$$U = \begin{pmatrix} X \\ Y \end{pmatrix}.$$

6. Если $A = 1$, выдать $W \leftarrow X$ и остановить выполнение алгоритма.
7. Если $A = 4$ и Y четные, выдать $W \leftarrow (4X + BY)/2$ и остановить выполнение алгоритма.

Выводы

1. Реализация ЦП в группах точек ЭК позволяет повысить стойкость криптопреобразований и одновременно уменьшить размеры общесистемных параметров и ключей. По сравнению с преобразованиями в полях с ростом длины модуля преобразований выигрыш увеличивается.
2. Проведенные нами сравнения показали, что алгоритм ECDSA обладает рядом преимуществ в области стойкости.
3. Алгоритмы ECDSA и ECSS обладают приблизительно одинаковой вычислительной сложностью прямых и обратных преобразований. Нами выявлены алгоритмы для ECDSA, позволяющие ускорить процедуры проверки подписи.
4. С учетом возможного появления новых методов и более производительных средств криптоанализа необходимо предусмотреть различные значения параметров ЭК. Например, использовать кривые с порядком базовых точек 2^{160}, 2^{192}, 2^{224}, 2^{256}, 2^{384}, 2^{512}.

31 октября 2001 года завершился первый двухлетний этап рассмотрения алгоритмов – кандидатов на стандарты в разных областях криптографии. После тщательного исследования различными криптографами мира алгоритм ЦП ECDSA прошел первый отборочный тур, а алгоритм ECSS не был представлен на рассмотрение [4].

Литература: 1. IEEE P1363 / D9 (Draft Version 9). Standard Specifications for Public Key Cryptography. Number-Theoretic Background. 1999. 2. www.rsa.com 3. X9.62 – 1999 Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA). 4. crypto.nessie.org.

УДК 681.3

КОНТРОЛЬ И ВОССТАНОВЛЕНИЕ ЦЕЛОСТНОСТИ ИНФОРМАЦИИ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ

Николай Будько, Вячеслав Василенко, Михаил Короленко
Открытое акционерное общество "КП ВТИ"

Аннотация: **Предлагается использование помехоустойчивого корректирующего кода для задач контроля, контроля и восстановления целостности информационных объектов автоматизированных систем.**
Summary: **Use of a noiseproof adjusting code for tasks control, control and restoration of information integrity the objects automated systems is offered.**
Ключевые слова: **Информация, целостность, контроль, восстановление.**

I Введение

В современных условиях обеспечение высокой надежности, эффективности и технологичности автоматизированных систем (АС) возможно только при условии обеспечения высокого уровня защищенности информации, которая циркулирует в этих АС. Для этого в соответствии с законами Украины об информации и ее защите, а также в соответствии с нормативными документами Системы технической защиты информации (ТЗИ) Украины необходимо применение в АС специальных средств защиты, которые предназначаются для достижения оптимального для данной АС объединения четырех свойств защищённости информации автоматизированных систем [1, 2, 3]: конфиденциальности, целостности, доступности и наблюдаемости. В зависимости от условий применения, сложности и класса АС, а также характеристик возможных угроз вес этих функциональных свойств может изменяться, но проблема обеспечения целостности является одной из основных при разработке и внедрении любых защищенных АС.

Система ТЗИ обеспечивает целостность информации, если она сохраняется, передается ли обрабатывается достоверной, полной и защищенной от неумышленных и преднамеренных искажений.

Одним из основных способов обеспечения целостности информации в автоматизированных системах является применение средств контроля целостности программных средств и обрабатываемой информации, включая в некоторых случаях и её восстановление.

Не останавливаясь на причинах нарушения целостности [4], следует подчеркнуть, что часть угроз целостности, в частности со стороны авторизованных пользователей и случайного влияния естественных и технических факторов, может быть выявленной, а, следовательно, и устраненной лишь за счет применения эффективных механизмов контроля и восстановления целостности, в которых используются процедуры защищённых от подделок преобразований информации. Это связано с тем, что основной задачей средств контроля целостности информационных ресурсов есть обеспечение такого состояния системы, когда невозможно утаивание факта любой несанкционированной модификации защищенной информации (вставки, изъятие, подмена и т. п.). С этой целью в состав информации, которая защищается, включают избыточную информацию – образ, отображение этой информации, процедура формирования которого известна лишь собственнику информации и авторизованным пользователям. Т. е. образы, которые формируются, должны обладать определенной стойкостью к подделкам – имитостойкостью. К механизмам контроля целостности относятся известные механизмы защиты [4] с использованием:

1. сигнатур важных объектов (в том числе хеш-функций);
2. цифровой подписи;
3. процедур помехоустойчивого кодирования для обеспечения целостности архивной информации, в том числе и резервных копий программных средств и баз данных.

