КОНТРОЛЬ АУТЕНТИЧНОСТИ СОДЕРЖИМОГО ЦИФРОВЫХ ИЗОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ ЦИФРОВЫХ ВОДЯНЫХ ЗНАКОВ И ПЕРЦЕПТИВНЫХ ХЕШ

Под аутентичностью данных понимается их свойство быть подлинными. Подлинность означает, что они были созданы законными участниками информационного процесса и не подвергались случайной или преднамеренной модификации. На сегодняшний день для контроля аутентичности (также как и для контроля целостности) электронных данных чаще всего применяются:

- циклические избыточные коды;
- криптографические хеш-функции;
- электронные подписи и коды аутентификации сообщения.

Кратко рассмотрим перечисленные методы.

Циклические избыточные коды (CRC, Cyclic Redundancy Check) – это семейство алгоритмов нахождения контрольных сумм, предназначенных для проверки целостности данных. Контрольная сумма представляет собой некоторое значение, вычисленное по определенной схеме на основе кодируемого сообщения. Алгоритмы CRC базируются на свойствах деления с остатком двоичных многочленов, то есть многочленов над конечным полем $GF(2)$. Конечная входная последовательность битов $b_0, b_1, \ldots b_{N-1}$, описывающая сообщение, для которого вычисляется CRC, может быть взаимно однозначно представлена в виде двоичного полинома $P(x) = \sum_{i=0}^{N-1} b_i x^i$. Так, двоичной последовательности 11000110011 соответствует многочлен $P(x) = x^{10} + x^9 + x^5 + x^4 + x + 1$. Количество различных многочленов степени, меньшей N, равно 2^N, что совпадает с числом всех двоичных последовательностей длины N. Значение CRC является остатком от деления многочлена, соответствующего входным данным, на некий фиксированный порождающий многочлен

$$R(x) = P(x)x^N \bmod G(x),$$

где многочлен $R(x)$ – результат вычисления контрольной суммы, представляющий битовую последовательность длины N, $P(x)$ – многочлен, представляющий входной поток бит, для которого вычисляется CRC, $G(x)$ – порождающий многочлен степени N, определяемый как битовая последовательность длины N (чаще всего $G(x)$ является неприводимым

многочленом), умножение x^N осуществляется приписыванием N нулевых битов к входной последовательности, что улучшает качество хеширования для коротких входных последовательностей. В известных, применяемых на практике генераторах CRC, используются стандартизированные образующие многочлены, обладающие хорошими математическими и корреляционными свойствами (минимальное число коллизий, простота вычисления).

Криптографические хеш-функции. По аналогии с CRC функции данного класса реализуют отображение, на вход которого подается сообщение переменной длины M, а на выходе формируется битовая строка фиксированной длины $h = H(M)$. Основное отличие заключается в длине генерируемого хеш и сложности внутренних алгоритмов его вычисления. Хорошая хеш-функция равномерно и случайно отображает множество всех возможных входных сообщений во множество результирующих хешей. Криптографическая хеш-функция должна обладать следующими свойствами:

1) она может быть применена к аргументу любого размера;

2) выходное значение H имеет фиксированный размер;

3) простота вычисления $H(M)$ для любого M;

4) однонаправленность, устойчивость к нахождению прообраза (для любого h вычислительно сложно найти такое M, что $H(M) = h$);

5) устойчивость к нахождению второго прообраза (для любого фиксированного M_1 вычислительно сложно найти такое $M_2 \neq M_1$, что $H(M_1) = H(M_2)$);

6) устойчивость к нахождению коллизий (вычислительно сложно найти два различных сообщения M_1 и M_2 такие, что $H(M_1) = H(M_2)$);

7) чувствительность к всевозможным изменениям в сообщении M, таким как вставки, выбросы, перестановки и т.п.

Реализация многих криптографических хеш-функций основана на использовании итеративной последовательной схемы – структуры Меркла-Дамгарда (рис. 1). На первом шаге работы данной схемы происходит выравнивание ($padding$) сообщения путем дополнения его нулями до кратности n – размера блока разбиения сообщения, подаваемого на функцию сжатия G. В конец сообщения также дописывается его длина $len(M)$. Полученное сообщение разбивается на блоки $M_1, M_2, \ldots M_r$ по n бит каждый. Далее реализуется последовательность вызовов функции

сжатия G, преобразующей два входных блока длины n и k бит в выходной блок длины k бит. В качестве G чаще всего используются блочные шифры или специально созданные функции сжатия. Алгоритм начинает вычисления с фиксированного значения (зависящего от конкретной реализации) – вектора инициализации IV. Для каждого последующего блока сообщения, функция сжатия вместо IV принимает результат предыдущего раунда сжатия. Итоговым значением функции хеширования являются выходные k бит последней итерации g_r. Для «упрочнения» хеша, g_r дополнительно может пропускаться через функцию финализации, которая может использоваться для уменьшения размера выходного хеша ($m < k$), или чтобы гарантировать лучшее смешивание битов и усилить влияние небольшого изменения входного сообщения на хеш.

