (0 %)
МККК *McEliece* на удлиненных *MEC*	152 (80,42 %)	189 (100 %)	0 (0 %)
ГККК удлиненных *MEC*	153 (80,95 %)	189 (100 %)	0 (0 %)
ГККК укороченных *MEC*	155 (82 %)	189 (100 %)	0 (0 %)

Таким образом, представленные результаты подтверждают возможность реализации данного подхода к формированию модифицированного каскадного алгоритма хеширования с использованием крипто-кодовых конструкций в качестве механизма обеспечения требуемого уровня стойкости в условиях постквантового периода. При этом использование той или иной конструкции зависит от вычислительной возможности и/или платформы на основе которой разработано ПО.

Дальнейшим направлением исследования является практическое подтверждение статистических оценок свойств универсальности и строгой универсальности модифицированного алгоритма UMAC с формированием подкладки на ККК (ГККК) Мак-Элиса.

10. Выводы

1. В условиях возрастания вычислительных ресурсов, расширения сферы цифровой экономики и услуг электронного банкинга одним из условий обеспечения услуг безопасности является поиск новых и/или модификация известных методов. Рост и комплексирование современных угроз, их гибридность и синергизм требуют введения жестких критериев к специальным механизмам обеспечения аутентичности. Среди известных алгоритмов формирования MAC-кодов особое место занимают универсальные хеш-функции. Однако их применение без дополнительного шифрования хеш-кода не обеспечивает требуемый уровень стойкости.

2. Анализ формирования хеш-кодов на основе каскадного применения универсальных хеш-функций показал, что использование в качестве механизма псевдослучайной подкладки в алгоритме UMAC блочного симметричного алгоритма AES не позволяет “сохранить” универсальность хеш-кода. А усовершенствование механизма на основе модулярной арифметики (алгоритм MASH–2) не удовлетворяет требованиям по оперативности преобразований. Таким образом, предлагается использовать одно из перспективных направлений – крипто-кодовые конструкции на основе алгеброгеометрических и ущербных кодов.

3. Предложены модификации построения каскадного алгоритма хеширования на основе использования ККК на EC, MEC, DC. Такой подход позволяет “сохранить” универсальность и все преимущества данного класса хеш-функций, а также обеспечить требуемые параметры по оперативности и стойкости в условиях постквантовой криптографии и появления полномасштабного квантового компьютера.

4. Предложенные алгоритмы оценки стойкости хеш-кодов модифицированного алгоритма UMAC на крипто-кодовых конструкциях Мак-Элиса с EC, MEC, DC позволяют не только оценить выполнение критериев универсальности и строгой универсальности, но и обеспечивают оценки стойкости хеш-кода к современным угрозам. В основе предлагаемых алгоритмов статистического исследования коллизионных свойств, формируемых MAC-кодов лежит эмпирическая оценка максимумов числа ключей (правил хеширования).

Литература

1. Евсеев, С. П., Коц, Г. П., Король, О. Г. (2015). Анализ законодательной базы к системе управления информационной безопасностью НСМЭП. Восточно-Европейский журнал передовых технологий, 5 (3 (77)), 48–59. doi: https://doi.org/10.15587/1729-4061.2015.51468

2. Евсеев, С. П., Абдуллаев, В. Г. (2015). Алгоритм мониторингу метода двухфакторной аутентификации на основе системы Passwindow. Восточно-Европейский журнал передовых технологий, 2 (2 (74)), 9–16. doi: https://doi.org/10.15587/1729-4061.2015.38779

3. Актуальные киберугрозы – 2017: тренды и прогнозы (2018). Positive technologies. URL: https://www.ptsecurity.com/ru-ru/research/analytics/cybersecurity-threatscape-2017/

4. Актуальные киберугрозы – 2018. Тренды и прогнозы (2019). Positive technologies. URL: https://www.ptsecurity.com/ru-ru/research/analytics/cybersecurity-threatscape-2018/