Однако, не все из этих механизмов обеспечивают потребную имитостойкость (см. ниже), в силу чего в настоящее время для контроля целостности информации в Украине межгосударственным стандартом (ГОСТ 34.310–94) предусмотрено применение процедуры формирования цифровой подписи на базе функции хеширования (ГОСТ 34.311–95), имитостойкость которых обеспечивается использованием процедур криптографического преобразования по ГОСТ 28147–89 и алгоритма RSA.

При этом известные механизмы защиты с использованием сигнатур или цифровых подписей важных объектов базируются на применении процедур выявления нарушений целостности и на последующем восстановлении искажённой информации за счет повторной передачи неискажённой информации или повторной записи неискажённой информации с резервной копии. Обе эти операции требуют значительных временных затрат.

Из рассмотренного можно сделать вывод о том, что повышение оперативности процессов обеспечения целостности возможно за счет разработки и применения согласованных между собою быстродействующих процедур как выявления нарушения целостности информации, так и ее восстановления. Такими процедурами являются процедуры, которые основываются на применении корректирующих помехоустойчивых кодов. Однако известные помехоустойчивые коды не в состоянии обеспечить главное из необходимых при этом свойств – имитостойкость, вследствие чего их использование в механизмах контроля целостности является невозможным.

Это связано с тем, что механизмы формирования контрольных признаков, которые можно было бы использовать в качестве соответствующих образов (сигнатур, хеш-функций и т. п.) не обеспечивают скрытности их формирования, так как константы этих кодов (элементы кодировочных таблиц, см. ниже) являются, как правило, общеизвестными. В отдельных случаях, когда такую скрытность можно было бы обеспечить (пример – коды Рида-Соломона) количество элементов преобразования (подматриц кодировочной матрицы) является ограниченным настолько, что трудно говорить о требуемой имитостойкости соответствующих контрольных признаков.

В статье предлагается использование помехоустойчивого кода, пригодного как для контроля, так и для восстановления целостности информационных объектов, который по совокупности своих свойств, по мнению авторов, превосходит известные.

II Краткие сведения о кодовых преобразованиях

Под кодовыми преобразованиями будем понимать результат умножения исходного кода A длиной в n символов (слова определённого алфавита, числа в некоторой системе счисления и т. п.), с возможным расширением его до k символов, рассматриваемого как матрица размерности $(1\times n)$, где n – число символов этого исходного кода, на кодировочную матрицу G размерности $(k\times k)$, где $k \geq n$, элементами которой есть некоторые числа или подматрицы. Наиболее общие требования к построению кодировочных матриц рассмотрены ниже. Операции умножения и сложения при вычислении элементов закодированного слова (при умножении матриц) могут быть либо обычными, либо модульными – выполнятся (все или отдельные из них) по модулю (в зависимости от типа кода – малой или большой величины) либо логическими, в том числе в виде поразрядных логических сложений и умножений.

В результате такого умножения получают преобразованный код – матрицу $B = A \times G$ размерностью $(1 \times k)$. Ясно, что для обратного преобразования, то есть для получения исходного кода A из B достаточно выполнить умножение B на матрицу G^{-1}, обратную G: $A = B \times G^{-1} = A \times G \times G^{-1}$. Матрица G в теории помехоустойчивого кодирования носит название порождающей, а матрица G^{-1} – проверочной.

Размерность кодировочной матрицы G (рис. 1), правила выбора, использования или формирования её элементов (подматриц) определяются видом преобразования, а также возможностями построения обратных матриц G^{-1}. Для определённости будем считать, что условия существования обратной матрицы G^{-1} выполняются. Обычно и кодировочная и проверочная матрицы по причинам, изложенным ниже, имеют размерность $(k \times k)$. Поскольку размерность k превышает длину исходного кода n, то возможны варианты использования элементов матрицы G или доведения длины исходного кода до k.

Первый вариант – криптопреобразования. При использовании исходного кода длиной в n символов и подматрицы g матрицы G (рис. 1) из n строк и n столбцов (или, что то же самое отдельной матрицы $(n \times n)$) и при определённых правилах выбора или формирования её элементов можно получить матрицы для криптографических преобразований (шифрования) исходного текста.

Код, полученный в результате умножения исходного кода на кодировочную матрицу, является некоторым криптографическим преобразованием исходного кода. Если механизм формирования элементов кодировочной матрицы является секретным, или механизм формирования элементов кодировочной матрицы является общеизвестным, но при их формировании используются некоторый секретный параметр – ключ, то зашифрованный код имеет определённую криптографическую стойкость, т. е. устойчивость к попыткам криптоаналитиков получить из зашифрованного кода исходный.