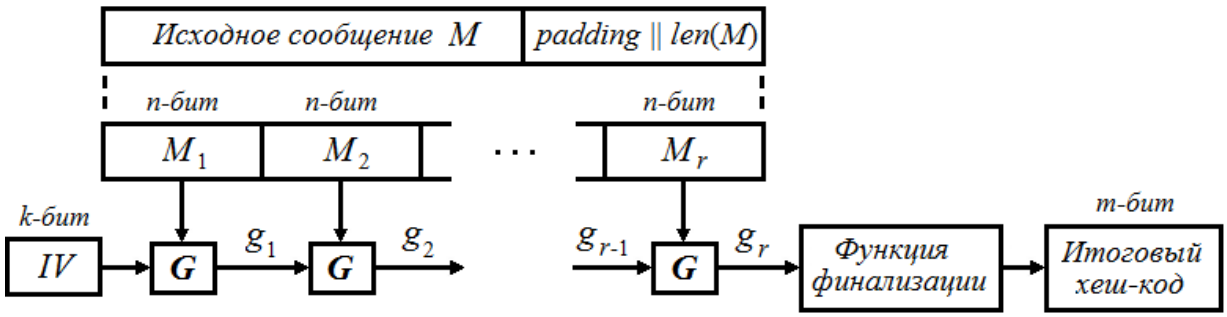

Рис. 1. Схема Меркла-Дамгарда формирования криптографического хеш-кода

Термин хеш-функция также используется для обозначения функции отображения при доступе к хеш-таблицам, при поиске дубликатов в сериях наборов данных, а также при поиске схожих по содержимому изображений. У таких функций много свойств, делающих их схожими с криптографическими хеш-функциями, но, тем не менее, это разные вещи. На практике криптографические функции хеширования применяются для обнаружения модификации сообщения – в качестве кодов обнаружения изменений (MDC, Manipulation Detection Code) или проверки целостности сообщения (MIC, Message Integrity Check), в качестве криптографических генераторов псевдослучайных чисел для создания нескольких ключей на основе одного секретного ключа, а также в составе алгоритмов вычисления электронных подписей для сокращения размера подписываемых данных.

Электронная подпись (ЭП) – это реквизит электронного документа, полученный в результате его криптографического преобразования с использованием закрытого ключа подписи и позволяющий проверить

отсутствие искажения информации в электронном документе с момента формирования подписи. На практике ЭП используются как эффективный механизм контроля целостности и защиты электронных документов от изменений и подделки. Также с их помощью фиксируется невозможность отказа от авторства подписанного документа, т.к. создать правильную ЭП возможно лишь в случае обладания закрытым ключом подписи, который по определению должен быть известен только владельцу - автору документа. Перечисленные свойства ЭП обуславливают их широкое распространение и применение в различных приложениях электронной экономики, электронного документального и денежного обращения.

Упрощенная схема формирования и проверки асимметричной ЭП приведена на рис. 2.

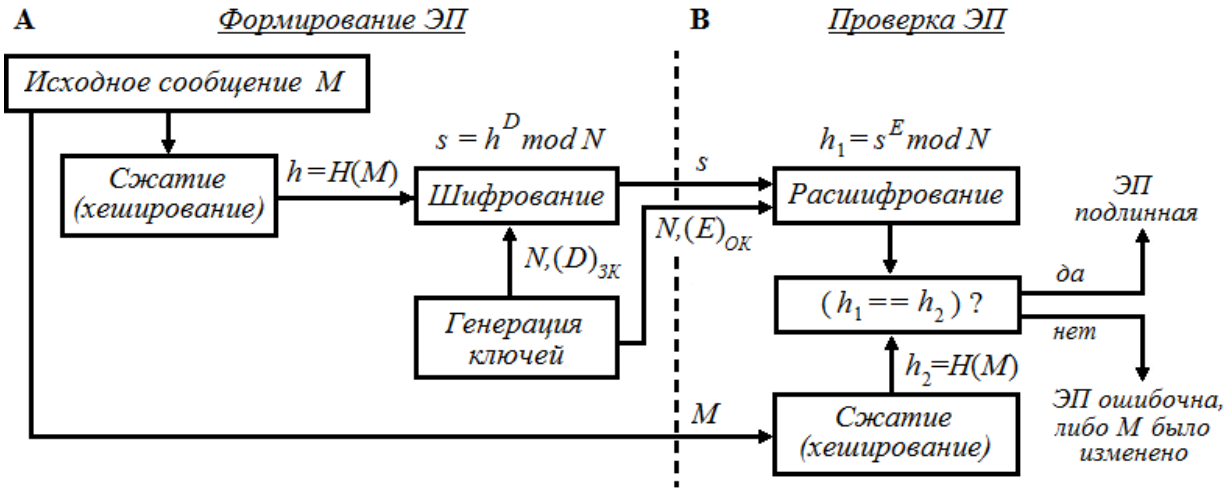

Рис. 2. Схема формирования и проверки электронной подписи

Перед формированием подписи сторона **A** (отправитель сообщения) генерирует пару ключей: закрытый (секретный) и открытый ключ. В данном примере генерация ключей, шифрование и расшифрование данных реализуется с использованием алгоритма RSA (Rivest, Shamir, Adleman). Используемый при проверке ЭП открытый ключ рассылается или делается доступным, например, на разделяемом ресурсе, для остальных абонентов сети. Для формирования цифровой подписи сторона **A** вычисляет значение хеш-функции $h = H(M)$ подписываемого сообщения M. Далее **A** шифрует хеш-код h своим закрытым ключом (D) $s = h^D \bmod N$, в результате чего получается цифровая подпись s. Сообщение M вместе с s отправляется стороне **B** (получателю).

