(19) Евразийское патентное ведомство

(11) 034935 (13) B1

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ

(45) Дата публикации и выдачи патента
2020.04.08

(21) Номер заявки
201890564

(22) Дата подачи заявки
2016.08.25

(51) Int. Cl. ***G06F 21/31*** (2013.01)
G06F 21/46 (2013.01)

(54) ВЕРИФИКАЦИЯ С ДОПУСКОМ ОШИБОК ДЛЯ ЗАЩИЩЕННЫХ ИДЕНТИФИКАТОРОВ ПРОДУКТА

(31) **62/209,798**
(32) **2015.08.25**
(33) **US**
(43) **2018.08.31**
(86) **PCT/EP2016/070138**
(87) **WO 2017/032861 2017.03.02**

(71)(73) Заявитель и патентовладелец:
ИНЕКСТО СА (CH)

(72) Изобретатель:
Фраде Эрван, Шателен Филипп, Шане Патрик (CH)

(74) Представитель:
Медведев В.Н. (RU)

(56) US-A1-2008066167
EP-A1-2472451
WO-A1-2012136138
"Inside Windows Product Activation", INTERNET CITATION, 1 July 2001 (2001-07-01), XP009130019, Retrieved from the Internet: URL:http://www.licenturion.com/xp/fully-licensed-wpa.txt [retrieved on 2010-03-05] Introduction

(57) Изобретение относится к системе автоматической коррекции, где допускаются некоторые ошибки, в то время как другие распознаются в качестве ошибок. Это применимо, в частности, к немашиночитаемым вводам или к оптическому распознаванию символов при считывании или транскрибировании кодов. При вводе некоторых символов эта система может допускать некоторые комбинации сходных символов и некоторое количество ошибок из такого набора, в то же время по-прежнему сохраняя целостность идентификационных кодов.

034935 B1

034935 B1

Настоящая заявка испрашивает приоритет предварительной заявки США № 62/209798, поданной 25 августа 2015, содержание которой включено в настоящий документ посредством ссылки во всей ее полноте.

Настоящее изобретение относится в целом к методам верификации продукта и автоматической коррекции или допущения определенных наборов ошибок. При попытке верифицировать идентификатор продукта, особенно при использовании немашинных методов считывания, некоторые ошибки возникают с большей вероятностью, чем другие. Эти ошибки часто делаются при вводе вручную символов для кода человеком или с помощью оптического распознавания символов. Настоящее изобретение направлено на решение проблемы, состоящей в том, как преодолеть некоторые типы ошибок, в то же время обеспечивая возможность достаточных уровней аутентификации.

Изобретение относится к системе автоматической коррекции, где допускаются определенные ошибки, в то время как другие распознаются и отклоняются. Это применяется особенно к немашиночитаемым вводам или оптическому распознаванию символов при чтении или транскрибировании кодов. При вводе определенных символов, некоторые символы могут быть похожи на другие, и эта система допускает некоторые комбинации похожих символов и некоторое количество ошибок из такого набора, сохраняя при этом целостность идентификационных кодов.

Строгое соблюдение распознавания символов повышает уровень безопасности в системе аутентификации. Однако, когда люди или оптическое распознавание символов используются для чтения или транскрибирования символов, могут возникать некоторые ошибки. Это может быть связано с проблемами остроты зрения или проблемами разрешения при попытке чтения символов. Для преодоления некоторых ошибок, когда вводятся подобные символы, система должна быть в состоянии распознавать определенные сходные символы, сохраняя при этом целостность и безопасность системы. Допущение ввода подобных символов снижает количество возможных комбинаций при использовании определенного количества символов.

Следующие варианты осуществления настоящего изобретения являются иллюстративными и не предназначены для ограничения объема настоящего изобретения. В то время как описаны один или более вариантов осуществления настоящего изобретения, различные изменения, дополнения, перестановки и их эквиваленты включаются в объем настоящего изобретения. В последующем описании вариантов осуществления даются ссылки на приложенные чертежи, которые составляют часть данного описания, которые показывают в качестве иллюстрации конкретные варианты осуществления заявленного предмета изобретения. Должно быть понятно, что могут быть использованы другие варианты осуществления, и что могут быть выполнены изменения или модификации, такие как структурные изменения. Такие варианты осуществления, изменения или модификации не обязательно являются отклонением от объема относительно предполагаемого заявленного предмета изобретения. В то время как приведенные ниже этапы могут быть представлены в определенном порядке, в некоторых случаях упорядочение может быть изменено таким образом, что определенные вводы предоставляются в другое время или в другом порядке без изменения функций описанных систем и способов. Различные вычисления, которые описаны ниже, такие как находящиеся в рамках процедур инициализации, генерации и аутентификации кода, не требуется выполнять в раскрытом порядке, и другие варианты осуществления с использованием альтернативных порядков вычислений могут быть легко реализованы. В дополнение к переупорядочению, вычисления можно также разбить на подвычисления с теми же результатами.

Варианты осуществления настоящего изобретения далее будут описаны в качестве примера со ссылками на прилагаемые чертежи.

Фиг. 1 иллюстрирует примерную таблицу буквенно-числового кода.

Фиг. 2 иллюстрирует примерный способ для инициализации кода.

Фиг. 3 иллюстрирует примерный способ для генерации кода.

Фиг. 4 иллюстрирует примерный способ для авторизации кода.

В соответствии с вариантом осуществления настоящего изобретения для верификации того, что пользовательский код, введенный в вычислительную систему, соответствует истинному коду, способ содержит: определение набора кодовых символов, которые могут быть использованы в идентификационном коде продукта; определение одного или более поднаборов кодовых символов, причем каждый из поднаборов содержит два или более предварительно определенных эквивалента идентификации набора кодовых символов, каждый поднабор имеет предварительно определенный идентификатор поднабора; определение преобразования для генерации идентификационной информации истинного кода на основе предварительно определенного идентификатора поднабора; прием введенного пользователем кода, содержащий прием набора из одного или более кодовых символов, введенных пользователем с целью верификации продукта; дешифрование введенного пользователем кода путем применения преобразования, чтобы получить идентификационную информацию введенного пользователем кода; генерирование идентификационной информации истинного кода; и верификацию того, что идентификационная информация введенного пользователем кода соответствует идентификационной информации истинного кода.

В альтернативном или дополнительном варианте осуществления способ дополнительно содержит определение истинного кода, содержащего один или более кодовых символов, полученных из идентифи-

кационной информации продукта. В альтернативном или дополнительном варианте осуществления способ дополнительно содержит этап сравнения пользовательского кода и истинного кода. В альтернативном или дополнительном варианте осуществления этап сравнения дополнительно содержит одно или более из определения количества ошибок или типов ошибок, обнаруженных между введенным пользователем кодом и истинным кодом. В альтернативном или дополнительном варианте осуществления одно или более из количества ошибок и типов ошибок, обнаруженных между введенным пользователем кодом и истинным кодом, записываются.

В альтернативном или дополнительном варианте осуществления этап верификации истинного кода содержит верификацию истинного кода только тогда, когда одно или более из количества ошибок и типа ошибок ниже предварительно определенного порога. В альтернативном или дополнительном варианте осуществления все кодовые символы содержатся в пределах одного или более поднаборов. В альтернативном или дополнительном варианте осуществления все поднаборы содержат только два кодовых символа.

В альтернативном или дополнительном варианте осуществления каждая ошибка имеет ассоциированную оценку вероятности, и коррекция одной или более ошибок, обнаруженных во введенном пользователем коде, выполняется на основе оценки вероятности, ассоциированной с одной или более обнаруженными ошибками, и код корректируется, только если оценка вероятности коррекции выше предварительно определенного порога. В альтернативном или дополнительном варианте осуществления на оценку вероятности, ассоциированную с ошибочным введенным пользователем символом, оказывают влияние символы, которые находятся рядом с ошибочным введенным пользователем символом.

В альтернативном или дополнительном варианте осуществления два последовательно введенных пользователем символа обнаруживаются как ошибки и коррекция обоих символов производится перестановкой двух последовательно введенных пользователем символов, обнаруженных как ошибки, приводя в результате к тому, что символы становятся корректными при сравнении с истинным кодом, независимо от того, превосходит ли оценка вероятности коррекции любой ошибки предварительно определенный порог.