Такая криптографическая стойкость является основным свойством таких преобразований и достаточно часто определяется числом вариантов ключей.

$$G = \begin{pmatrix} g_{11} & g_{12} & \cdot & g_{1n} & \cdot & g_{1k} \\ g_{21} & g_{22} & \cdot & g_{2n} & \cdot & g_{2k} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{n1} & g_{n2} & \cdot & g_{nn} & \cdot & g_{nk} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{k1} & g_{k2} & \cdot & \cdot & \cdot & g_{kk} \end{pmatrix}$$

Подматрица g $(n \times n)$

Рисунок 1 – Общий вид кодировочной матрицы

Второй вариант – помехоустойчивое кодирование. Описанные в варианте 1 преобразования обеспечивают чрезвычайно важное свойство защищённости информации – конфиденциальность, однако не позволяют решать проблему контроля, а тем более, восстановления целостности информации. (Единственным, пожалуй, исключением является случай, когда факт невозможности дешифровки зашифрованного слова можно истолковать как факт наличия в нём искажения). Это связано с тем, что операция вычисления новой матрицы $B = A \times g$ не приводит к увеличению в закодированном слове количества информации (появлению в нём новой информации), необходимой для последующего обнаружения факта искажения, места искажения и его величины.

Следовательно, для преобразований, позволяющих осуществлять контроль целостности (возможно с последующим её восстановлением) необходимо ввести потребную для этого дополнительную информацию, т. е. использовать матрицы размерности $k > n$ и, как следствие этого, исходные слова для кодирования длиной k символов. Тогда исходное слово из n символов преобразуются в закодированное слово, как вариант – в помехоустойчивый код, длиной в k символов.

Данный вариант предусматривает расширение исходного кода длиной в n символов до исходного слова для кодирования длиной в k символов и использование кодировочной матрицы специального вида – порождающей матрицы (в терминах помехоустойчивого кодирования). Наиболее простой процедурой превращения исходного кода длиной в n символов в исходное слово для кодирования длиной в k символов является добавление (вставка) $r = (k - n)$ дополнительных символов, например в конец исходного кода (в некоторых кодах, например, в кодах Хемминга, такая вставка может осуществляться и между символами исходного кода). Порождающая матрица в этом случае (рис. 2) в качестве подматрицы g содержит единичную матрицу, r дополнительных строк и столбцов, элементы которых в n строках определяются

требуемыми свойствами (типом) помехоустойчивого кода. В результате умножения исходного слова на кодировочную матрицу получают k – символьный код, в котором первые n элементов совпадают с соответствующими элементами исходного кода, а информация, которая формируется в дополнительных, избыточных r символах закодированного слова в теории помехоустойчивого кодирования носит название контрольного признака.

Если, например, использовать кодировочную матрицу, в которой $r - n = 1$, а n элементов k-го столбца равны единице, то получим обнаруживающий помехоустойчивый код (рис. 3), в котором контрольный признак получается путём суммирования (например, поразрядного логического или по модулю 2^b, где b – двоичная длина символов исходного кода т. е. его длина в битах, и т. д.) всех n элементов исходного кода (эквивалент контрольного суммирования).

$$G = \begin{pmatrix} 1 & 0 & . & 0 & . & g_{1k} \\ 0 & 1 & . & 0 & . & g_{2k} \\ . & . & . & . & . & . \\ 0 & 0 & . & 1 & . & g_{nk} \\ . & . & . & . & . & . \\ 0 & 0 & . & . & . & 1 \end{pmatrix}$$

Единичная подматрица g $(n \times n)$

Рисунок 2 – Общий вид кодировочной матрицы для помехоустойчивых кодов

$$G = \begin{pmatrix} 1 & 0 & . & 0 & . & 1 \\ 0 & 1 & . & 0 & . & 1 \\ . & . & . & . & . & . \\ 0 & 0 & . & 1 & . & 1 \\ . & . & . & . & . & . \\ 0 & 0 & . & . & . & 1 \end{pmatrix}$$

Единичная подматрица g $(n \times n)$

Рисунок 3 – Общий вид кодировочной матрицы для частного случая обнаруживающего кода (контрольное суммирование)

При умножении закодированного слова на проверочную матрицу G^{-1} получают вектор-строку ошибок, элементы которой при выборе избыточности, достаточной для решения задач исправления ошибок (корректирующие коды) несут информацию о наличии, месте и величине искажений в проверяемом коде. При недостаточной избыточности эти элементы несут информацию о месте или просто о наличии искажений (обнаруживающие коды).

Основными характеристиками помехоустойчивых кодов являться вероятность обнаружения факта наличия искажений в исходном коде (т. е. вероятность обнаружения нарушения целостности) $p_{\text{иск}}$ и необходимая при этом избыточность. Известно, что при использовании таких кодов вероятность обнаружения $p_{\text{иск}} = 2^{-br}$. При этом требуемая избыточность зависит от необходимых свойств кода (обнаружения факта, места и величины искажения).

В дальнейшем речь идёт о помехоустойчивом коде, принадлежащем в зависимости от допустимой избыточности к классу корректирующих или обнаруживающих и позволяющем, при этом, решать задачи контроля, либо контроля и восстановления целостности контролируемых слов.