При верификации подписи **B** расшифровывает s открытым ключом отправителя $h_1 = s^E \bmod N$ и вычисляет хеш-код принятого сообщения $h_2 = H(M)$. Если полученные значения хешей равны $h_1 = h_2$, то ЭП является подлинной. В противном случае имеет место изменение сообщения либо ЭП.

Код аутентификации сообщения (MAC, Message Authentication Code) или имитовставка. По аналогии с ЭП MAC – это реквизит электронного документа, полученный в результате его криптографического преобразования, предназначенный для обеспечения целостности документа и аутентификации источника данных. В отличие от ЭП в MAC применяются секретные ключи совместного использования, как при формировании, так и при проверке кода аутентификации.

Среди наиболее распространенных способов создания MAC можно отметить:

- CBC-MAC, предполагающий использование блочных симметричных алгоритмов шифрования в режимах CBC «сцепления блоков шифротекста» или CFB «обратной связи по шифротексту» с выбором в качестве имитовставки последнего шифрованного блока сообщения. В режиме CBC код аутентификации вычисляется в виде

$$MAC = E_{K_S}\left(M_N \oplus E_{K_S}\left(M_{N-1} \oplus \ldots E_{K_S}\left(M_2 \oplus E_{K_S}\left(M_1 \oplus IV\right)\right)\right)\right),$$

где E – блочный алгоритм шифрования, K_S – секретный ключ, IV – вектор инициализации, M_i, $i = \overline{1, N}$ – блоки сообщения $M = M_1 \| \ldots \| M_N$;

- шифрование симметричным алгоритмом результата вычисления хеш-функции от исходного сообщения $MAC = E_{K_S}(H(M))$, где K_S – секретный ключ.

Одна из проблем использования MAC заключается в необходимости знания секретных ключей всеми взаимодействующими сторонами, что потенциально может привести к возможности генерации сообщений, имеющих те же значения имитовставки, что и у ранее присланных сообщений, таким образом, имитовставка на основе симметричных шифров не позволит точного определить лицо сформировавшее данную имитовставку. Для ЭП подобная проблема отсутствует.

Отметим, что для отдельных приложений информационной безопасности, связанных с контролем аутентичности внутреннего (смыслового, содержательного, визуального) наполнения цифровых объектов, допускающих малозначительные для человеческого восприятия

изменения, рассмотренные выше классические подходы, будут не способны обеспечить корректное решение задачи. В качестве примера можно привести задачу контроля аутентичности содержимого некоторого отсканированного документа. Если хеш-код, MAC или CRC вычисляются для исходной, например несжатой, копии скана, то соответствующие значения хеш, MAC или CRC для внешне неотличимой сжатой копии того же изображения-скана будут совершенно другими, что справедливо будет свидетельствовать об их отличии на уровне битового представления, но не на уровне внутреннего содержимого. Кроме того, использование классических подходов контроля целостности и аутентичности будет затруднительным в случае если необходимо решать не бинарную (объект в целом аутентичен оригиналу или нет), а более тонкую задачу, связанную с локализацией фрагментов цифровых объектов, подвергшихся изменению, при проведении анализа данных объектов в отсутствие их оригинальных копий.

В указанных выше случаях для контроля аутентичности содержимого цифровых объектов и локализации их областей, содержащих потенциальные изменения, могут эффективно использоваться методы стеганографии. Так, в работе [Weng L. Robust image content authentication using perceptual hashing and watermarking / L. Weng, R. Darazi, B. Preneel, B. Macq, A. Dooms // Pacific-Rim Conference on Multimedia, PCM 2012: Advances in Multimedia Information Processing. – 2012. – P. 315-326] описан подход к контролю аутентичности содержимого цифровых изображений путем стеганографического кодирования выделенной информативно значимой составляющей контейнера в сам защищаемый контейнер. Рассмотрим его подробнее.

Контроль аутентичности изображений с использованием технологии ЦВЗ. Графические данные считаются аутентичными (в рассматриваемом контексте применения ЦВЗ) если они схожи не на уровне своего битового представления, а на уровне соответствующих им локальных дескрипторов или перцептивных хеш-значений. Построение перцептивной хеш-функции заключается в отображении среднего значения низких частот изображения, исключая таким образом его детализацию, но сохраняя и показывая его структуру. Значения хеш и локальных дескрипторов должны быть устойчивы к большинству типовых преобразований графических данных вроде фильтрации или компрессии и одновременно должны быть неустойчивы к низкочастотным преобразованиям, направленным на искажение информационной

составляющей защищаемого объекта (ручному добавлению или удалению объектов на изображении, подделке, коррекции символов в отсканированных документах и т.п.) При использовании хеш и локальных дескрипторов совместно с классическими вариантами создания невидимых ЦВЗ можно реализовать эффективные интегрированные технологии многоуровневой защиты цифровых объектов по отношению к преднамеренным воздействиям, направленным на несанкционированную коррекцию и искажение маркированного контейнера. Основное преимущество подобного подхода состоит в наличии условной зависимости между событием подмены объекта идентификации и наличии элемента защиты – скрытого водяного знака. Подмена объекта идентификации приведет к выводу о подделке документа или его части.