5. Актуальные киберугрозы: итоги 2019 года (2020). Positive technologies. URL: https://www.ptsecurity.com/ru-ru/research/analytics/cybersecurity-threatscape-2019/

6. Yevseiev, S., Hryhorii, K., Liekariev, Y. (2016). Developing of multi-factor authentication method based on niederreiter-mceliece modified crypto-code system. Eastern-European Journal of Enterprise Technologies, 6 (4 (84)), 11–23. doi: https://doi.org/10.15587/1729-4061.2016.86175

7. Yevseiev, S., Korol, O., Kots, H. (2017). Construction of hybrid security systems based on the crypto-code structures and flawed codes. Eastern-European Journal of Enterprise Technologies, 4 (9 (88)), 4–21. doi: https://doi.org/10.15587/1729-4061.2017.108461

8. Yevseiev, S., Tsyhanenko, O., Ivanchenko, S., Aleksiyev, V., Verheles, D., Volkov, S. et. al. (2018). Practical implementation of the Niederreiter modified crypto-code system on truncated elliptic codes. Eastern-European Journal of Enterprise Technologies, 6 (4 (96)), 24–31. doi: https://doi.org/10.15587/1729-4061.2018.150903

9. Сидельников, В. М. (2002). Криптография и теория кодирования. Материалы конференции “Московский университет и развитие криптографии в России”.

10. Bartock, M., Cichonski, J., Souppaya, M., Smith, M., Witte, G., Scarfone, K. (2016). Guide for cybersecurity event recovery. NIST. doi: https://doi.org/10.6028/nist.sp.800-184

11. Security requirements for cryptographic modules. URL: https://csrc.nist.gov/csrc/media/publications/fips/140/2/final/documents/fips1402.pdf

12. Cichonski, J., Franklin, J. M., Bartock, M. (2017). Guide to LTE security. NIST. doi: https://doi.org/10.6028/nist.sp.800-187

13. Lohachab, A., Lohachab, A., Jangra, A. (2020). A comprehensive survey of prominent cryptographic aspects for securing communication in post-quantum IoT networks. Internet of Things, 9, 100174. doi: https://doi.org/10.1016/j.iot.2020.100174

14. Petrenko, K., Mashatan, A., Shirazi, F. (2019). Assessing the quantum-resistant cryptographic agility of routing and switching IT network infrastructure in a large-size financial organization. Journal of Information Security and Applications, 46, 151–163. doi: https://doi.org/10.1016/j.jisa.2019.03.007

15. Hryshchuk, R., Yevseiev, S., Shmatko, A. (2018). Construction methodology of information security system of banking information in automated banking systems. Vienna: Premier Publishing s. r. o., 284. doi: https://doi.org/10.29013/r.hryshchuk_s.yevseiev_a.shmatko.cmissbiabs.284.2018

16. Горбенко, Ю. І., Ганзя, Р. С. (2014). Аналіз можливостей квантових комп’ютерів та квантових обчислень для криптоаналізу сучасних криптосистем. Східно-Європейський журнал передових технологій, 1 (9 (67)), 8–16. doi: https://doi.org/10.15587/1729-4061.2014.19897

17. Король, О., Пархуць, Л., Евсеев, С. (2013). Метод каскадного формирования мас-кодов с использованием модулярных преобразований. Научные ведомости Белгородского государственного университета. Серия: Экономика. Информатика, 15 (158), 147–157.

18. Король, О. Г., Кузнецов, А. А., Евсеев, С. П. (2012). Исследование коллизионных свойств кодов аутентификации сообщений UMAC. Прикладная радиоэлектроника, 11 (2), 171–183.

19. Євсеєв, С. П., Йохов, О. Й., Король, О. Г. (2013). Гешування даних в інформаційних системах. Харків: Вид. ХНЕУ, 312.