Третий вариант – помехоустойчивая криптография. Предварительно следует отметить тот факт, что зачастую задачи обеспечения конфиденциальности и целостности информации приходится решать одновременно по отношению к одним и тем же информационным объектам. При этом наиболее часто процедура такой защиты заключается в последовательном применении криптографического преобразования и помехоустойчивого кодирования зашифрованного текста. На приёмной стороне вначале проверяется и восстанавливается целостность, а потом осуществляется дешифрование текста. Т. е. речь идёт о двуфазности

процедур преобразования на обеих сторонах и, как следствие этого, о снижении общего быстродействия средств, реализующих эти процессы.

Отметим [5], что использование кодировочной матрицы вида 1 позволяет использовать однофазные процедуры преобразования, что, по мнению авторов, позволит повысить общее быстродействие средств преобразования. Как вариант помехоустойчивого криптографического преобразования, может быть предложено преобразование исходного цифрового кода, который изначально считается представлением в позиционной системе счисления некоторого числа A, в систему счисления в остаточных классах. C этой целью следует: символы исходного кода рассматривать как символы выбранного позиционного представления a_i (i = 1, 2, …, n) с соответствующими весовыми коэффициентами c_i; расширить исходный код на один (в наиболее простом случае) символ $a_к$ = 0; в качестве элементов кодировочной матрицы g использовать величины

$$g_{ij} = \{c_i\}_{pj},$$

где j = 1, 2, …, k, знак $\{c_i\}_{pj}$ означает вычисление остатка (вычета) от деления c_i на p_j. Все операции при вычислении символов преобразованного кода α_i следует выполнять по модулю p_j где p_j – соответствующее основание системы счисления в остаточных классах. При этом исходный код $A = a_1, a_2, ..., a_n, 0$ преобразуется в число в остаточных классах $A_{сок} = \alpha_1, \alpha_2, ..., \alpha_n, \alpha_k$, по отношению к которому применимы все известные механизмы контроля, либо контроля и восстановления целостности, изложенные в разделе 3.

Если при этом правило выбора оснований p_j или величины этих оснований не известны неавторизованному пользователю, то код $A_{сок}$ обладает и определенной криптографической стойкостью, анализ которой выходит за пределы данной статьи и которую нетрудно довести до требуемой. Т. е., механизмы, предлагаемые ниже на базе преобразований из области счисления в системе остаточных классов, позволяют использовать их в задачах помехоустойчивой криптографии.

III Код условных вычетов

По мнению авторов для осуществления всех описанных выше преобразований можно использовать [5] код условных вычетов (лишків умовних код – ЛУ-код). Это связано с двумя аспектами. Первый обусловлен тем, что на базе этого кода можно строить алгоритмы контроля целостности, контроля и восстановления целостности, криптографических преобразований с контролем целостности и, наконец, криптографических преобразований с контролем и восстановлением целостности. Второй из них заключается в том, что данный код одновременно с контролем и восстановлением целостности обеспечивает и нужный уровень имитостойкости. В рамках данной статьи рассматриваются возможности кода по контролю, контролю и восстановлению целостности.

В этом коде, как и в других, для контроля наличия искажений – контроля целостности, а в дальнейшем и восстановления целостности базовых кодовых слов – нужно иметь r дополнительных (избыточных) символов, которые сохраняют в себе в специфическом виде – в виде или контрольного признака (в терминах помехоустойчивого кодирования), или хеш-функции (в терминах криптографических преобразований) образ – некоторое отображение информации базовых кодовых слов, которая контролируется по ее состоянию на время формирования этого образа (а не после нарушения целостности!).

В общем случае базовым кодовым словом называется часть обобщенного кодового слова $A = \alpha_1, , ..., \alpha_n$, длина которого n в символах (α_i), например в байтах, удовлетворяет равенству

$$N = n \cdot \lambda ,$$

где: N – число символов в обобщенном кодовом слове (в блоке информации); λ – число базовых кодовых слов (глубина перемежения) в обобщенном кодовом слове; n – число символов в базовом кодовом слове (длина базового кодового слова).

В свою очередь, под обобщенным кодовым словом понимается некоторая часть файла или последовательного набора данных – блок информации длиной N символов.

Контрольный признак H любого из базовых кодовых слов при применении процедур ЛУ-кода формируется с использованием кодировочной матрицы вида, представленного на рис. 2, элементами k–го ($k = n + 1$) столбца которой являются величины $g_{ki} = m_i/p_i$ (i = 1,…, n), а операции умножения при вычислении контрольного признака выполняются по модулю $p_к$, где $p_к$ – так называемое контрольное основание. Вычисление скалярного произведения базового кодового слова (вектор – строки) на этот k–й столбец кодировочной матрицы эквивалентно вычислениям контрольного признака H в соответствии с выражением

$$H = \left\{ p_k - \left\{ \left[Z \cdot p_k \right] \cdot R_k \right\}_{p_k} \right\}_{p_k}, \qquad (1)$$

где:

$$Z = \sum_{i=1}^{i=n1} \frac{\alpha_i \cdot m_i}{p_i} - \left[\sum_{i=1}^{i=n1} \frac{\alpha_i \cdot m_i}{p_i} \right]$$

В этих выражениях:

знак $\{X\}_y$ означает операцию по модулю y (вычисление остатка от деления X на y), а знак $[X]$ – вычисление целой части переменной X.