Алгоритм создания защищенного - маркированного контейнера включает следующие шаги.

Шаг 1. Разбиение оригинального изображения I размером $W \times H$ пикселей на непересекающиеся блоки фиксированного размера $n \times n$ пикселей

$$I_{xy}^{(b)} \subset I,\ x = \overline{1, n_W},\ y = \overline{1, n_H},\ n_W = W/n,\ n_H = H/n.$$

В общем случае размеры блока зависят от используемого алгоритма ССИ и длины встраиваемого хеш-кода, а также от разрешения изображения. Обычно n задается равным 32 или 64. В дальнейшем каждый блок обрабатывается независим.

Шаг 2. Выделение низкочастотных информативно-значимых составляющих для каждого блока. Как вариант – вычисление перцептивного хеш-кода по одному из известных алгоритмов

$$h_{xy} = hash\left(I_{xy}^{(b)}\right),$$

h_{xy} – m-битная строка (обычно $m = 64$).

Шаг 3. С использованием одного из криптографических алгоритмов шифрование перцептивного хеш-кода на секретном ключе K

$$\tilde{h}_{xy} = E_K\left(h_{xy}\right).$$

По аналогии с классическими схемами ЭП шифрование с использованием K предназначено для защиты от злоумышленных действий типа навязывания ложной информации. В качестве алгоритмов шифрования целесообразно использовать синхронные поточные шифры, исключающие эффект размножения ошибок.

Шаг 4. Стеганографическое скрытие (зашифрованного) хеш-кода в соответствующий ему блок изображения

$$\tilde{I}_{xy}^{(b)} = F\left(I_{xy}^{(b)}, \tilde{h}_{xy}, x, y, n\right).$$

В качестве стегоалгоритма целесообразно использовать робастные алгоритмы с достаточно высокой пропускной способностью. Требование робастности обуславливается необходимостью обеспечения устойчивости скрытых данных (хешей) к типовым преобразованиям, не разрушающим низкочастотной составляющей блоков. Требование высокой пропускной способности обуславливается необходимостью записи множества хеш-значений в блоки изображения относительно небольшого размера. К числу подходящих для решения задачи стегоалгоритмов можно отнести рассмотренные во второй и третьей главе алгоритмы QIM и Коха-Жао.

Шаг 5. Из блоков $\tilde{I}_{xy}^{(b)}$, $x = \overline{1, n_W}$, $y = \overline{1, n_H}$ формируется итоговое маркированное изображение $\tilde{I}$. В случае если исходные размеры I не кратны заданному n, то граничные области исходного изображения не входящие в блоки разбиения, могут быть добавлены в $\tilde{I}$ без изменения.

Обобщенная схема работы алгоритма создания маркированного изображения приведена на рис. 3.

Алгоритм проверки аутентичности содержимого контейнера-изображения включает следующие шаги.

Шаг 1. Разбиение маркированного изображения $\tilde{I}$ размером $W \times H$ пикселей на непересекающиеся блоки фиксированного размера $n \times n$ пикселей

$$\tilde{I}_{xy}^{(b)} \subset \tilde{I},\ x = \overline{1, n_W},\ y = \overline{1, n_H},\ n_W = W/n,\ n_H = H/n.$$

Размеры блоков должны совпадать с исходными, использовавшимися при формировании маркированного контейнера.

Шаг 2. Выделение низкочастотных информативно-значимых составляющих для каждого блока по алгоритму, использовавшемуся при формировании маркированного контейнера $h'_{xy} = hash\left(I_{xy}^{(b)}\right)$.

Шаг 3. Извлечение из каждого блока $\tilde{I}_{xy}^{(b)}$ стеганографически скрытых зашифрованных перцептивных хеш-кодов

$$\tilde{h}_{xy} = F^{-1}\left(\tilde{I}, x, y, n\right).$$

Шаг 4. Расшифрование извлеченных данных на ключе K и восстановление хеш-кодов $h_{xy} = E_K^{-1}\left(\tilde{h}_{xy}\right)$.