В альтернативном или дополнительном варианте осуществления система для верификации того, что пользовательский код, введенный в вычислительную систему, соответствует истинному коду, содержит компьютеризованный процессор, сконфигурированный для исполнения инструкций для: определения набора кодовых символов, которые могут быть использованы в идентификационном коде продукта; определения одного или более поднаборов кодовых символов, причем каждый из поднаборов содержит два или более предварительно определенных эквивалента идентификации из набора кодовых символов, каждый поднабор имеет предварительно определенный идентификатор поднабора; определения преобразования для генерации идентификационной информации истинного кода на основе предварительно определенного идентификатора поднабора; приема введенного пользователем кода, содержащего прием набора из одного или более кодовых символов, введенных пользователем с целью верификации продукта; дешифрования введенного пользователем кода путем применения преобразования, чтобы получить идентификационную информацию введенного пользователем кода; генерации идентификационной информации истинного кода и верификации того, что идентификационная информация введенного пользователем кода соответствует идентификационной информации истинного кода.

В альтернативном или дополнительном варианте осуществления система дополнительно содержит инструкции для определения истинного кода, содержащего один или более кодовых символов, полученных из идентификационной информации продукта.

В альтернативном или дополнительном варианте осуществления система дополнительно содержит инструкции для сравнения пользовательского кода и истинного кода.

В альтернативном или дополнительном варианте осуществления система дополнительно содержит инструкции для определения количества ошибок или типов ошибок, обнаруженных между введенным пользователем кодом и истинным кодом.

В одном варианте осуществления изобретения различные типы ошибок могут быть символами, которые представляются подобными, такими как O, 0 и Q, другой тип ошибки может быть символами, которые находятся в физической близости друг от друга, такими как клавиши символов, окружающие корректный символ. Еще одним примером другого типа ошибки может быть перестановка двух символов, которые находятся рядом друг с другом в коде. Примером может быть случай, где истинным кодом является AZSXDC, а пользователь вводит AZXSDC. Хотя S и X не являются визуально сходными, они находятся в физической близости. Их можно считать как две ошибки типа ошибок физической близости. Альтернативно, так как местоположение предыдущей ошибки и ее тип записываются, это может быть идентифицировано как ошибка перестановки, поскольку обе ошибки связаны с ошибками физической близости рядом друг с другом, и если эта одна коррекция выполнена, то код становится корректным. Также существуют другие примеры, например при использовании не-QWERTY клавиатур, или если ввод требует перехода с одного языка или кодировки на другие, например с ввода двоичного эквивалента шестнадцатеричного кода или перехода от QWERTY (стандартной) клавиатуры на японскую иероглифическую систему записи (кандзи).

В некоторых случаях более чем один вид ошибки может существовать для той же самой клавиши.

Примером в клавиатуре QWERTY являются O и 0, находящиеся в непосредственной физической близости друг от друга, и они часто считаются оптически подобными. Однако O и Q не находятся в непосредственной физической близости, но могут считаться оптически подобными. Таким образом, в способе каждая из этих ошибок может приниматься только как одна, например, O и 0 рассматривается только как оптическая ошибка или ошибка физического местоположения. В другом случае это может быть посчитано за две ошибки, например, если истинный код имеет вид OO0O, а введено O0OO, то это может быть посчитано как две ошибки физического местоположения, или две оптически подобные ошибки, или как одна ошибка перестановки, или как одна ошибка каждого вида ошибок. Способ может иметь предел по общему числу ошибок. Альтернативно он может иметь пределы по каждому виду ошибок. Он также может иметь пределы, как, например, различные комбинации ошибок не подсчитываются, например, если имеется ошибка перестановки, которая также может подсчитываться как две оптические ошибки или две ошибки физического местоположения, он записывает ярлык для всех этих типов, но при подсчете количества ошибок для принятия кода он считает это как одну ошибку перестановки или любым другим способом, который минимизирует количество ошибок. В качестве альтернативы система, для обеспечения максимальной безопасности, может быть установлена так, чтобы вместо этого считать максимально возможное количество ошибок.

Фиг. 1 представляет собой иллюстративный пример буквенно-числового кода для использования в настоящем изобретении. Длина кода, количество символов и типы символов приведены только для примера и не являются ограничительными по показанному количеству или типам.

Для данного изобретения код вводится пользователем или с использованием метода оптического распознавания символов. Затем этот код преобразуется в идентификационную информацию. Идентификационная информация является дешифрируемым кодом, введенным пользователем, например, как если код, который необходимо дешифровать, чтобы получить исходные значения, или пропустить через хэш, или обработать каким-либо другим способом является не точной строкой символов, он будет сравниваться с истинной идентификационной информацией продукта, которая является надлежащим кодом, для которого пользователь пытается получить совпадение. Если не используется обфускация, шифрование, хэширование или другой метод и только сравниваются введенные пользователем символы, то в этом случае введенный пользователем код является идентификационной информацией.

При автокоррекции имеется меньше доступной идентификационной информации, чем кодов, ввиду предварительно определенных поднаборов, которые могут быть использованы для коррекции ошибок. Настоящее изобретение также использует предварительно определенные эквиваленты идентификации, которые, например, включают в себя, без ограничения указанным, буквенно-числовые символы, символы, такие как символы ASCII, и символы других алфавитов, такие как китайские символы, символы хирагана и катакана, кириллица и другие используемые символы.

Для группы (G) из n полей каждое n поле (F) содержит по меньшей мере m отдельных элементов (e), то есть элементы различаются по n полям и являются частью группы элементов E. Элементы могут быть любым буквенно-числовым символом, включая неанглийские буквы и числа, символы ASCII и т.д. Существует функция Size (размер) как число элементов в группе, G(i), где $0<i<n$ является группой, содержащей по меньшей мере m уникальных элементов e, где $e \in E$. F(i,j), где $0<i<n$ и $0<j<Size(G(i))$. (Отметим, что $Size(G(i)) \geq m$.) Свойства этих групп являются такими, как $Size(E) \geq m \cdot n$. Для группы кодов по E длины l запишем ее как C, где $C=E^l$ таково, что такой код определен как элемент C, где код=$\{e_0, e_1, ..., e_{l-1}\}$. Также определим для кода идентификационную информацию как $\{i_0, i_1, ..., i_{l-1}\}$, так что $e_k \in G(i)$ для $0 \leq k<1$, с числовым представлением таким, что идентификационная информация равна $\Sigma i_k \cdot m^k$, где $0 \leq$ идентификационная информация $<m_1$, и определим для кода безопасность (защиту) как $\{j_0, j_1, ..., j_{i-1}\}$, так что $e_k=F(i_k,j_k)$ для $0 \leq k<1$, с числовым представлением безопасности = $\sum_{k=0}^{k<l} j_k \cdot \prod_{\alpha=0}^{\alpha<k} Size(G(i_\alpha))$.

Одним из преимуществ настоящего изобретения является то, что для получения идентификационной информации кода пользователю не нужно знать безопасность, и, следовательно, пользователь может принимать ошибки в чтении или вводе кода для элементов одного и того же поля. Например, если E={A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, 1, 2, 3, 4, 5, 6, 7, 8, 9} и G={{A,R}, {B,8}, {C,G}, {D,O,Q}, {E,4}, {F,P}, {H,N}, {I,1}, {J,Y}, {M,W}, {S,5}, {T,7}, {U,V}, {X,K}, {Z,2}, {L,6}, {3,9}}. В этом примере n=17 и m=2, и если код длины определяется как 12 символов, идентификационная информация в этом отношении представляет собой: 0≤ идентификационная информация $<17^{12}$. 12 используется только в качестве примера, поскольку используемый идентификационный номер имеет длину 12 цифр. Кроме того, определяя размер безопасности как $\prod_{k=0}^{k<12} Size(G(i_k))$ и основываясь на G, как указано выше, размер безопасности для этого набора G равен $2*2*2*2*2*2*2*2*2*2*2*3=2^{11}*3=6144$. Если какое-либо из полей было бы равно четырем символам, например {D,O,0,Q}, то мультипликативный множитель для набора G составлял бы четыре вместо трех в предыдущем уравнении, когда поле было равно только {D,O,Q}. Это позволяет принять больше вариаций ошибок, но одновременно привело бы к снижению безопасности системы. Размер безопасности может быть использован в способе дешифрования символов в оригинал на основе используемого метода,

например, такого как усеченный хэш или другие средства шифрования/дешифрования или расшифровки.