переменная $n1$ при формировании контрольного признака принимает значение n (то есть $n1 = n$), а при контроле целостности и коррекции искажений – значение $(n + r)$, r – количество дополнительных (избыточных) байтов, необходимых для решения задачи выявления факта наличия, места и величины искажений;

переменная α_i – числовой (двоичный) эквивалент i-го информационного символа контролируемой части файла (базового кодового слова);

переменная m_i – константа ЛУ-кода, – "вес" так называемого ортогонального базиса;

p_i – i-ое $(i = 1, ..., k)$ основание (элемент криптографического ключа, с помощью которого обеспечивается нужная имитостойкость, см. дальше). Эти основания выбираются как совокупность из k взаимно простых чисел;

i – номер основания $(i = 1, 2,..., n1)$.

Величины p_i должны, исходя из теории ЛУ-кода [5], всегда превышать любое значение α_i, а переменную p_k следует избирать из условия

$$p_k > p_n \cdot p_{n-1},$$

где величины p_n и p_{n-1} есть наибольшими из оснований p_i. При этом, поскольку величины α_i, из условия обеспечения технологичности [5] механизмов формирования контрольных признаков следует выбирать восьмибитовыми, то величины p_i должны быть, как минимум, девятибитовыми, а переменная p_k, соответственно, должна иметь не менее 18 разрядов. Последнее, снова таки из условия обеспечения технологичности, приводит к необходимости иметь переменную p_k разрядностью не менее 3 байтов.

С этой целью переменная p_k выбирается в виде произведения r простых чисел – контрольных оснований $p_{k1}, p_{k2}, ..., p_{kr}$ из диапазона восьмиразрядных чисел так, чтобы это произведение превышало бы произведение двух наибольших оснований (p_n и p_{n-1}) из их набора p_i:

$$\prod_{i=1}^{i=r} p_{ki} \geq p_n \cdot p_{n-1} \cdot$$

Переменная r определяет при этом нужное количество контрольных оснований и, через них, – длину контрольного признака.

Переменные m_i и величины p_i есть базовыми константами ЛУ-кода. Расчет базовых констант (весовых коэффициентов) m_i осуществляется исходя из стандартных процедур, и потому не приводится, а величины p_i есть элементами конфиденциального ключа (секретным элементом при известном алгоритме преобразования) и выбираются пользователем (собственником информации).

После формирования контрольного признака его значение для любого из базовых кодовых слов записывается либо после информационных наборов, либо в другом месте файла, который контролируется, либо в отдельном файле и сохраняется для последующего контроля целостности этих же базовых кодовых слов.

Контрольный признак обобщенных кодовых слов формируется как конкатенация контрольных признаков всех базовых кодовых слов.

Примечание 1. Под конкатенацией двух символов понимается слово, длина которого в битах (байтах) равняется сумме числа бит (байтов) любого из символов. При этом левая половина данного нового слова есть первым словом, а правая – вторым словом.

Результирующий контрольный признак текста (файла) формируется как конкатенация контрольных признаков всех блоков информации – обобщенных кодовых слов – и сохраняется (записывается) в конце файла или после всех файлов носителя информации.

Для последнего из обобщенных кодовых слов в случае, если количество базовых кодовых слов в нем есть величиной не целой, меньшей чем λ, или если длина файла есть небольшой, в качестве контрольных признаков отсутствующих базовых кодовых слов принимаются нулевые контрольные признаки, то есть контрольные признаки, в которых каждый из r байтов, которые входят в их состав, есть арифметическим нулем.

При организации контроля и восстановления целостности путем использования свойств ЛУ-кода, в соответствии с алгоритмом вычисления контрольных признаков базовых кодовых слов, осуществляется

вычисление величины Z в соответствии с выражением (1), в котором переменная $n1$ принимает значение $n = n + r$.

Таким образом, при вычислении величин Z для любого из базовых кодовых слов на этапе контроля целостности используются не только символы контролируемой информационной части файла (носителя) – базового кодового слова, но и символы контрольного признака этого базового кодового слова.

Полученное при этом значение величины Z сравнивается с константой ЛУ-кода

$$Z<1/p_k,$$

где переменная p_k, как и раньше, – контрольное основание ЛУ-кода.

Если это неравенство удовлетворяется, то это есть критерием того, что целостность данного базового кодового слова не нарушена, и осуществляется контроль целостности следующего базового кодового слова, до тех пор пока не осуществится контроль всего носителя.

Если это неравенство не удовлетворяется, то это есть критерием того, что целостность данного базового кодового слова нарушена, и осуществляется его восстановление в соответствии с ниже изложенным Z-алгоритмом восстановления целостности. После этого осуществляется контроль целостности следующего базового кодового слова до тех пор, пока не закончится контроль всего носителя.