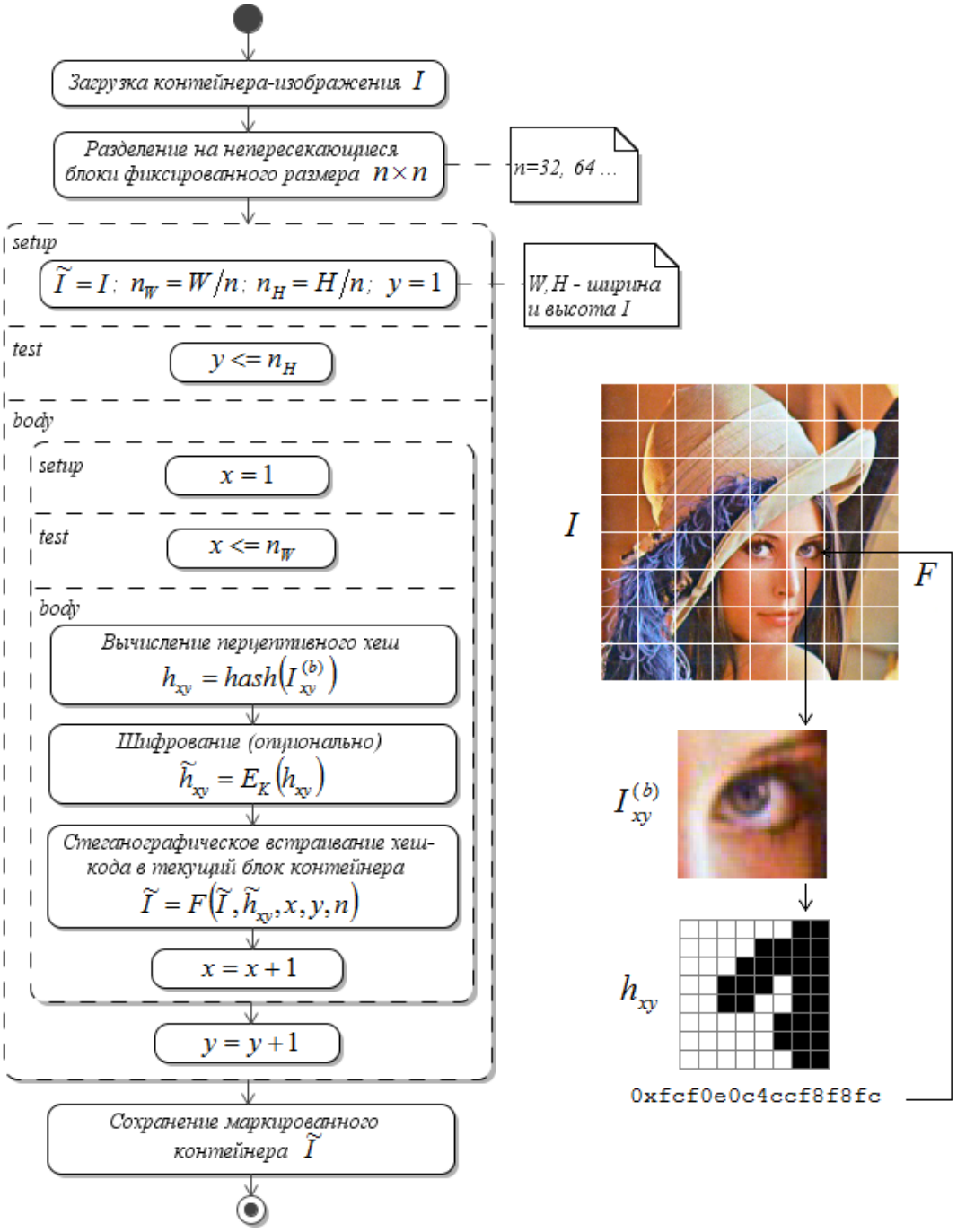

Рис. 3. Схема создания маркированных контейнеров-изображений в задаче контроля аутентичности их содержимого

При использовании в качестве E синхронного поточного шифра число искаженных бит зашифрованной последовательности $\tilde{h}_{xy}$ будет равно числу искаженных бит в расшифрованной последовательности h_{xy}.

Шаг 5. Сравнение вычисленного h'_{xy} и извлеченного h_{xy} хеш-значений с использованием заданной функции расстояния. В качестве

последней, например, можно использовать нормализованное расстояние по Хеммингу

$$\rho_{xy} = \rho\left(h'_{xy}, h_{xy}\right) = \frac{1}{n \cdot n} \sum_{h'_{xy}(i) \neq h''_{xy}(i),\ i=0}^{n \cdot n} 1, \quad x = \overline{1, n_W}, \ y = \overline{1, n_H}.$$

Расстояние Хэмминга определяется числом позиций, в которых соответствующие символы двух слов одинаковой длины различны. Если $\rho_{xy} < \tau$, где τ – параметрически задаваемое пороговое значение, определяющее минимально допустимые различия хеш-кодов, то блок изображения с координатами (x, y) считается аутентичным, в противном случае (x, y)-й блок помечается, как потенциально измененный.

Обобщенная схема проверки аутентичности содержимого блоков маркированного изображения приведена на рис. 4.

Алгоритмы вычисления перцептивных хеш. Перцептивные хеш-алгоритмы предназначены для описания класса функций, генерирующих сравнимые хеши изображений. Перцептивные хеш-функции извлекают определенные признаки из изображения и на их основе вычисляют хеш – индивидуальный «отпечаток» содержимого изображения. В отличие от криптографических алгоритмов хеширования, формирующих принципиально разные хеш-значения для минимально отличающихся копий исходных данных, алгоритмы перцептивного хеширования для схожих изображений будут формировать близкие хеш-значения. Сравнивая между собой перцептивные хеши можно делать вывод о степени различия двух наборов данных. Именно это свойство и используется для определения степени схожести изображений в рассматриваемой задаче контроля аутентичности. Далее рассмотрим некоторые из наиболее распространенных и эффективных алгоритмов формирования перцептивных хеш.