20. Кузнецов, О. О., Горбенко, Ю. І., Кіян, А. С., Уварова, А. О., Кузнецова, Т. Ю. (2018). Порівняльні дослідження та аналіз ефективності гібридної

кодової криптосистеми. Радиотехника, 195, 61–69. URL: http://nbuv.gov.ua/UJRN/rvmnts_2018_195_9

21. Marquez-Corbella, I., Tillich, J.-P. (2016). Using Reed-Solomon codes in the (U | U + V) construction and an application to cryptography. 2016 IEEE International Symposium on Information Theory (ISIT). doi: https://doi.org/10.1109/isit.2016.7541435

22. Kapshikar, U., Mahalanobis, A. (2018). A Quantum-Secure Niederreiter Cryptosystem using Quasi-Cyclic Codes. Proceedings of the 15th International Joint Conference on e-Business and Telecommunications. doi: https://doi.org/10.5220/0006843005060513

23. Abidin, A. (2012). On Security of Universal Hash Function Based Multiple Authentication. Lecture Notes in Computer Science, 303–310. doi: https://doi.org/10.1007/978-3-642-34129-8_27

24. Handschuh, H., Preneel, B. (2008). Key-Recovery Attacks on Universal Hash Function Based MAC Algorithms. Advances in Cryptology – CRYPTO 2008, 144–161. doi: https://doi.org/10.1007/978-3-540-85174-5_9

25. Abouhogail, R. A. (2011). New multicast authentication protocol for entrusted members using advanced encryption standard. The Egyptian Journal of Remote Sensing and Space Science, 14 (2), 121–128. doi: https://doi.org/10.1016/j.ejrs.2011.11.003

26. Carter, J. L., Wegman, M. N. (1979). Universal classes of hash functions. Journal of Computer and System Sciences, 18 (2), 143–154. doi: https://doi.org/10.1016/0022-0000(79)90044-8

27. Stinson, D. R. (1994). Combinatorial techniques for universal hashing. Journal of Computer and System Sciences, 48 (2), 337–346. doi: https://doi.org/10.1016/s0022-0000(05)80007-8

28. Sarvate, D. G. Seberry, J. (1986) Encryption methods based on combinatorial designs. URL: https://ro.uow.edu.au/cgi/viewcontent.cgi?article=2034&context=infopapers

29. Халимов, Г. З. (2013). Строго универсальное хеширование. Прикладная радиоэлектроника, 12 (2), 220–224.

30. Simmons, G. J. (1988). An Impersonation-Proof Identity Verification Scheme. Lecture Notes in Computer Science, 211–215. doi: https://doi.org/10.1007/3-540-48184-2_17

31. Simmons, G. J. (1985). Authentication Theory/Coding Theory. Lecture Notes in Computer Science, 411–431. doi: https://doi.org/10.1007/3-540-39568-7_32

32. Кузнецов, А. А., Король, О. Г., Босько, В. В. (2011). Модель формирования кодов аутентификации сообщений с использованием универсальных хеширующих функций. Системи обробки інформації, 3 (93), 117–125.

33. Алексеев, М. О. (2014). Защита от алгебраических манипуляций на основе операции скалярного умножения. Проблемы информационной безопасности. Компьютерные системы, 2, 47–53.

34. Алексеев, М. О., Мирончиков, Е. Т. (2011). Об обнаружении ошибок с помощью нелинейных кодов. Научная сессия ГУАП, 1, 40–43.

35. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P. (1999). UMAC: Fast and Secure Message Authentication. Lecture Notes in Computer Science, 216–233. doi: https://doi.org/10.1007/3-540-48405-1_14

36. Фергюсон, Н., Шнайдер, Б. (2004). Практическая криптография. М.: Издательский дом “Вильямс”, 432.

37. Кузнецов, А. А., Пушкарев, А. И., Сватовский, И. И., Шевцов, А. В. (2016). Несимметричные криптосистемы на алгебраических кодах для постквантового периода. Радиотехника, 186, 70–90.