После того как код выбран, этот способ может быть использован для получения идентификационной информации введенного пользователем кода и сравнения его с идентификационной информацией оригинала. Идентификационная информация оригинала представляет собой корректный, без исправления ошибок, набор символов. Даже если идентификационная информация была зашифрована, этот способ исправления ошибок может быть использован. При выполнении алгоритма шифрования каждая из комбинации подобных символов может маркироваться, так что, если один из символов введен и алгоритм выполняется, он может проверить, является ли другой подобный символ корректным символом. Если они являются одним из наборов подобных символов, то он подтверждается в качестве корректного.

Эта система также может быть установлена, чтобы допускать принятие только определенного количества ошибок подобных символов, например, если имеется цепочка из 20 символов, то не более, например, трех символов могут быть введены ошибочно. Также может быть установлено, что если более трех символов в последовательности введены некорректно, если два или более являются тем же самым символом и используется один и тот же некорректный символ, он считается одиночным или не засчитывается как некорректный символ. Например, если пользователь вводит число 5 для символов два, шесть и восемь, система может быть установлена, чтобы принимать код и считать как одну ошибку, когда символ два вводится как 5 вместо S, или как ноль. То же самое может быть сделано, если S вводится для всех трех символов. Альтернативно, если S вводится для символов два и шесть, но 5 вводится для символа восемь, система может снова принять это или посчитать это одной ошибкой. Кроме того, если 5 вводится для символа два и S для символа шесть и восемь, это может быть посчитано как ноль ошибок, одна, две или три ошибки в зависимости от того, насколько строгой должна быть система. Может считаться одной ошибкой то, когда 5 и S были переставлены друг с другом, поэтому в противном случае это правильно, двумя ошибками, например, когда все символы S были переставлены с 5, и все символы 5 были переставлены с S. Альтернативно это может происходить на индивидуальной основе, так что каждый отдельный символ, который был переставлен, является ошибкой, но принимается.

Существуют различные способы и времена, в которые включается способ автокоррекции. Первым является то, когда символы вводятся впервые. Набор может распознавать, что они являются частью набора коррекции ошибок и автоматически устанавливаться, чтобы использовать один из символов, например, из набора {D,O,Q}, чтобы использовать только O, в этом примере O будет репрезентативным символом для определенного поднабора {D,O,Q}. Этот метод имеет то преимущество, что, даже если исправление ошибок завершается раньше, существует меньше вариантов для отдельно используемых символов. Проверка ошибок может быть выполнена автоматически.

Альтернативой является выполнение нормального способа преобразования, следующего от представления символов, через хэширование, дешифрование или любой(е) другой(е) метод(ы) расшифровки, чтобы определить, какова первоначальная нумерация (до первого шифрования числа/буквы), а затем проверку ошибок, чтобы определить, попадают ли символы в один из наборов автокоррекции. Это можно сделать, зная, после того как способ был запущен и число или буква оригинала получены, что упомянутое зашифрованное число для числа оригинала находилось в одном из полей набора G. Этот метод также не требует какой-либо замены. Он позволяет принимать любой из введенных кодов, если символы попадают в предварительно определенные поднаборы.

В качестве альтернативы может быть использован способ, в котором, после того как пользователь вводит код, проверяется, чтобы убедиться, следует ли выполнить какое-либо исправление ошибок до расшифровки кода, посредством хэширования, дешифрования или другого способа. Если один или более символов введены некорректно, т.е. которые попадают в предварительно определенные поднаборы, они могут быть приняты как корректные, и затем код может быть расшифрован и принят как корректный. Это позволяет поддерживать введенный пользователем код, даже если он может не быть истинным идентификационным кодом.

Автокоррекция также может быть сделана после того, как вся строка буквенно-числовых символов введена, но перед их расшифровкой. С помощью этого метода строка символов может быть разделена на предыдущие и последующие сегменты строки на основе местоположения символа, который является частью определенного поднабора, или достаточно высокой оценки вероятности содержания. Новая строка создается с использованием предшествующего сегмента строки, скорректированного символа и последующей строки. Процесс может запускаться итеративно, так что только один символ корректируется каждый раз, при изменении предыдущей и последующей строк. С использованием фиг. 1 в качестве примера, первый раз, когда символ должен быть заменен, соответствует символу числа шесть, поэтому предыдущая строка будет включать в себя символы от одного до пяти и последующая строка будет включать в себя символы от семи до двадцати. Новая строка будет использовать предыдущую строку, скорректированный символ шесть, а затем последующую строку, и исправление ошибок будет продолжаться с проверкой остальных символов и, например, находить, что символ 17 должен быть исправлен автокоррекцией, предыдущая строка будет включать в себя символы от одного до 16 со скорректированным символом шесть, и последующая строка будет включать в себя символы от 18 до 20. В качестве альтернативы несколько предыдущих и последующих сегментов строки может быть создано, если более чем

один символ должен быть скорректирован. Используя пример, приведенный выше, первая предыдущая строка будет представлена символами от одного до пяти, вторая строка будет представлена символами от семи до 16 и конечная последующая строка будет представлена символами от 18 до 20. Символы шесть и 17 будут скорректированы, и затем строка будет восстановлена с использованием нескольких сегментов и исправленных символов.

Замена символов также может быть выполнена вместо использования идентификационной информации кода с использованием их вероятности замещения ошибкой. Оценки вероятности содержания могут быть вычислены или заранее определены. Более высокая оценка вероятности содержания может быть установлена для O, замещаемого на 0, чем O и Q или O и C. Кроме того, на стандартной клавиатуре QWERTY оценки вероятности содержания могут быть установлены для клавиш, окружающих и ближайших к вводимой клавише. Например, на макете клавиатуры QWERTY буква D окружена буквами W, E, R, S, F, X, C и V. Каждая из них может иметь оценку вероятности содержания, ассоциированную с ними, и они могут быть разными, например C может считаться оптически более похожей на D, чем на W или R, и как таковая может быть установлена, чтобы иметь более высокую оценку вероятности содержания для замещения. Некоторые клавиши могут иметь несколько оценок вероятности содержания, ассоциированных с ними. Например, одну оценку для физической близости и другую для сходства символов. Эти оценки могут быть определены по-разному для разных клавиатур, например для сканеров штрихкодов, которые имеют клавиатуру, или клавиатуры смартфона, которая отличается от настройки QWERTY, и т.д. Оценки могут быть объединены нелинейным образом, принимая во внимание как местоположение клавиши, так и оптическое сходство клавиш, чтобы увеличить оценку, или игнорировать физическое местоположение, основываясь на оптическом сходстве буквы, или устанавливаться, чтобы только суммироваться с оценкой вероятности содержания. Клавиши могут также иметь различные оценки для заглавных букв по сравнению со строчными буквами или цифрами на клавишах. Когда оценки вероятности содержания используются, пороговая оценка может устанавливаться системой, чтобы выполнять определенные замены.

Оценка вероятности содержания также может быть назначена строке символов вместо или в дополнение к назначению на индивидуальной посимвольной основе. Если некоторые комбинации символов определяются как более вероятные для ввода, то они могут контролироваться, и если они будут обнаружены, то может выполняться их автокоррекция. Например, строке "TEH" может быть назначена высокая вероятность того, что она представляет собой "THE" и, следовательно, если распознана предыдущая строка, то выполняется коррекция или принятие ее в качестве последней строки. Это может также работать для числовых элементов, например, если используется двоичная последовательность и пользователь вводит "01007011" вместо "01001011", то система снова может исправлять ее на последнюю форму с более высокой оценкой вероятности содержания.

Графики состояний также могут быть использованы для исправления ошибок. Это особенно полезно, когда используются полные слова или фразы, и оценка содержания вычисляется для каждого слова и для всей фразы в целом. Если определенное слово имеет низкую оценку содержания и коррекция этого слова привела бы к высокой оценке содержания, то слово может быть исправлено, например, заменой "too" на "to" или "two" в зависимости от увеличенной оценки содержания фразы и слова. Это может быть сделано по сходству буквенно-цифровых строк или по словам и т.д.

В одном варианте осуществления может вводиться одиночный символ. В этом варианте осуществления может быть предпочтительным уменьшить вероятность содержания ввода ошибки или запретить замещение символов совсем, если это желательно. Это также может быть достигнуто путем привязки количества допустимых ошибок к длине количества вводимых символов, например, количество ошибок для исправления меньше, чем количество введенных символов. Это может быть изменено обычными способами принятия большого или меньшего количества исправлений, от полного введенного количества вплоть до нуля, в зависимости от вводимого количества или любого другого нормального алгоритма, который можно использовать, чтобы определить, сколько ошибок может быть или будет исправлено.