Перцептивный хеш по среднему aHash (Average hash). Это один из простейших вариантов формирования хеш, заключающийся в отображении среднего значения низких частот изображения. Включает следующие шаги.

1) Уменьшение размера изображения до 8×8 пикселей (длина хеш 64 бита).

2) Приведение к представлению в градациях серого.

3) Вычисление среднего значения яркости $l = \frac{1}{64} \sum_{x=0}^{7} \sum_{y=0}^{7} J(x, y)$, где J – уменьшенная копия кадра в градациях серого размером 8×8 пикселей.

4) Построчная развертка пикселей J в вектор v.

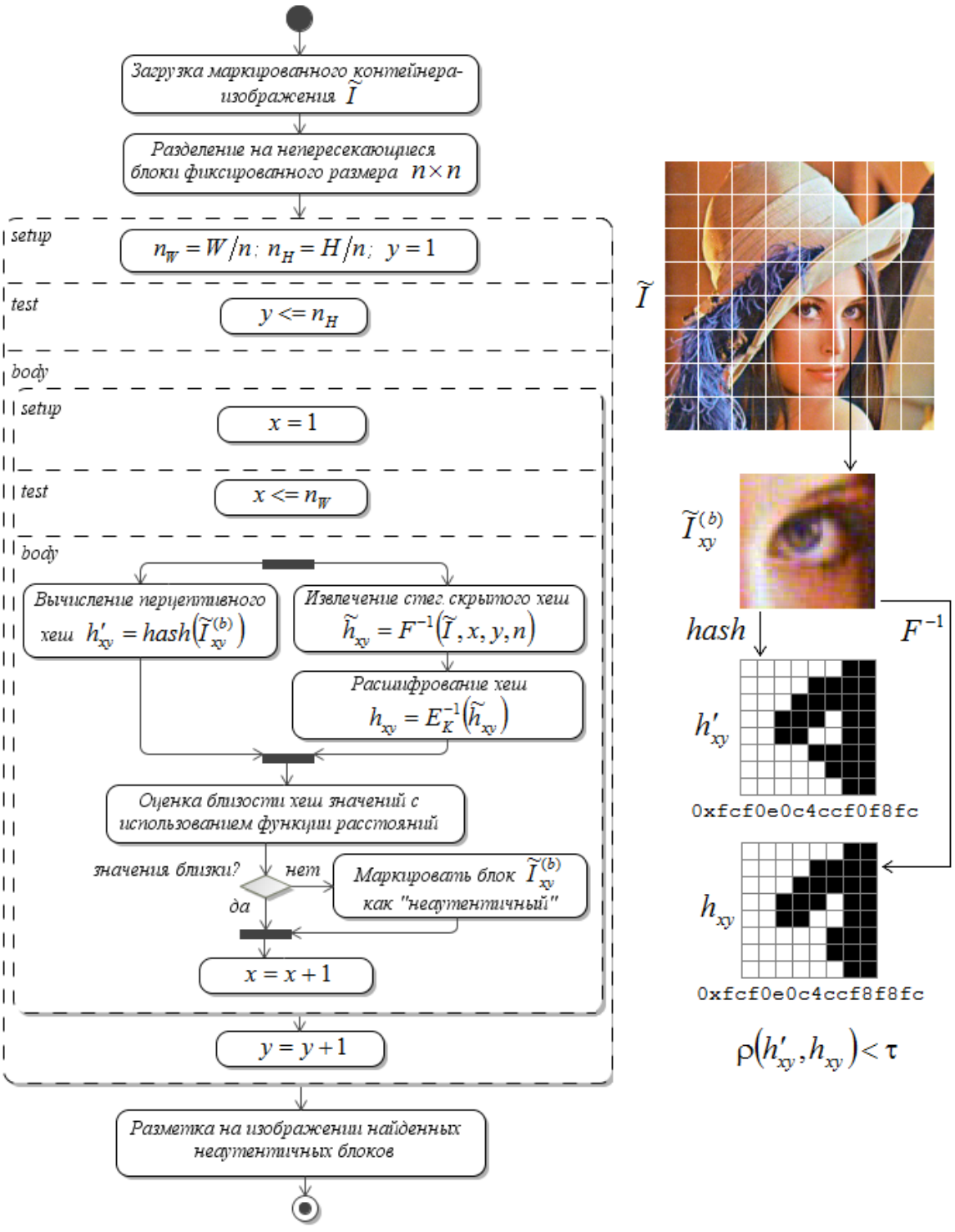

Рис. 4. Схема проверки аутентичности содержимого контейнеров-изображений

5) Бинарная сегментация элементов ν по l: $h_i = \begin{cases} 0, \nu_i < l \\ 1, \nu_i \geq l \end{cases}$, $i = \overline{0,63}$.