38. Krovetz, T., Rogaway, P. (2001). Fast Universal Hashing with Small Keys and No Preprocessing: The PolyR Construction. Information Security and Cryptology – ICISC 2000, 73–89. doi: https://doi.org/10.1007/3-540-45247-8_7

39. Krovetz, T. (2000). Software-Optimized Universal Hashing and Message Authentication. University of California Davis, 269.

40. Krovetz, T. (Ed.). (2006). UMAC: Message Authentication Code using Universal Hashing. doi: https://doi.org/10.17487/rfc4418

41. Король, О. Г. (2015). Оценка вычислительной сложности некоторых функций хеширования. Системи обробки інформації, 4, 105–110.

42. Krovetz, T., Black, J., Halevi, S., Hevia, A., Krawczyk, H., Rogaway, P. (2000). UMAC. Primitive submitted to NESSIE, 157–160.

43. Bosselaers, A., Govaerts, R., Vandewalle, J. (1996). Fast Hashing on the Pentium. Lecture Notes in Computer Science, 298–312. doi: https://doi.org/10.1007/3-540-68697-5_23

44. Final report of European project number IST-1999-12324, named New European Schemes for Signatures, Integrity and Encryption. Version 0.15 (beta). Springer-Verlag.

45. Евсеев, С. П., Король, О. Г., Огурцов, В. В. (2014). Усовершенствованный алгоритм UMAC на основе модулярных преобразований. Восточно-Европейский журнал передовых технологий, 1 (9 (67)), 16–23. doi: https://doi.org/10.15587/1729-4061.2014.20130

46. Yevseiev, S., Kots, H., Minukhin, S., Korol, O., Kholodkova, A. (2017). The development of the method of multifactor authentication based on hybrid crypto-code constructions on defective codes. Eastern-European Journal of Enterprise Technologies, 5 (9 (89)), 19–35. doi: https://doi.org/10.15587/1729-4061.2017.109879

47. Евсеев, С. П. (2017). Использование ущербных кодов в криптокодовых системах. Системи обробки інформації, 5, 109–121. URL: http://nbuv.gov.ua/UJRN/soi_2017_5_17

48. Havrylova, A., Korol, O., Milevskyi, S. (2019). Mathematical model of authentication of a transmitted message based on a mceliece scheme on shorted and extended modified elliptic codes using UMAC modified algorithm. Cybersecurity: Education, Science, Technique, 5, 40–51. doi: https://doi.org/10.28925/2663-4023.2019.5.4051

49. Yevseiev, S., Havrylova, A. (2020). Improved umac algorithm with crypto-code mceliece’s scheme. Modern problems of computer science and IT-

education. Vienna, 79–92. doi: https://doi.org/10.29013/melnikk.shmatkoo.mpcsie.2020.352

50. Korol, O., Havrylova, A., Yevseiev, S. (2019). Practical UMAC algorithms based on crypto code designs. Przetwarzanie, transmisja I bezpieczenstwo informacji. Vol. 2. Bielsko-Biala: Wydawnictwo naukowe Akademii Techniczno-Humanistycznej w Bielsku-Bialej, 221–232.

51. Yevseiev, S., Rzayev, K., Korol, O., Imanova, Z. (2016). Development of mceliece modified asymmetric crypto-code system on elliptic truncated codes. Eastern-European Journal of Enterprise Technologies, 4 (9 (82)), 18–26. doi: https://doi.org/10.15587/1729-4061.2016.75250

52. Мищенко, В. А., Виланский, Ю. В. (2007). Ущербные тексты и многоканальная криптография. Минск: Энциклопедикс, 292.

53. Мищенко, В. А., Виланский, Ю. В., Лепин, В. В. (2007). Криптографический алгоритм MV 2. Минск: Энциклопедикс, 176.

54. Король, О. Г. (2010). Исследование коллизионных свойств кодов аутентификации сообщений UMAC. Системи обробки інформації. Проблеми і перспективи розвитку IT-індустрії, 7 (88), 221.

55. Rukhin, A., Sota, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S. et. al. (2000). A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST. doi: https://doi.org/10.6028/nist.sp.800-22