Примерные исполняемые компьютером инструкции для выполнения предлагаемых способов, представлены ниже.

```
using System;
using System.Collections.Generic;
```

```
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApplication1
{
class Program
{
static string[] Groups=new[] { "AR",
"B8",
"CG",
"DOQ",
"E4",
"FP",
"HN",
"I1",
"JY",
"MW",
"S5",
"T7",
"UV",
"XK",
"Z2",
"L6",
"39"};
static int numberOfGroups=Groups.Length;
static void Main(string[] args)
{
//codeGenID
int id=122334;
int noise=SecurityFunction(id);
var gr=new int[12];
//Convert your ID to base numberOfGroups (17)
for (int i=0; i < 12; i++)
{
gr[i]=(id % numberOfGroups);
id=id/numberOfGroups;
```

```
    }
    // Calculate the total size of the security (Noise) Number
of Elements in each selected group
    int noiseSize=1;
    for (int i=0; i < 12; i++)
    {
    noiseSize=noiseSize * Groups[gr[i]].Length;
    }
    // Truncate the generated security with the available one
in the code
    var usedNoise=noise % noiseSize;
    // Create the code by selecting the element in the group
based on the security
    StringBuilder stb=new StringBuilder(12);
    for (int i=0; i < 12; i++)
    {
    stb.Append(Groups[gr[i]][noise % Groups[gr[i]].Length]);
    noise=noise/Groups[gr[i]].Length;
    }
    Console.WriteLine("Code=" +stb.ToString());
    Console.WriteLine("Enter User Code");
    string code=Console.ReadLine();
    id=0;
    usedNoise=0;
    int carryOver=0;
    // Extracting the id and the user entered security from the
user entered code
    for (int i=11; i >= 0; i--)
    {
    var ch=code[i];
    for (int j=0; j < numberOfGroups; j++)
    {
    var index=Groups[j].IndexOf(ch);
    if (index >= 0)
    {
    id=id * numberOfGroups+j;
```

```
usedNoise=usedNoise * carryOver+index;
carryOver=Groups[j].Length;
}
}
}
Console.WriteLine("Id="+id);
Console.WriteLine("Compare      Security="+(usedNoise      ==
(SecurityFunction(id) % noiseSize)));
Console.WriteLine("Enter to exit");
Console.ReadLine();
}
private static int SecurityFunction(int id)
{
return 345;
}
}
}
```

Интеграция с защищенными системами производства

Описанные выше системы и способы для верификации могут быть использованы в комбинации с системами для генерации защищенных идентификаторов для использования с производством.

Как используется в настоящем документе, объект может относиться к: i) лицу, такому как потребитель продукта; ii) группе, такой как группа, имеющая общий интерес, например розничные торговцы; iii) вычислительному устройству; iv) вычислительному узлу в сетевой системе; v) месту хранения, такому как ячейка хранения памяти, хранящая документ; vi) виртуальной точке в сети, например, представляющей бизнес-функцию в рамках коммерческого предприятия, и т.п. Кроме того, объект может представлять собой точку в технологическом потоке, например для авторизации, которая может выполняться лицом, ответственным за этот аспект технологического потока, или вычислительным устройством, которое обеспечивает автоматизированную обработку. Термин "объект" не предназначается для ограничения каким-либо одним из этих примеров и может распространяться на другие ситуации в соответствии с концепциями, описанными в настоящем документе.

Модуль контроля

Модуль управления (110), см. фиг. 2, также известный как "Orchestrator" (координатор), может принимать ввод от любого из других модулей или внешних источников и может предоставлять инструкции на другие модули в системе, основываясь на предварительно сконфигурированных программах и/или вводах оператора. Он также может генерировать сводку состояния системы на приборной панели.

Ввод в модуль управления может включать в себя любые или все данные (105) конфигурации. Вводимые данные конфигурации могут указывать любой или все параметры, включая, без ограничения указанным, машину для производства, производственную линию, завод, производимый продукт и объем продукта. Данные конфигурации могут указывать, какие элементы (например, продукты) должны быть маркированы защищенными идентификаторами и каким образом эти элементы могут производиться. Данные конфигурации могут указывать номенклатуру (диапазон, серию) продукции, как, например, начальный и конечный идентификаторы продуктов. В некоторых вариантах осуществления номенклатура может представлять собой набор идентификаторов продуктов. Данные конфигурации могут быть предоставлены оператором системы или могут генерироваться динамически или автоматически. Данные конфигурации могут включать в себя дополнительные исполняемые инструкции или интерпретируемый алгоритм. Данные конфигурации могут быть основаны на вводе оператора или выводе системы управления производством или другой централизованной системы для инструктирования, что и как производить.

Модуль управления (110) может передавать данные конфигурации на любой модуль, в том числе, без ограничения указанным, на модуль авторизации (130), модуль идентификации (140) и модуль подписи (145).

Модуль управления может запросить авторизацию от модуля авторизации для выполнения операции производства. Этот процесс включает в себя передачу запроса (включая некоторые или все данные конфигурации) на модуль авторизации и прием подписанных или зашифрованных данных конфигурации. В некоторых вариантах осуществления модуль авторизации может возвращать данные конфигурации на модуль управления, включая цифровую подпись, применяемую к этим данным конфигурации. Модуль авторизации определяет, следует ли авторизовать запрос от модуля управления, на основе данных, которые он принимает. Кроме того, информация, возвращаемая модулем авторизации, включенная

в данные конфигурации, может быть использована для связывания генерируемых кодов с обеспечиваемой авторизацией. Поскольку данные подписаны модулем авторизации, в системе может быть предотвращено модифицирование данных конфигурации. В качестве неограничивающего примера модификация запроса на производство одной марки (бренда) вместо другой может контролироваться, разрешаться или отклоняться.

Авторизации, принятые от модуля авторизации, также могут передаваться на модуль верификации, так что запросы верификации могут затем обрабатываться в отношении этих авторизаций. Данные, передаваемые на модуль верификации, могут включать в себя защищенный идентификатор, а также любые из данных конфигураций. В некоторых примерах данные конфигурации, отправленные на модуль авторизации, могут включать в себя информацию о номенклатуре продуктов.

Подписанные или подтвержденные данные конфигурации могут быть некоторыми или всеми из набора параметров ввода модуля управления, верифицированных и подтвержденных модулем авторизации, которые остаются в силе в течение производства. Маркер безопасности может быть выводом из модуля авторизации и/или параметром ввода модуля управления. Маркер безопасности может быть доказательством того, что идентификатор продукта соответствует подтвержденным данным конфигурации и, следовательно, авторизованному производству. Маркер безопасности может быть вводом в модуль подписи, чтобы генерировать подпись для идентификатора одного продукта, или подпись одного идентификатора продукта, или самого идентификатора продукта, или номенклатуры продуктов или идентификаторов продуктов. Маркер безопасности может быть уникальным кодом, случайным кодом или псевдослучайным кодом. Маркер безопасности может быть любым числовым, или буквенным, или комбинацией числовых и буквенных символов.

Модуль авторизации

Модуль авторизации работает, чтобы подтверждать запросы на авторизацию для выполнения действия в системе идентификации. В некоторых вариантах осуществления он может работать в качестве менеджера лицензий.

Модуль авторизации может принимать данные конфигурации. Модуль авторизации может также принимать информацию о номенклатуре и/или алгоритме. В некоторых вариантах осуществления модуль авторизации может получать введенные данные конфигурации от модуля управления. Выводимая номенклатура может опционально определять номенклатуру продуктов, машины, предприятия, серии или объемы продуктов, которые авторизованы. Вывод может также включать в себя информацию о номенклатуре и/или включать в себя алгоритм, который содержит набор исполняемых или интерпретируемых инструкций, которые будут использоваться для генерации маркеров безопасности. Модуль авторизации может быть централизованным на уровне завода или децентрализованным на каждой производственной линии или комбинацией того и другого.