Полученный в результате двоичный вектор h является перцептивным хеш-кодом, который в дальнейшем используется в качестве встраиваемой в защищаемое изображение цифровой метки, контролирующей его аутентичность.

К достоинствам алгоритма *aHash* можно отнести высокую скорость работы, а к недостаткам – чувствительность к операциям яркостной коррекции, приводящим к изменению среднего по яркости.

Перцептивный хеш на основе дискретного косинусного преобразования pHash (Perceptive Hash). Алгоритм формирования хеш включает следующие шаги.

1) Уменьшение размера изображения до 32×32 пикселей.

2) Приведение к представлению в градациях серого.

3) Вычисление дискретного косинусного преобразования

$$C(u,v) = \frac{1}{4}\lambda(u)\lambda(v)\left[\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} J(x,y)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N}\right],$$

где C – матрица коэффициентов ДКП размером $N \times N$, J – уменьшенная копия кадра в градациях серого размером $N \times N$, $N = 32$, $\lambda(u) = \lambda(v) = 1/\sqrt{2}$, когда u и v равны нулю и $\lambda(u) = 1$, $\lambda(v) = 1$ в других случаях.

4) Сокращение матрицы коэффициентов ДКП. Выделение в матрице C левого верхнего подблока размером 8×8, содержащего коэффициенты, соответствующие низкочастотной составляющей обрабатываемого изображения. Длина хеш в этом случае составит 64 бита.

5) Построчная развертка выделенного блока в вектор с вычислением среднего значения его элементов $t = \frac{1}{64}\sum_{i=0}^{63} c_i$, где c_i – элементы вектора коэффициентов ДКП. Вместо среднего также может вычисляться медианное значения коэффициентов.

6) Сегментация вектора низкочастотных коэффициентов с использованием среднего t в качестве порога $h_i = \begin{cases} 0, & c_i < t \\ 1, & c_i \geq t \end{cases}$, $i = \overline{0,63}$.

Двоичный вектор h является перцептивным хеш-кодом.

Перцептивный хеш на основе градиента dHash (Difference Hash). Алгоритм формирования хеш включает следующие шаги.

1) Уменьшение размера изображения до 9×8 пикселей.

2) Приведение к представлению в градациях серого.

3) Построчное вычисление разницы между значениями яркости соседних пикселей на уменьшенной копии кадра в градациях серого. В результате формируется матрица J размером 8×8.

4) Построчная развертка матрицы J в вектор v.

5) Формирование вектора h (хеш-кода) по правилу $h_i = \begin{cases} 0, v_i \le 0 \\ 1, v_i > 0 \end{cases}$, $i = \overline{0,63}$.

К достоинствам алгоритмов *pHash* и *dHash* можно отнести устойчивость к большинству типовых преобразований (гамма-коррекции, изменению гистограммы, изменению контрастности, JPEG-сжатию и масштабированию).

Перцептивный хеш на основе оператора Марра-Хилдрета (Marr-Hildreth Operator Based Hash). Алгоритм формирования хеш включает следующие шаги.

1) Приведение изображения к представлению в градациях серого.

2) Уменьшение размера изображения до 128×128 пикселей.

3) Гауссово размытие уменьшенного представления. Данный шаг необходим для снижения влияния шумов при последующем определении границ объектов на изображении.

4) Построение оператора Марра-Хилдрета. Данный оператор представляет собой оператор Лапласа от фильтра Гаусса (LoG, Laplacian of Gaussian) и определяется в виде

$$h_{x,y}^{(c)} = \nabla^2 G_{x,y}^{(c)} = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \cdot G_{x,y}^{(c)}$$

где $G_{x,y}^{(c)} = e^{-\frac{x^2+y^2}{2\sigma^2}}$ – фильтр Гаусса. Для использования LoG в дискретной форме выполняется его дискретизация с заданной переменной масштаба (по умолчанию 1).

5) Применение оператора LoG к изображению, используя дискретную свертку $L_{x,y}^{(c)} = h_{x,y}^{(c)} * J_{x,y}^{(c)}$, где $J^{(c)}$ – преобразованное в градации серого уменьшенное представление исходного изображения, $*$ – операция свертки.

6) Уменьшение размера изображения $L^{(c)}$ до 8×8 пикселей с построчной его разверткой в вектор h, в результате чего получается хеш-код размером 64 байта.

К достоинствам алгоритма перцептивного хеширования на основе оператора Марра-Хилдрета можно отнести хорошую устойчивость к масштабированию, затемнению и сжатию изображений.

Стоит отметить, что для выделения информативно значимой низкочастотной составляющей изображений в схеме создания маркированных контейнеров (рис. 3) помимо перцептивных хеш могут рассматриваться алгоритмы вычисления контрольных точек и дескрипторов (SIFT, SURF), а также градиентные методы и алгоритмы выделения границ (операторы Лапласа, Canny и пр.).

На рис. 5 приведен пример работы алгоритма контроля аутентичности содержимого цифровых изображений.