Восстановление информации при контроле целостности с использованием свойств ЛУ-кода не требует использования резервных копий, а есть сугубо расчетным с полным использованием информации, которая сосредоточена в избыточных символах – в контрольных признаках любого из базовых кодовых слов.

В соответствии с Z-алгоритмом коррекция искажённой переменной $\widetilde{\alpha}_i$, то есть вычисление неискаженного значения этой же переменной α_i, происходит в соответствии с выражением

$$\alpha_i = \{\widetilde{\alpha}_i - \{[Z \cdot p_i] \cdot R_i\}_{pi}\}_{pi},$$

в котором смысл всех переменных совпадает с раньше определенными.

В последнем выражении не определенным есть лишь значение i – номера искаженного символа $\widetilde{\alpha}_i$. Это значение находится из системы неравенств

$$Z \cdot p_i - [Z \cdot p_i] < p_i/p_k, \ (i = 1, 2, ..., n).$$

За искомое значение i принимается номер того неравенства (того p_i), для которого удовлетворяется это условие.

IV Технология формирования контрольных признаков

Формирование контрольных признаков предусматривает использование технологии, адаптивной к длине информационных объектов (текстовых и программных файлов, наборов данных и т. п.), целостность которых контролируется. С этой целью используются переменная длина обобщённых и базовых кодовых слов, зависящая от длины информационных объектов.

При длине информационного объекта, превышающей 255 байтов, в качестве обобщённого кодового слова принимается слово длиной 256 байтов, состоящее из 8 базовых кодовых слов. Т. е., для каждого из этих базовых кодовых слов в соответствии с выражением (1) формируются четырёхбайтовый контрольный признак. Этот контрольный признак вычисляется с использованием n = 32 “рабочих” и r =4 “контрольных” оснований, которые, в свою очередь, выбираются из множества взаимно простых чисел из промежутка [131,...,251], поскольку их разрядность (из условия технологичности программной реализации) должна быть равной 8 битам. Нетрудно убедиться, что количество таких чисел равняется 29. Рабочие основания, исходя из условия обеспечения восьмибитовых условных вычетов, которыми являются символы контролирующего текста, следует выбирать разрядностью большей, чем 8. Т. е. такими основаниями для механизма контроля и восстановления могут быть взаимно простые числа из промежутка [257,...,1579]. Правая граница этого диапазона (число 1579) определена из приведенного прежде условия, чтобы произведение этого числа на ближайшее, меньшее него, взаимно простое число (1571), т. е. произведение двух наибольших рабочих оснований, не превышало бы произведения трех наименьших оснований из числа контрольных. Поскольку последнее произведение равняется произведению 131·137·139, то это и определяет наибольшие основания из рабочих (1571 и 1579):

$$131 \cdot 137 \cdot 139 > 1571 \cdot 1579.$$

При этом количество взаимно простых чисел в диапазоне 257,...,1579 насчитывает 195.

Выше уже отмечено, что контрольный признак обобщённого кодового слова формируются как конкатенация контрольных признаков базовых кодовых слов. При этом длина контрольного признака обобщённого кодового слова составляет 32 байта. Для формирования контрольных признаков текстов лишь для контроля целостности объем избыточной информации сводится к длине в 32 байта и упрощается сама процедура формирования контрольных признаков. Это возможно потому, что сформированная в

соответствии с выше рассмотренной процедурой избыточная совокупность контрольных признаков включает в себя информацию, нужную для определения в дальнейшем наличия искажения, его места и величины. При контроле же лишь целостности достаточно располагать информацией, нужной для определения в дальнейшем только наличия искажения. Поэтому при формировании контрольных признаков наряду с конкатенацией в пределах каждого из обобщённых кодовых слов используется и операция поразрядного сложения по модулю 2 контрольных признаков обобщённых кодовых слов. За счет этого длина контрольного признака (объем избыточной информации) легко сводится к длине, определенной межгосударственным стандартом ГОСТ 34.311–95 для хеш-функций.

При длине информационного объекта, меньшей 255 байтов, этот информационный объект рассматривается в качестве обобщённого кодового слова, состоящего из нескольких базовых кодовых слов. Число этих базовых кодовых слов, их длина n и число контрольных оснований r выбираются так, чтобы длина контрольного признака оставалась не меньшей 32 байтов. С учётом того, что максимальное число "контрольных" оснований равно 29, минимальная длина информационных объектов составляет 2 байта.

Решение о наличии или отсутствии нарушений целостности в данном случае удобно принимать по результатам сравнения контрольных признаков H, вычисленных при кодировании исходного кода и при собственно контроле целостности.

V Имитостойкость данных механизмов контроля целостности

Под имитостойкостью предложенных механизмов контроля целостности информации понимается способность используемых ключевых наборов (наборов p_i) быть нераскрываемыми, а также способность не допускать сокрытия намеренных нарушений целостности информации (имитацию отсутствия нарушений) со стороны неавторизованных пользователей (злоумышленников).