Модуль авторизации может хранить и/или генерировать один или более ключей шифрования. В некоторых вариантах осуществления ключ, сохраненный в модуле авторизации, может быть частным открытым ключом шифрования согласно инфраструктуре открытого ключа (PKI). В некоторых вариантах осуществления модуль авторизации хранит единственную копию частного ключа. В других вариантах осуществления модуль авторизации распределен по нескольким объектам, которые копируют ключи между собой. В случае PKI модуль авторизации может выводить подписанные данные конфигурации. В некоторых вариантах осуществления модуль авторизации может шифровать данные конфигурации и/или подписывать вывод данных конфигурации.

В некоторых вариантах осуществления система сконфигурирована таким образом, что только модуль авторизации может считывать защищенные параметры ввода модуля управления, необходимые для генерации маркеров безопасности. В некоторых вариантах осуществления ключ предоставляется в модуль авторизации из другого источника.

Модуль авторизации может быть выполнен в виде аппаратного защищенного модуля (HSM) или другого типа физического вычислительного устройства, которое обеспечивает защиту и управляет цифровыми ключами для надежной аутентификации и обеспечения криптографической обработки. Функциональность модуля авторизации может выполняться компьютером со встроенной платой с ключом шифрования или частным ключом PKI. Модуль может быть оснащен такими функциями, что попытки получить доступ к данным могут привести к тому, что он становится нечитаемым или недоступным.

Если ввод в модуль авторизации является номенклатурой и алгоритмом, то модуль авторизации может выводить идентификационную информацию в диапазоне авторизации и маркеры безопасности идентификатора. Например, выводимая идентификационная информация может быть в диапазоне от 0 до 1000 с маркером безопасности для каждого элемента в диапазоне.

Модуль авторизации может генерировать ключ из любого параметра, используемого в модуле управления. В некоторых вариантах осуществления модуль авторизации может генерировать или выводить ключ из существующего ключа из любого параметра, используемого в модуле управления таким образом, что только конкретный модуль авторизации может использовать этот ключ. Оборудование и программное обеспечение, реализующие этот метод открытого ключа, могут быть воплощены в асимметричной криптосистеме.

Выводом модуля авторизации может быть информация, такая как данные конфигурации и, опционально, один или более маркеров безопасности, с цифровой подписью, предоставленной модулем подписи. Альтернативно выводом модуля авторизации могут быть данные конфигурации, зашифрованные ключом, хранящимся в модуле авторизации. Вывод модуля авторизации может быть предоставлен на модуль управления.

В соответствии с вариантом осуществления способ аутентификации производства продуктов включает в себя хранение в электронном виде данных конфигурации для производственного цикла, причем данные конфигурации для производственного цикла определяют параметры, используемые в производстве продуктов; определение, авторизованы ли данные конфигурации для производственного цикла; если производственный цикл авторизован, генерирование маркеров защиты и ассоциирование маркеров с данными конфигурации; и снабжение цифровой подписью данных конфигурации путем генерации цифровой подписи и ассоциирование цифровой подписи с данными конфигурации; прием снабженных цифровой подписью данных конфигурации и цифровой подписи в производственной машине; в производственной машине верификацию цифровой подписи, ассоциированной со снабженными цифровой подписью данными конфигурации; вычисление набора защищенных идентификаторов продуктов на основе снабженных цифровой подписью данных конфигурации; производство продуктов в производственном цикле в соответствии со снабженными цифровой подписью данными конфигурации; и печать набора защищенных идентификаторов продуктов на продуктах в соответствии со снабженными цифровой подписью данными конфигурации.

В альтернативном или дополнительном варианте осуществления данные конфигурации представляют номенклатуру продуктов, которые должны производиться. В альтернативном или дополнительном варианте осуществления данные конфигурации представляют номенклатуру продуктов, машины, предприятия, диапазоны или объемы продуктов, которые авторизованы. Альтернативные или дополнительные варианты осуществления могут включать в себя прием запроса на верификацию, причем запрос содержит идентификатор продукта и определение, авторизованы ли данные конфигурации для производственного цикла, посредством ссылки на менеджер лицензий. Альтернативные или дополнительные варианты осуществления могут включать в себя генерирование маркера безопасности для номенклатуры продуктов и ассоциирование маркера безопасности с номенклатурой продуктов.

Модуль подписи

Модуль подписи, см. фиг. 2-4, может принимать данные конфигурации, ключ авторизации, маркер безопасности или любую их комбинацию так же, как и уникальный идентификатор продукта, сгенерированный модулем идентификации. В некоторых вариантах осуществления модуль подписи может принимать, кроме того, одну или более собственных характеристик машины, и/или продукта, и/или характеристик элемента продукта. Модуль подписи может создать цифровую подпись на основе любых или всех из этих вводов, в общем упоминаемых здесь как данные конфигурации.

Для генерации цифровой подписи в некоторых вариантах осуществления модуль подписи может сначала сгенерировать дайджест или другое представление данных конфигурации. В некоторых вариантах осуществления дайджест может быть сгенерирован путем вычисления криптографического хэш-значения данных конфигурации в соответствии с алгоритмом цифровой подписи, предоставленным модулем подписи, выполняющим алгоритм цифровой подписи. В качестве неограничивающих примеров хэш может быть вычислен в соответствии с функциями MD5, SHA-1, SHA-2, SHA-3/Keccak. Дайджест может быть зашифрован с использованием частного ключа, полученного с помощью модуля подписи для генерации цифровой подписи.

В некоторых вариантах осуществления цифровая подпись может использовать технологию инфраструктуры открытого ключа (PKI), чтобы установить аутентичность данных конфигурации. Системы PKI используют сертификаты и ключи для идентификации объектов, физических лиц или организаций. Модуль аутентификации использует частный ключ, чтобы подписывать данные конфигурации, и ассоциирует данные конфигурации с сертификатом, включая открытый ключ, используемый в модуле аутентификации.

Модуль получателя использует открытый ключ для верификации цифровой подписи и, таким образом, аутентичности подписанных данных конфигурации. Поддерживающие технологии могут быть использованы для создания других характеристик защиты от неправомерных отречений (отказа от авторства), как, например, время подписания и статус ключей подписи. Открытый ключ может быть предоставлен объекту-получателю непосредственно либо путем публикации в онлайн хранилище или каталоге.

Модуль идентификации

Модуль идентификации может принимать данные конфигурации и генерировать идентификаторы для элементов, подлежащих маркировке. Модуль идентификации может принимать цифровую подпись, сгенерированную модулем подписи, которая будет объединена с уникальным идентификатором, чтобы генерировать составной уникальный идентификатор.

Идентификаторы могут включать в себя или быть основаны на дате и/или времени производства продукта, подлежащего маркировке, и цифровой подписи, принятой от модуля подписи. В некоторых вариантах осуществления генерируемые защищенные идентификаторы могут быть уникальными или, по

существу, уникальными. В некоторых вариантах осуществления защищенные идентификаторы могут быть маркерами безопасности.

В случае диапазонов модуль идентификации может генерировать идентификатор диапазона и набор идентификаторов в пределах генерируемого диапазона.

Созданные идентификаторы могут быть выведены в модуль управления печатью для прямой печати на продукт или могут быть введены в дальнейшую обработку, чтобы генерировать другой код, который печатается на упаковке продукта.

Модуль верификации

Модуль верификации (150), см. фиг. 4, может быть сконфигурирован, чтобы использовать усовершенствованные методы верификации, описанные выше. Модуль верификации может быть дополнительно сконфигурирован, чтобы принимать верифицированные данные конфигурации и, основываясь на подтвержденных данных конфигурации, подтверждать запрос на авторизацию (305) для сообщенных данных для предприятия, машины, продукта или объема производства. Вводы в модуль верификации могут включать в себя некоторые или все из верифицированных данных конфигурации, вывода из модуля подписи, идентификаторов, маркеров безопасности и/или информации о номенклатуре (диапазоне). Модуль верификации может генерировать информацию для модуля авторизации с этими параметрами с целью верификации/подтверждения идентификатора продукта.

Модуль верификации может генерировать дешифрование (320) запроса, что включает в себя один или более идентификаторов или диапазоны идентификаторов (315) и данные подписи (310), включающие в себя один или более маркеров безопасности.

Если маркер безопасности введен в модуль верификации, модуль верификации может возвратить информацию, относящуюся к авторизации, данным конфигурации и/или диапазонам. Если один маркер безопасности используется для номенклатуры продуктов, маркер безопасности может быть предоставлен модулю верификации для верификации параметров, ассоциированных с номенклатурой продуктов, а не с отдельными продуктами. Этот вариант осуществления может быть особенно полезным в контексте регулирования экспорта.