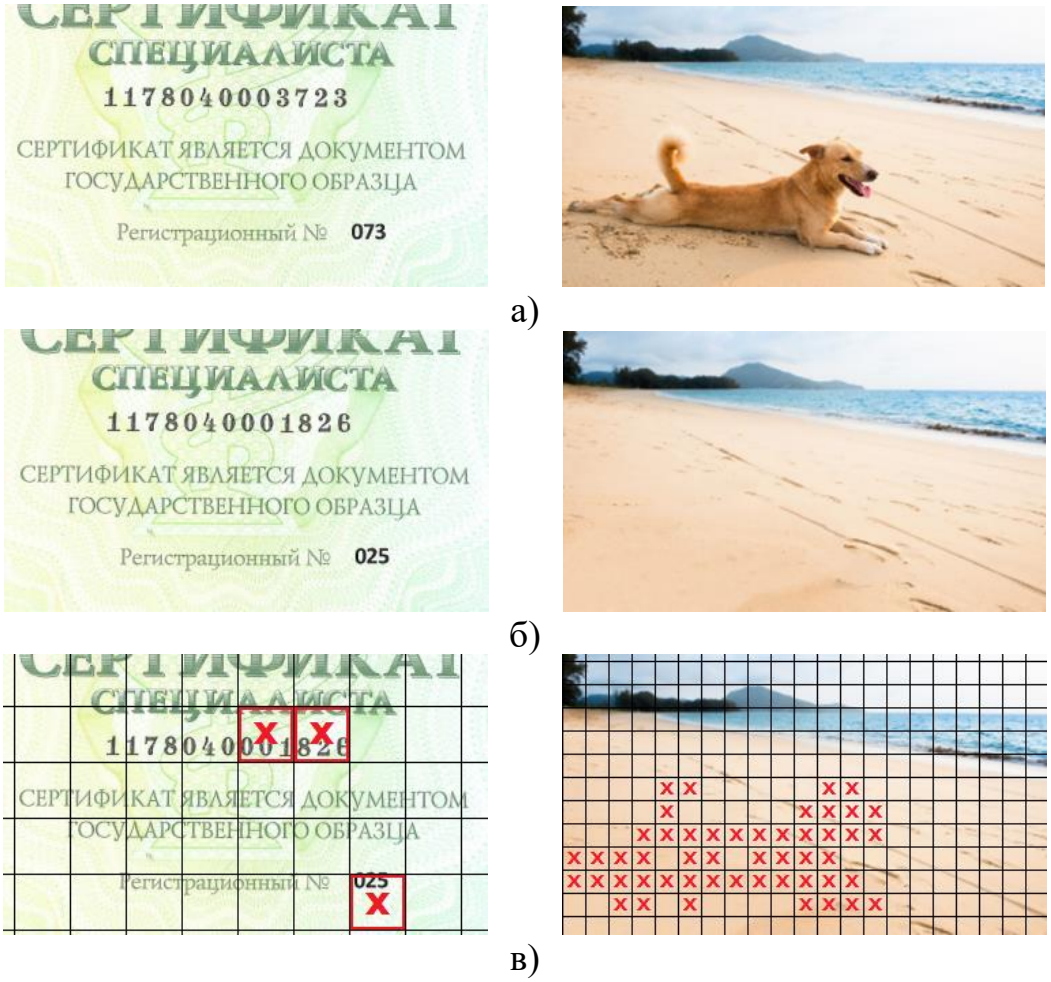

Рис. 5. Фрагменты исходных контейнеров-изображений (а); фрагменты маркированных контейнеров, подвергшихся редактированию (б); результат проверки аутентичности содержимого маркированных контейнеров с разметкой неаутентичных блоков (в)

Фрагменты исходных контейнеров представлены на рис. 5а. На рис. 5б приведены фрагменты маркированных контейнеров (размер блока 64×64 пикселей, функция хеширования *pHash*, алгоритм ССИ QIM), подвергшихся значимой коррекции содержимого (моделирование работы злоумышленника). На изображении слева скорректированы три из четырех последних цифр номера сертификата, а также последние две цифры регистрационного номера. На правом изображении с использованием алгоритма Inpaint удалено изображение собаки. На рис. 5в приведена разметка блоков маркированных искаженных контейнеров по результатам вычисления нормализованного расстояния по Хеммингу ($\tau = 0.078$). Потенциально измененные блоки отмечены символом «X».

Лабораторные работы

Реализовать схему контроля аутентичности содержимого цифровых изображений на основе стеганографического встраивания информативной низкочастотной составляющей блоков изображения в само изображение. Схема должна обеспечивать не только контроль аутентичности содержимого исходного изображения в целом, но также возможность локализации областей, подвергшихся искажениям низкочастотного характера (например, редактирование отдельных цифр/символов отсканированного текста, удаление информации о правообладателе в растровом представлении и т.п.).

Смоделировать работу злоумышленника – реализовать различные типы негативных воздействий по отношению к маркированным контейнерам (фильтрацию, компрессию, дорисовку/удаление объектов и т.п.). Оценить вероятности ошибок первого и второго рода при проверке аутентичности содержимого блоков контейнера для различных типов маркируемых изображений (сканированные текстовые документы, фотографические изображения), различных типов внесенных негативных воздействий и различных значений порогов τ.