При этом стойкость механизмов контроля целостности информации определяется стойкостью вычислений контрольных признаков, которая зависит от длины выбранных ключей криптозащиты (количества оснований), а также от статистической зависимости начального текста (информационного блока) с его криптографическим отображением.

Под ключом криптозащиты в предложенных механизмах контроля целостности информации понимается набор чисел, которые являются то ли номерами оснований (i), то ли собственно основаниями (p_i). Основания в первом случае выбираются по их номерам из набора простых чисел. При этом формируется ключевой набор, иначе, таким ключевым набором есть набор, который задан в виде совокупности оснований (p_i). Количество этих оснований определяет упомянутую выше длину ключевого набора.

Процесс вычисления контрольных признаков по известным алгоритмам, не по неизвестным ключам или ключевым наборам, **есть по сути криптографическим преобразованием** в контрольный признак информации, целостность которой должна контролироваться. При этом неавторизованный пользователь, не зная ключевого набора, не имеет возможности замаскировать нарушение целостности информации путем формирования такого контрольного признака, который бы скрывал это нарушение.

Примечание 2. Обратим внимание на то, что в предложенных механизмах контроля целостности информации, доступной для анализа неавторизованными пользователями, есть лишь часть закрытой информации – контрольный признак и соответствующая часть (при $n = 32$, $r = 3$ это около 8%) ключевого набора для их формирования (оснований для формирования избыточной информации – контрольных признаков). Основная же часть (для тех же условий – близко 92%) ключевого набора есть недоступной для анализа, поскольку не является представленной в явном виде в результате преобразования информации – в контрольных признаках. Этим самим обеспечивается скрытность ключа или результатов преобразования информации в соответствии с этим ключом. Это связанно с тем, что контрольные признаки, которые формируются, даже простейшие, есть отображением результатов преобразования информации лишь по незначительному количеству элементов ключа. Такое свойство есть не чем иным, как дополнительной возможностью повышения имитостойкости предложенных механизмов контроля целостности информации за счет отсутствия непосредственной статистической связи между первичным текстом и его контрольным признаком, и дает возможность говорить об отсутствии возможности или значительном затруднении выяснения такой статистической связи начального текста (информационного блока) с его криптографическим отображением – контрольным признаком. Последнее, в свою очередь, разрешает говорить о возможности определения ключа только путем прямого перебора и определять стойкость предложенного механизма контроля целостности информации лишь через количество вариантов ключей.

Если при контроле и восстановлении целостности используется n из 195 рабочих и r из 29 контрольных оснований, то общее количество вариантов ключей определяется как произведение количества перестановок (размещений) из 195 элементов по n на количество перестановок (размещений) из 29 элементов по r:

$$N_{EK} = A_{195}^{n} \cdot A_{29}^{r}.$$

При применении этого механизма для контроля целостности последнее ограничение на величины рабочих оснований снимается, поэтому их количество ограничено лишь возможностями процессоров по обработке многобайтовых чисел и есть значительно большим. Например, лишь количество простых чисел, которые являются большими чем 256 и меньшими 6000 превышает 650. Поэтому и количество вариантов есть значительным.

VI Возможности контроля и восстановления целостности информации

Приведенные механизмы контроля, контроля и восстановления целостности информации позволяют разную, в зависимости от потребностей, организацию процедур контроля целостности, контроля и восстановления целостности программных или других информационных объектов (файлов, последовательных наборов данных и т. д.).

Контроль целостности возможен по отношению к любым отдельным информационным объектам, папкам (наборам, директориям), содержащим отдельные информационные объекты, и т. п. Характеристики программной реализации предложенного алгоритма по скорости выполнения операций (ПЭВМ типа Pentium 166 МГц, ОЗУ – 32 Мб; операционная система Windows NT Server 4.0) приведены в табл. 1.

Организация контроля и восстановления целостности информационных объектов имеет некоторые особенности, зависящие от поставленных задач. В частности, можно организовать контроль в частях информационных файлов – обобщенных кодовых словах, длина которых ($\lambda \cdot n$ символов) может быть переменной, например $\lambda = 2, 3, \ldots$(блочно – групповой метод контроля), или

$$\lambda = [N/n]+1,$$

где $N = N_{сф}$, или даже $N = N_{сн}$, где $N_{сф}$ – количество символов в файле (контроль по файлам), $N_{сн}$ – количество символов в информации, которое сохраняется на всем носителе (контроль носителей).