Системные процессы

Инициализация кода идентификации

Инициализация кода идентификации может быть выполнена для подтверждения авторизации и параметров. В некоторых вариантах осуществления, по соображениям производительности, это может быть выполнено один раз в начале производства. Модуль управления (110), см. фиг. 2, может получить доступ к хранилищу данных (115) для дополнительных параметров, или дополнительные параметры могут быть предоставлены в модуль. Параметры и данные конфигурации, после того как подписаны модулем авторизации (130), формируют подтвержденные данные конфигурации (135). Модуль управления получает верифицированные данные конфигурации, как описано выше, в ответ на его запрос в модуль авторизации (130).

Авторизация может представлять собой авторизацию на производство продукта, или на маркировку продукта определенным ID, или и то и другое. Данные конфигурации и дополнительные параметры передаются в модуль авторизации и используются модулем авторизации для генерации маркеров безопасности. Модуль авторизации может подписать данные конфигурации и дополнительные параметры, формируя подписанные данные конфигурации. Как обсуждалось выше, данные конфигурации могут указывать определенный производственный цикл или другие продукты и действия. Модуль авторизации может генерировать блок авторизации, включая ключ, авторизованные идентификаторы и маркеры безопасности. В некоторых вариантах осуществления ключ может генерироваться модулем авторизации или может предоставляться на него. Модуль авторизации может передавать блок авторизации в модуль управления. Модуль управления может передавать подтвержденные данные конфигурации и другую информацию, например список идентификаторов, диапазон идентификаторов и/или один или более маркеров безопасности, на модуль подписи (145). Модуль подписи может подписывать данные и отправлять подписанные данные и подпись на модуль управления. Модуль идентификации (140) может затем принимать от модуля управления блок инициализации, включая идентификаторы и/или диапазоны идентификаторов для продуктов.

Вариант осуществления настоящего изобретения может включать в себя способ для инициализации процесса для безопасного управления производственным оборудованием, включающего в себя прием электронным образом данных конфигурации из электронного хранилища данных; сохранение электронным образом данных конфигурации для производственного цикла, причем данные конфигурации для производственного цикла задают параметры, используемые в производстве продуктов; передачу данных конфигурации на модуль авторизации; в модуле авторизации: определение, авторизован ли производственный цикл; генерирование подтвержденных данных конфигурации, содержащих ключ, представление множества авторизованных идентификаторов продуктов и маркер безопасности; передачу подтвержденных данных конфигурации на модуль подписи; и в модуле подписи: снабжение подписью подтвержденных данных конфигурации.

Альтернативные или дополнительные варианты осуществления могут включать в себя определение,

являются ли данные конфигурации для производственного цикла авторизованными; если производственный цикл авторизован, генерирование маркера защиты и ассоциирование маркера с данными конфигурации; и снабжение цифровой подписью данных конфигурации путем генерирования цифровой подписи и ассоциирования цифровой подписи с данными конфигурации.

Альтернативные или дополнительные варианты осуществления могут включать в себя прием снабженных цифровой подписью данных конфигурации и цифровой подписи на производственной машине; на производственной машине верификацию цифровой подписи, ассоциированной со снабженными цифровой подписью данными конфигурации; и вычисление набора защищенных идентификаторов продуктов на основе снабженных цифровой подписью данных конфигурации.

Альтернативные или дополнительные варианты осуществления могут включать в себя производство продуктов в производственном цикле в соответствии со снабженными цифровой подписью данными конфигурации; и печать набора защищенных идентификаторов продуктов на продукты в соответствии со снабженными цифровой подписью данными конфигурации.

Альтернативные или дополнительные варианты осуществления могут включать в себя определение, авторизован ли производственный цикл, дополнительно содержащее извлечение данных лицензирования из сервера лицензирования.

Генерация идентификационного кода

Процесс генерации кода, см. фиг. 3, генерирует коды во время производственного процесса. Процесс генерации идентификационного кода может начинаться с запроса на модуль идентификации (130) о получении идентификатора или диапазона идентификаторов, которые затем возвращаются на модуль управления (110). Эти идентификаторы затем отправляются на модуль подписи (145), который подписывает идентификаторы и возвращает подписанные идентификаторы на модуль управления. Модуль подписи может принимать маркер безопасности. В некоторых вариантах осуществления модуль подписи не нуждается в управлении внешними инструкциями, и если какой-либо идентификационный код должен считаться, то код может быть связан с одним маркером безопасности. Модуль подписи может управляться модулем авторизации. Модуль управления может затем отправить выходные данные для управления печатью в модуль принтера (210). Выходные данные, отправляемые на управление печатью, могут быть зашифрованы перед передачей. Данные конфигурации могут передаваться на модуль верификации (150) для обработки последующих запросов верификации.

Вариант осуществления настоящего изобретения включает в себя способ для генерации кода для защищенной идентификации продуктов, производимых на производственном оборудовании, включающий в себя прием электронным образом данных конфигурации из электронного хранилища данных; сохранение электронным образом данных конфигурации для производственного цикла, причем данные конфигурации для производственного цикла задают параметры, используемые в производстве продуктов; передачу данных конфигурации на модуль авторизации; в модуле авторизации: определение, авторизован ли производственный цикл; генерирование подтвержденных данных конфигурации, содержащих ключ, представление множества авторизованных идентификаторов продуктов и маркер безопасности; передачу подтвержденных данных конфигурации на модуль подписи; в модуле подписи: снабжение цифровой подписью подтвержденных данных конфигурации; в модуле идентификации: прием запроса на идентификатор продукта и генерацию идентификатора продукта в ответ на запрос; передачу идентификатора продукта от модуля идентификации на модуль подписи; снабжение цифровой подписью идентификатора продукта в модуле подписи и передачу снабженного цифровой подписью идентификатора продукта на модуль принтера.

Альтернативные или дополнительные варианты осуществления могут включать в себя прием электронным образом данных конфигурации из электронного хранилища данных; сохранение электронным образом данных конфигурации для производственного цикла, причем данные конфигурации для производственного цикла задают параметры, используемые в производстве продуктов; передачу данных конфигурации на модуль авторизации; в модуле авторизации: определение, авторизован ли производственный цикл; генерирование подтвержденных данных конфигурации, содержащих ключ, представление множества авторизованных идентификаторов продуктов и маркер безопасности; передачу подтвержденных данных конфигурации на модуль подписи; в модуле подписи: снабжение цифровой подписью подтвержденных данных конфигурации.

В альтернативных или дополнительных вариантах осуществления запрос осуществляется для диапазона идентификаторов. Альтернативные или дополнительные варианты осуществления могут включать в себя определение, авторизованы ли данные конфигурации для производственного цикла; если производственный цикл авторизован, генерирование маркера защиты и ассоциирование маркера с данными конфигурации; и снабжение цифровой подписью данных конфигурации путем генерации цифровой подписи и ассоциирования цифровой подписи с данными конфигурации.

Верификация идентификационного кода

Как описано выше, модуль верификации (рассматриваемый здесь в единственном числе как последовательные или параллельные отношения нескольких логических или физических модулей верификации) может принять запрос на верификацию. Запрос может включать в себя один или более идентифика-

ционных кодов. Модуль верификации может расшифровать или иным образом демаскировать принятый идентификационный код. Полученная информация, будучи расшифрованной, может включать в себя компоненты подписи и идентификатор. Полученный идентификатор может быть тогда связан с исходными данными конфигурации, ранее сохраненными в ассоциации с идентификатором. Связанные данные могут включать в себя другие идентификаторы в диапазоне, маркер безопасности и другую информацию, сохраненную в связи с производством продукта, несущего этот идентификационный код.

Некоторые варианты осуществления могут включать в себя дополнительные функциональные возможности для обработки идентификаторов, которые предоставляются на модуль верификации на основании стороны, запрашивающей верификацию кода. Различным сторонам могут быть предоставлены различные средства для доступа к модулю верификации. Например, розничному торговцу или иному торговцу может быть предоставлен другой портал или канал связи, чем потребителю. Розничному торговцу также может потребоваться аутентифицировать себя по отношению к модулю верификации.

В некоторых вариантах осуществления система может быть сконфигурирована таким образом, что верификация потребителем приводит в результате к маркировке идентификатора как верифицированного. Система может быть дополнительно сконфигурирована так, чтобы сохранять те коды, для которых верификация запрошена потребителем.

Любые последующие запросы на верификацию тех уже верифицированных кодов могут быть отклонены или в противном случае быть обработаны иным образом.

Функции экспорта

Варианты осуществления настоящего изобретения могут быть применены в контексте экспорта кода к третьим сторонам. Эти варианты осуществления могут включать в себя функцию экспорта, сконфигурированную, чтобы генерировать отдельный код для этой цели. Экспортируемый код может генерироваться путем сбора одного или более идентификаторов продуктов и/или маркеров защиты и снабжения подписью этих идентификаторов и/или маркеров. Идентификаторы и/или маркеры могут быть собраны в любой момент в процессе производства. Подписанные идентификаторы и/или маркеры в форме экспортируемых кодов могут быть предоставлены третьей стороне, которая может хранить их и выполнять верификацию действительности этих идентификаторов и/или маркеров.

Архитектура системы

Системы и способы, описанные здесь, могут быть реализованы в программном обеспечении, или аппаратных средствах, или в любой их комбинации. Системы и способы, описанные здесь, могут быть реализованы с использованием одного или более вычислительных устройств, которые могут или не могут быть физически или логически отделены друг от друга. Кроме того, различные аспекты способов, описанных в данном документе, могут быть объединены или включены в другие функции. В некоторых вариантах осуществления проиллюстрированные элементы системы могут быть объединены в единое аппаратное устройство или разделены на несколько аппаратных устройств. Если используется несколько аппаратных устройств, аппаратные устройства могут быть физически расположены вблизи или удаленно друг от друга.

Способы могут быть реализованы в виде компьютерного программного продукта, доступного из используемого компьютером или считываемого компьютером носителя хранения, который обеспечивает программный код для использования посредством или в соединении с компьютером или любой системой выполнения инструкций. Используемый компьютером или считываемый компьютером носитель хранения может быть любым устройством, которое может содержать или хранить программу для использования посредством или в соединении с компьютером или системой, устройством или прибором для выполнения инструкций.

Система обработки данных, подходящая для хранения и/или выполнения соответствующего программного кода, может включать в себя по меньшей мере один процессор, связанный прямо или косвенно с компьютеризованным устройством хранения данных, таким как элементы памяти. Устройства ввода/вывода (I/O) (включая, без ограничения указанным, клавиатуры, дисплеи, указательные устройства и т.д.) могут быть связаны с системой. Сетевые адаптеры могут быть также связаны с системой, чтобы система обработки данных связывалась с другими системами обработки данных, или удаленными принтерами, или устройствами хранения данных через промежуточные частные или общедоступные сети. Для обеспечения взаимодействия с пользователем признаки могут быть реализованы на компьютере с устройством отображения, таким как CRT (электронно-лучевая трубка), LCD (жидкокристаллический дисплей) или другой тип монитора для отображения информации пользователю, и клавиатурой и устройством ввода, таким как мышь или трекбол, с помощью которого пользователь может обеспечить ввод данных в компьютер.

Компьютерная программа может представлять собой набор инструкций, которые могут быть использованы, прямо или косвенно, в компьютере. Системы и способы, описанные здесь, могут быть реализованы с использованием языков программирования, таких как Flash™, Java™, C++, C, C#, Visual Basi™, JavaScript™, PHP, XML, HTML и т.д., или комбинации языков программирования, включая компилируемые и интерпретируемые языки, и могут быть развернуты в любой форме, в том числе как от-

дельная программа или как модуль, компонент, подпрограмма или другой блок, пригодный для использования в вычислительной среде. Программное обеспечение может включать в себя без ограничения указанным встроенное программное обеспечение, резидентное программное обеспечение, микрокод и т.д. Протоколы, такие как SOAP/HTTP, могут быть использованы при реализации интерфейсов между модулями программирования. Компоненты и функциональные возможности, описанные здесь, могут быть реализованы на любой операционной системе рабочего стола с исполнением в виртуализованной или невиртуализованной среде с использованием любого языка программирования, подходящего для разработки программного обеспечения, включая, без ограничения указанным, различные версии Microsoft Windows™, Apple™ Mac™, iOS™, Uni™/X-Windows™, Linux™ и т.д.

Подходящие процессоры для выполнения программы инструкций включают без ограничения указанным микропроцессоры общего и специального назначения и единственный процессор или один из нескольких процессоров или ядер любого вида компьютера. Процессор может получать и хранить инструкции и данные из компьютеризованного устройства хранения данных, такого как постоянная память, память с произвольным доступом или обе, или любая комбинации устройств хранения данных, описанных в настоящем документе. Процессор может включать в себя любую схему обработки или схему управления, выполненные с возможностью управления операциями и рабочими характеристиками электронного устройства.

Процессор может также включать в себя или быть операционно связанным, чтобы осуществлять связь с одним или более устройствами хранения данных для хранения данных. Такие устройства хранения данных могут включать в себя в качестве неограничивающих примеров магнитные диски (включая внутренние жесткие диски и съемные диски), магнитооптические диски, оптические диски, постоянную память, память с произвольным доступом и/или флэш-память. Устройства хранения, подходящие для материального воплощения инструкций компьютерной программы и данных, могут также включать в себя все формы энергонезависимой памяти, в том числе, например, полупроводниковые устройства памяти, такие как EPROM, EEPROM и устройства флэш-памяти; магнитные диски, такие как внутренние жесткие диски и съемные диски, магнитооптические диски и диски CD-ROM и DVD-ROM. Процессор и память могут дополняться или быть включены в ASIC (специализированные интегральные схемы).

Системы, модули и способы, описанные в настоящем документе, могут быть реализованы с использованием любой комбинации элементов программного обеспечения или аппаратных средств. Системы, модули и способы, описанные здесь, могут быть реализованы с использованием одной или более виртуальных машин, работающих по отдельности или в сочетании друг с другом. Любое применимое решение виртуализации может быть использовано для инкапсулирования физической платформы вычислительной машины в виртуальную машину, которая выполняется под управлением программного обеспечения виртуализации, работающего на вычислительной платформе аппаратных средств или хосте. Виртуальная машина может иметь как аппаратные средства виртуальной системы, так и программное обеспечение гостевой операционной системы.

Системы и способы, описанные здесь, могут быть реализованы в компьютерной системе, которая включает в себя компонент внутреннего интерфейса, такой как сервер данных, или которая включает в себя компонент промежуточного программного обеспечения, такой как сервер приложений или Интернет-сервер, или которая включает в себя компонент внешнего интерфейса, такой как клиентский компьютер, имеющий графический пользовательский интерфейс, или Интернет-браузер, или любую их комбинацию. Компоненты систем могут быть соединены с помощью любой формы или среды передачи цифровых данных, такой как сеть связи. Примеры сетей связи включают в себя, например, LAN, WAN и компьютеры и сети, которые образуют Интернет.

Один или более вариантов осуществления настоящего изобретения могут быть реализованы на практике с другими конфигурациями компьютерных систем, включая портативные устройства, микропроцессорные системы, основанную на микропроцессорах или программируемую бытовую электронику, миникомпьютеры, универсальные компьютеры и т.д. Кроме того, изобретение может быть реализовано на практике в распределенных вычислительных средах, где задачи выполняются удаленными устройствами обработки, которые связаны через сеть.

Хотя были описаны один или более вариантов осуществления изобретения, различные их изменения, дополнения, перестановки и эквиваленты включены в объем изобретения.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ верификации того, что пользовательский код, введенный в вычислительную систему, соответствует истинному коду, содержащему один или более кодовых символов, полученных из идентификационной информации продукта, причем способ содержит этапы, на которых

определяют набор кодовых символов, которые могут быть использованы в идентификационном коде продукта;

определяют один или более поднаборов кодовых символов, причем каждый из поднаборов содержит два или более предварительно определенных эквивалента идентификации из набора кодовых симво-

лов, причем каждый поднабор имеет предварительно определенный идентификатор поднабора;

определяют преобразование для генерации идентификационной информации истинного кода на основе предварительно определенного идентификатора поднабора, причем преобразование включает в себя обфускацию, шифрование, хэширование, или при этом истинный код является идентификационной информацией;

принимают введенный пользователем код, в том числе принимают набор из одного или более кодовых символов, введенных пользователем с целью верификации продукта;

дешифруют введенный пользователем код путем применения упомянутого преобразования, чтобы получить идентификационную информацию введенного пользователем кода, причем преобразование является обфускацией, шифрованием, хэшированием, или при этом введенный пользователем код является идентификационной информацией;

генерируют идентификационную информацию истинного кода;

сравнивают пользовательский код и истинный код и определяют количество ошибок или типов ошибок, обнаруженных между введенным пользователем кодом и истинным кодом;

причем каждая ошибка имеет ассоциированную оценку вероятности, и выполняют коррекцию одной или более ошибок, обнаруженных во введенном пользователем коде, на основе оценки вероятности, ассоциированной с упомянутыми одной или более обнаруженными ошибками, и причем код корректируют, только если оценка вероятности коррекции выше предварительно определенного порога;

причем оценка вероятности содержания вычисляется или предварительно определена для замещения символов, включающих в себя для некоторых символов первую оценку вероятности содержания для физической близости и вторую оценку вероятности содержания для сходства символов, и

верифицируют то, что идентификационная информация введенного пользователем кода соответствует идентификационной информации истинного кода.

2. Способ по п.1, дополнительно содержащий комбинирование нелинейно первой оценки вероятности содержания и второй оценки вероятности содержания.

3. Способ по любому одному или более предыдущим пунктам, дополнительно содержащий этап, на котором определяют истинный код, содержащий один или более кодовых символов, выведенных из идентификационной информации продукта.

4. Способ по любому одному или более предыдущим пунктам, в котором одно или более из количества ошибок и типов ошибок, обнаруженных между введенным пользователем кодом и истинным кодом, записывают.

5. Способ по любому одному или более предыдущим пунктам, в котором этап верификации истинного кода содержит этап, на котором верифицируют истинный код только тогда, когда одно или более из количества ошибок и типа ошибок ниже предварительно определенного порога.

6. Способ по любому одному или более предыдущим пунктам, в котором все кодовые символы содержатся в одном или более поднаборах.

7. Способ по любому одному или более предыдущим пунктам, в котором все поднаборы содержат только два кодовых символа.

8. Способ по любому одному или более предыдущим пунктам, в котором на оценку вероятности, ассоциированной с ошибочным введенным пользователем символом, оказывают влияние символы, которые находятся рядом с ошибочным введенным пользователем символом.

9. Способ по любому одному или более предыдущим пунктам, в котором два последовательно введенных пользователем символа обнаруживают как ошибки и выполняют коррекцию обоих символов перестановкой этих двух последовательно введенных пользователем символов, обнаруженных как ошибки, в результате чего символы становятся правильными при сравнении с истинным кодом, независимо от того, превышает ли или нет оценка вероятности коррекции любой ошибки предварительно определенный порог.

10. Система для верификации того, что пользовательский код, введенный в вычислительную систему, соответствует истинному коду, причем система содержит компьютеризованный процессор, сконфигурированный для исполнения инструкций для

определения набора кодовых символов, которые могут быть использованы в идентификационном коде продукта;

определения одного или более поднаборов кодовых символов, причем каждый из поднаборов содержит два или более предварительно определенных эквивалента идентификации набора кодовых символов, причем каждый поднабор имеет предварительно определенный идентификатор поднабора;

определения преобразования для генерации идентификационной информации истинного кода на основе предварительно определенного идентификатора поднабора, причем преобразование включает в себя обфускацию, шифрование, хэширование, или при этом истинный код является идентификационной информацией;

приема введенного пользователем кода, в том числе приема набора из одного или более кодовых символов, введенных пользователем с целью верификации продукта;

дешифрования введенного пользователем кода путем применения преобразования, чтобы получить

идентификационную информацию введенного пользователем кода, причем преобразование является обфускацией, шифрованием, хэшированием, или при этом введенный пользователем код является идентификационной информацией;

генерирования идентификационной информации истинного кода;

сравнивают пользовательский код и истинный код и определяют количество ошибок или типов ошибок, обнаруженных между введенным пользователем кодом и истинным кодом;

причем каждая ошибка имеет ассоциированную оценку вероятности, и выполняют коррекцию одной или более ошибок, обнаруженных во введенном пользователем коде, на основе оценки вероятности, ассоциированной с упомянутыми одной или более обнаруженными ошибками, и причем код корректируют, только если оценка вероятности коррекции выше предварительно определенного порога;

причем оценка вероятности содержания вычисляется или предварительно определена для замещения символов, включающих в себя для некоторых символов первую оценку вероятности содержания для физической близости и вторую оценку вероятности содержания для сходства символов,

и верификации того, что идентификационная информация введенного пользователем кода соответствует идентификационной информации истинного кода.

11. Система по п.10, дополнительно содержащая комбинирование нелинейно первой оценки вероятности содержания и второй оценки вероятности содержания.

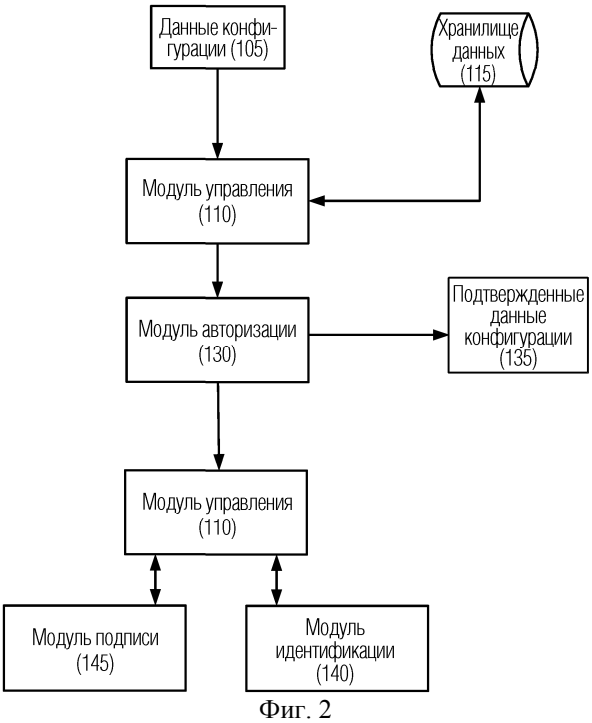
12. Система по любому одному или более из пп.10 или 11, дополнительно содержащая инструкции для определения истинного кода, содержащего один или более кодовых символов, выведенных из идентификационной информации продукта.

13. Система по любому одному или более из пп.10-12, дополнительно содержащая инструкции для сравнения пользовательского кода и истинного кода.

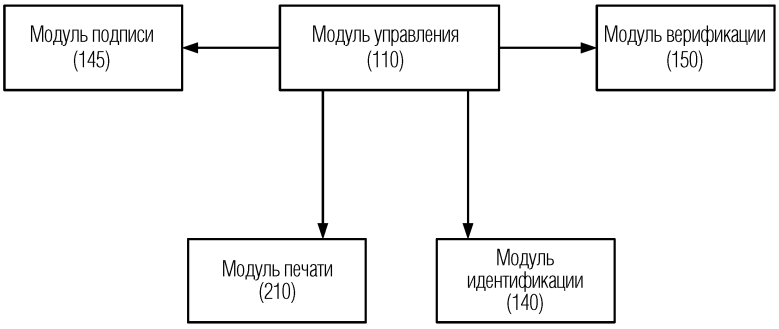
14. Система по любому одному или более из пп.10-13, дополнительно содержащая инструкции для определения количества ошибок или типов ошибок, обнаруженных между введенным пользователем кодом и истинным кодом.

Номер символа	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Код	3	S	6	8	P	5	F	5	W	X	Z	9	L	M	J	4	6	2	E	L

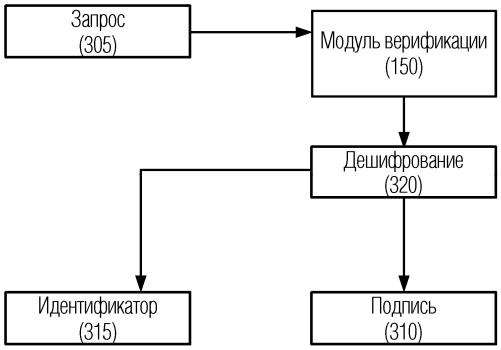
Фиг. 1



Фиг. 2



Фиг. 3



Фиг. 4

Евразийская патентная организация, ЕАПВ
Россия, 109012, Москва, Малый Черкасский пер., 2