Таблица 1 – Скорость выполнения операций формирования признаков целостности (ФПЦ) и контроля целостности (КЦ)

№ п/п	Тип информацион-ного объекта	Количество файлов в папке	Размер информацион-ного объекта в Мб	Операция	Время выполнения алгоритма в секундах	Скорость формирования ПЦ (в Мб/с)
1	Отдельный файл	1	61,787	ФПЦ	40	1,54
2				КЦ	40	1,54
3	Папка	56	1,144	ФПЦ	2	0,572
4				КЦ	1	1,144
5	Папка	1713	61,849	ФПЦ	116	0,533
6				КЦ	100	0,618

Это позволяет обнаружить нарушение целостности информации в границах *любого из обобщенных кодовых слов* и исправить выявленные в нем искажения, длина которых $\boldsymbol{B}_B$ может быть (при их произвольном расположении в границах обобщенного кодового слова) от одного до

$$\boldsymbol{B}_B = [(\lambda-1)b_c+1]$$

двоичных символов (бит), где b_c – длина символов в контролируемой информации в битах, а λ – глубина перемежения (количество базовых кодовых слов в одном обобщенном). Т. е. наиболее возможная длина исправляемых произвольно расположенных в границах обобщенного кодового слова искажений равняется $(\lambda-1)$ символов. Общее количество исправляемых искажений такой длины равняется количеству обобщенных кодовых слов в составе файла.

Следует отметить, что условия применения этих процедур влияют на выбор метода организации контроля. В условиях контроля процессов сохранения информации следует учитывать то, что:

1. Методы блочно-группового контроля целостности информации позволяют вскрыть ***много групп искажений малой длины***, что, безусловно является очень полезным для ***организации контроля целостности*** тех носителей, на которые искажения имеют естественный, а не искусственный характер. Такими носителями могут быть, например, ***резервные копии программных средств или базы данных*** и т. д.

Но для контроля искажений большей длины, а это присуще искажением искусственного характера, этот вид контроля применять нецелесообразно.

2. Контроль целостности информации в файлах позволяет вскрыть значительно меньшее, чем при предшествующем виде контроля, количество искажений, но ***наибольшей, максимальной при пофайловом контроле длины*** и потому ***этот вид контроля целесообразно применять при контроле целостности информации*** файлов, в случае необходимости такого контроля, например, целостности информации, ***которая дискретно оперативно изменяется.***

При контроле же процессов обмена информацией следует учесть следующее:

1. Методы блочно-группового контроля целостности информации, с их выше определенными особенностями, могут быть легко приспособленными для контроля процессов обмена без использования механизмов решающей обратной связи, что приведёт к повышению скорости обмена (длина блока должна отвечать длине сообщения, принятой в соответствующем протоколе. Это просто реализуется, например, в протоколе X.25).

2. Контроль целостности информации в блоках, длина которых превышает длину сообщения (контроль в файлах), разрешает применять принципы каскадных кодов, и обеспечивать целостность в условиях влияния то ли помех большой продолжительности (количество искажений превышает корректирующие свойства внутреннего кода), то ли продолжительных (в том же понимании) замираний сигналов.

Следует, кроме того, обратить внимание и на то, что вычисление контрольных признаков предложенными методами имеет все признаки, необходимые для хеш–функций и некоторых видов *цифровой подписи*, а следовательно могут в определенных случаях выполнять их функции.

Литература: 1. Нормативный документ Системы технической защиты информации "Общие положения про защиту информации в компьютерных системах от несанкционированного доступа" (НД ТЗИ 1.1 – 002 – 99). 2. Нормативный документ Системы технической защиты информации "Критерии оценки защищенности информации в компьютерных системах от НСД" (НД ТЗИ 2.5 – 004 – 99). 3. Нормативный документ Системы технической защиты информации "Классификация автоматизированных систем и стандартные функциональные профили защищенности обрабатываемой информации от несанкционированного доступа" [НД ТЗИ 2.5.–005 –99]. 4. Василенко В. С., Короленко М. П. Целостность информации в автоматизированных системах. // Корпоративные системы. 1999. – № 3.– с. 52–57. 5. Василенко В. С., Курочкин С. Г. Использование метода помехоустойчивой криптографии в системах обработки кредитно-финансовой информации // Машинная обработка информации. Межведомственный научный сборник.– Вып. 60, 1997. – с. 169–174. 6. Будько М. М., Василенко В. С., Короленко М. П. Механізми контролю цілісності та її поновлення // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні, – К., 2000. – с. 130 – 139.

УДК 519.25

РЕАЛИЗАЦИЯ МЕХАНИЗМОВ ЗАЩИТЫ И АНАЛИЗА ОБРАЗНОГО ОПЕРАЦИОННОГО ЯДРА

Александр Манухин
Служба безопасности Украины

Анотація: **Показана можливість побудови образного операційного ядра на прикладах реалізації механізмів цілісності та автентичності даних.**
Summary: **There is shown possibility of building of figurative operating kernel on examples of realization of mechanisms of wholeness and authenticity of data.**
Ключові слова: **Конфіденційність, цілісність даних, операційна система, образ, функція хешування, кластер.**

I Введение

Относительно информации, как объекта исследования, технологические процессы (субъекты) выступают источником объекта, его потребителем или нарушителем, поскольку на нынешнем этапе развития технологий информационное окружение – это автоматизированные системы (АС) обработки данных. Собственно АС представляет собой совокупность взаимосвязанных организационно-технических компонентов: