УДК 004.056.53

ИЯД МОХД МАДЖИД АХМАД ШАХРУРИ

ИСПОЛЬЗОВАНИЕ ХЕШ-ПАМЯТИ ДЛЯ БЫСТРОЙ АУТЕНТИФИКАЦИИ АБОНЕНТОВ МНОГОПОЛЬЗОВАТЕЛЬСКИХ СИСТЕМ

В статье предлагается подход к повышению скорости аутентификации абонентов многопользовательских систем. Этот подход имеет в основе использование хеш-памяти и учитывает многоуровневый характер организации памяти. Разработана вероятностная модель хеш-поиска в квазипостоянных массивах ключей, хранящихся в двухуровневой памяти. На основе этой модели предложена эффективная организация хеш-доступа к аутентификационной информации. Показано, что предложенная организация хеш-поиска требует не более одного обращения к медленной памяти нижнего уровня.

In paper an approach for increasing the rate of multi-user system abonent authentication has been proposed. This approach is based on utilization of hash-memory and takes into account multilevel modern memory organization. The probabilistic model of hash-searching in near-permanent keys arrays which are storages in two-level memory has been developed. On the base of this model the effectiveness organization of hash-access to authentication information has been proposed. It has been shown that proposed hash-access organization requires ensure no more then one access to slow low-level memory.

Введение

Одним из определяющих факторов прогресса во всех областях человеческой деятельности является развитие и расширение информационной интеграции на основе сетевых технологий. Компьютерные сети, предоставляя качественно новые возможности доступа к информации, позволяют заметно повысить эффективность интеллектуальной деятельности. В настоящее время процесс информационной интеграции продолжает динамично углубляться и охватывать новые области применения на основе расширения использования беспроводных линий передачи данных [1]. Тенденцией последних лет стал рост доли среди терминальных устройств сетей портативных микроконтроллеров: датчиков и управляющих устройств.

Важнейшим условием эффективности информационной интеграции на основе компьютерных сетей является обеспечение надежной защиты данных и разграничение прав доступа в интегрированной информационной среде. Расширение использования беспроводных технологий передачи данных, резкий рост числа абонентов компьютерных сетей, повышение производительности компьютерных систем, используемых для нарушения защиты – являются, в современных условиях, факторами повышения риска информационной безопасности интегрированных систем. С другой

стороны, процесс расширения использования интегрированных систем обработки данных все в большей степени охватывает области человеческой деятельности, связанные с техногенным и иными видами рисков. Указанные факторы определяют необходимость совершенствования средств защиты данных в интегрированных информационных средах и, в частности, повышения эффективности аутентификации удаленных абонентов.

В ряду основных направлений совершенствования аутентификации удаленных абонентов весомую роль играет повышение оперативности реализации функций разграничения прав доступа. Быстрый рост числа абонентов интегрированных систем повышает важность проблемы обеспечения высокой скорости их аутентификации.

Таким образом, актуальной и важной составляющей для современного этапа развития технологии информационной интеграции является повышение скорости аутентификации удаленных абонентов интегрированных систем хранения и обработки информации.

Анализ существующих решений проблемы эффективной аутентификации удаленных пользователей

Практически все современные многопользовательские системы имеют средства защиты от несанкционированного доступа к их ресурсам. В большинстве систем информация, связанная с предоставлением прав пользования ресурсами, представлена в форме закрытого пароля пользователя [2]. Обычно в системе имеется список легальных пользователей, в котором им сопоставляются пароль и перечни ресурсов, к которым возможен доступ. При этом пароль выполняет функции аутентификации пользователя, входящего в систему.

В частности, в широко известной системе UNIX [1] поиск выполняется по идентификатору пользователя, введенный им пароль шифруется с использованием модифицированного алгоритма DES и сравнивается с зашифрованным паролем, хранящимся в списке. Существует ряд многопользовательских систем для которых поиск производится по паролю [2].

При реализации аутентификации пользователей системы возникает технологическая проблема поиска по паролю или идентификатору пользователя. В большинстве систем способ реализации процедуры поиска не специфицирован. Для систем с небольшим числом пользователей поиск вполне может выполняться последовательно. Однако, с развитием информационной интеграции, растет число систем, количество легальных пользователей которых превышает десятки и сотни тысяч [1]. Большинство таких систем можно рассматривать как системы массового обслуживания, для которых важным показателем является время реакции на запрос Пользователя. Существует вполне реальная опасность уменьшения времени реакции системы на запросы пользователя, вызванная умышленным

или случайным ростом интенсивностей потока обращений со стороны несанкционированных пользователей. Следует иметь в виду еще один важный аспект рассматриваемой проблемы: повышение скорости аутентификации положительно сказывается на безопасности многопользовательских систем, поскольку ряд технологий незаконного проникновения в такие системы основаны на задержках аутентификации [3].

Поэтому, для рассматриваемого класса многопользовательских систем актуальной является проблема ускорения аутентификации пользователей за счет повышения скорости поиска по идентификатору или паролю.

Необходимо отметить, что проблема повышения эффективности аутентификации пользователей систем и сетей включает ряд других задач, решение которых может быть достигнуто за счет ускорения поиска в больших массивах данных. К таким задачам, в частности, относится проверка качества пароля при его генерации системой или выборе пользователем [4]. Одним из наиболее перспективных способов решения этой задачи, а также задачи селекции паролей, используемых для несанкционированного доступа к ресурсам системы, является использование списка потенциально опасных паролей. Для того, чтобы такой подход был достаточно эффективным, этот список должен быть достаточно большим и составлять более 1 Гбайта. Поиск в таком списке должен осуществляться при каждом цикле аутентификации пользователя, следовательно, процедуру поиска необходимо реализовать достаточно быстро.

Память всех современные компьютерных систем имеет виртуальную организацию, которая предполагает наличие быстродействующей кэш-памяти относительно небольшого объема и медленной основной памяти большого объема. Обмен данными между кэш-памятью и основной памятью производится страницами. Эту важную особенность необходимо учитывать при организации поиска пароля в списке тех, доступ к ресурсам для которых разрешен. Для интегрированных систем с большим числом абонентов списки паролей и соответствующих им кодов доступа хранятся в основной памяти. Организация хранения паролей должна быть выполнена таким образом, чтобы поиск идентифицирующей информации пользователя требовал перемещения в кэш-память не более, чем одной страницы. При этом полагается, что для реализации поиска в кэш-памяти выделяется определенное количество страниц. При поиске идентифицирующего кода на соответствие одному из хранящихся в памяти системы, этот код играет роль ключа поиска, поэтому в дальнейшем изложении целесообразно придерживаться традиционной для поиска терминологии, понимая под ключом поиска идентификатор или пароль удаленного пользователя. В общем виде можно полагать, что поиск ключа производится в массиве из m ключей, причем в одной странице можно максимально разместить w ключей вместе со связанной с ними информацией, используемой для указания прав доступа.

Основные особенности поиска при аутентификации пользователей состоят в следующем:

1. Интенсивность использования ключей для поиска на несколько порядков превышает интенсивности их использования, что позволяет считать массив ключей и связанной с ними информации квазипостоянным. Эта особенность характерна и для указанной выше задачи поиска пароля в списке потенциально опасных.

2. Для исключения несанкционированного доступа к паролям, хранящимся в памяти, необходимо исключить их хранение там в явном виде.

Приемлемая скорость сравнения паролей может быть достигнута при использовании двух технологий поиска:

1. Применение логического (с использованием бинарных деревьев) или физического упорядочения массивов ключей и бинарного поиска.

2. Применение хеш-адресации.

Реализация бинарного поиска с учетом перемещения в кэш-память не более одной страницы из основной памяти требует организации упорядоченного каталога страниц в кэш-памяти. В таком каталоге необходимо хранить коды первых ключей каждой страницы, то есть m/w ключей. Существенные затраты времени может потребовать изменение ключа или добавление нового. Для их уменьшения используют резерв [1] свободной памяти в каждой из страниц. Коэффициент β загрузки памяти определяется в виде отношения максимального числа w ключей, хранящихся в странице к числу w_R ключей, хранящихся в странице с учетом резерва: $\beta=w_R/w<1$. .

Существенно большую эффективность поиска по ключу обеспечивает хеш-адресация. При квазипостоянном характере массива ключей-паролей существует возможность подбора хеш-преобразования, не образующего коллизий, что обеспечивает потенциально наибольшую скорость поиска, поскольку гарантирует необходимость перемещения в кэш-память не более одной страницы. Присущая хеш-адресации неполная загрузка памяти позволяет достаточно просто организовать добавление новых ключей. Важным достоинством применения хеш-памяти по сравнению с упорядочением ключей представляется отсутствие необходимости в каталоге, занимающем кэш-память.

К настоящему времени подробно исследованы вопросы получения хеш-преобразования, не образующего коллизий для постоянных массивов ключей. Такие хеш-преобразования получили название совершенных (perfect hash transformation). В работе [5] намечены подходы к организации хеш-адресации динамических массивов ключей с учетом виртуальной организации хеш-памяти. Однако хеш-адресация квазипостоянных массивов ключей в системах с виртуальной памятью имеет ряд важных особенностей, не позволяющих прямо использовать существующие решения.

Таким образом, для достижения потенциально высоких характеристик поиска, обеспечиваемых применением хеш-адресации при аутентификации пользователей систем и сетей, необходимо решить ряд задач, связанных с организацией хеш-поиска с учетом специфики указанного ее применения. Решение этих задач требует проведения специальных исследований и может быть получено на основе модели хеш-поиска, учитывающей особенности задач аутентификации.

Целью работы является разработка математической модели хеш-поиска в квазистатических массивах идентификаторов пользователей, учитывающей особенности виртуальной организации памяти, а также создание на основе этой модели технологии аутентификации абонентов многопользовательских систем с использованием хеш-поиска.

Разработка модели хеш-поиска в квазистатических массивах

Задачей разрабатываемой модели хеш-поиска является получение аналитических зависимостей между характеристиками хеш-памяти, определяющими ее организацию, позволяющих эффективно решать задачи оптимизации структуры хеш-памяти на этапе ее проектирования исходя из заданных критериев эффективности хеш-поиска.

В основу разрабатываемой модели положена концепция получения хеш-преобразования $H(X)$, обеспечивающего отображение заданного множества Ω из m ключей на s страниц хеш-памяти таким образом, чтобы количество ключей, адресуемых в каждую из страниц, не превышало $(\alpha+\delta)\cdot w$, где α – коэффициент загрузки хеш-памяти, δ - допустимая вариация загрузки страницы хеш-памяти, $\alpha+\delta\leq 1$. Коэффициент α загрузки хеш-памяти определяется отношением количества m сохраняемых в ней записей к максимально возможному их количеству $M = s\cdot w$, исходя из объема хеш-памяти:.

$$\alpha = \frac{m}{M} = \frac{m}{s \cdot w} \qquad (1)$$

Под записью понимается информация о абоненте системы, сохраняемая в хеш-памяти. Запись содержит блок идентифицирующей информации и блок данных, связанных с пользователем. Этот блок содержит данные о правах доступа и другую, связанную с пользователем информацию. Вместо блока данных в записи может содержаться адресная ссылка на место его хранения.

Получение хеш-преобразования $H(X)$, удовлетворяющего указанному выше условию может быть получено путем подбора. В качестве механизма подбора хеш-преобразования целесообразным представляется использование стандартизированного шифроблока типа DES или Rijndael, который осуществляет криптографическое однозначное шифрование с использованием ключа K данных D в код C: $C = H_K(D)$ [6]. Ключевая для поиска информация X, в таком варианте, используется в качестве входных

данных шифроблока, ключ K шифроблока выступает в роли настроечного кода и является, по существу, номером хеш-преобразования. Выходной код C шифроблока разделяется на две части: h-разрядный фрагмент используется в качестве хеш-адреса $A_K(X)$ страницы, а оставшиеся разряды представляют собой хеш-свертку $S_K(X)$ ключа X поиска. Соответственно, подбор хеш-преобразования $H_K(X)$ выполняется путем изменения ключа K шифроблока.

Считая, что на странице может разместиться w ключей, хеш-адрес $A_K(X)$ страницы представляет собой $h=\log_2 s$ – разрядный код. Хеш-функция распределяет m ключей по s группам, которые содержат $\eta_1,\eta_2,\ldots,\eta_s$ ключей, причем $\sum_{j=1}^{s}\eta_j = m$. Для того, чтобы хеш-преобразование распределяло ключи по страницам хеш-памяти, необходимо, чтобы количество хеш-адресов, каждой из страниц не превышало максимально допустимого числа $u=(\alpha+\delta)\cdot w$ записей в странице: $\forall j\in\{1,\ldots,s\}$: $\eta_j \le u$. При этом, в каждой из страниц хеш-памяти обеспечивается наличие свободного объема памяти, достаточного для размещения $w\cdot(1-\alpha-\delta)$ записей.

Если исходить из того, что хеш-преобразование $H_K(X)$ формирует равномерно распределенные хеш-адреса, то в одну страницу, в среднем, попадает m/s хеш-адресов. В качестве теоретической модели распределения хеш-адресов представляется корректным использовать вероятностную модель Бернулли. Согласно этой модели, попадание хеш-адресов m ключей в пределы фиксированной страницы памяти можно рассматривать как m опытов, в каждом из которых с вероятностью $1/s$ происходит событие – попадание хеш-адреса в адресное пространство этой страницы. Тогда, согласно свойствам рассматриваемой модели, можно считать, что математическое ожидание числа хеш-адресов, попадающих в фиксированную страницу равно m/s с дисперсией $m\cdot\frac{1}{s}\cdot(1-\frac{1}{s})$. Тогда, согласно теореме Муавра-Лапласса, количество хеш-адресов, попадающих в рамки страницы, подчинено распределению Гаусса с математическим ожиданием m/s и дисперсией $m\cdot\frac{1}{s}\cdot(1-\frac{1}{s})$. Если учесть, что число s страниц хеш-памяти достаточно велико, то дисперсию числа хеш-адресов, попадающих в фиксированную страницу, можно приближенно считать равной m/s.

Вероятность P_{os} переполнения страницы, то есть, вероятность того, что число хеш-адресов, попадающих в фиксированную страницу хеш-памяти, превысит u, для нормального распределения с учетом (1) определиться следующим выражением:

$$P_{os} = 0.5 - \Phi\left(\frac{u - m/s}{\sqrt{m/s}}\right) = 0.5 - \Phi\left(\frac{w\cdot(\alpha+\delta) - m/s}{\sqrt{m/s}}\right) = 0.5 - \Phi\left(\delta\cdot\sqrt{\frac{w}{\alpha}}\right) \qquad (2)$$

Для постоянного массива ключей, то есть, в ситуации, когда в странице нет необходимости иметь запас свободной памяти для $w\cdot(1-\alpha-\delta)$ записей, $\delta = 1-\alpha$ и вероятность P_{osw} переполнения страницы, то есть, вероятность того, что число хеш-адресов, попадающих в фиксированную страницу хеш-памяти, превысит w, определиться следующим выражением:

$$P_{osw} = 0.5 - \Phi(\sqrt{w}\cdot\frac{1-\alpha}{\sqrt{\alpha}}) \tag{3}$$

Из выражения (3) следует, что вероятность переполнения страницы в определяющей степени зависит от значения коэффициента δ запаса свободной памяти в странице, а также от коэффициента α заполнения хеш-памяти, а также от объема страницы и памяти, требующейся для хранения одной записи.

Для того, чтобы исключить переполнение всех s страниц при из заполнении фиксированным множеством Ω из m ключей, необходимо подобрать соответствующее хеш-преобразование. Вероятность P_0 того, что за одну пробу хеш-преобразования удастся исключить переполнение всех страниц хеш-памяти определяется произведением вероятностей того, что каждая из s страниц не будет переполнена:

$$P_0 = (1-P_{OS})^S = \left[0.5 + \Phi(\delta\cdot\sqrt{\frac{w}{\alpha}})\right]^S \tag{4}$$

Среднее число g проб, которое необходимо выполнить для подбора хеш-преобразования, при котором исключается переполнение страниц хеш-памяти, определяется следующим выражением:

$$g = \sum_{j=1}^{\infty} j\cdot P_0\cdot(1-P_0)^{j-1} = \frac{1}{P_0} \tag{5}$$

Проведенные экспериментальные исследования на основе статистического моделирования доказали адекватность изложенной математической модели хеш-адресации квазистатического массива ключей в хеш-памяти, разделенной на страницы. Предложенную модель можно использовать для оптимизации параметров хеш-памяти.

Из (5) следует, что при заданном числе g_z проб для подбора хеш-преобразования, обеспечивающего распределение записей по страницам с заполнением последних не более, чем на $(\alpha+\delta)\cdot100\%$ значения α, δ и w должны удовлетворять условию:

$$\sqrt{\frac{w}{\alpha}}\cdot\delta = \Phi^{-1}(\sqrt[S]{\frac{1}{q_Z}} - 0.5) \tag{6}$$

Если необходимо обеспечить быстрый подбор хеш-преобразования, то можно исходить из условия $s\cdot P_{os} \approx 1$. Тогда при больших значениях s справедливо следующее приближение:

$$P_0 = (1 - P_{OS})^S = 1 - \sum_{i=1}^{s} (-1)^i \cdot \left(\begin{matrix} s \\ i \end{matrix} \right) \cdot {P_{OS}}^i \approx 1 - \sum_{i=1}^{s} (-1)^i \cdot \frac{1}{i!} = 1 - \frac{1}{e} \approx 0.632 \cdot$$

соответственно, $q \approx 1/P_{os} = 1.58$. Это означает, что для того, чтобы обеспечить быстрый подбор хеш-преобразования, значения α, δ и w должны удовлетворять условию:

$$\Phi\left(\sqrt{\frac{w}{\alpha}} \cdot \delta\right) = 0.5 - \frac{1}{s} \qquad (7)$$

Анализ полученной математической модели показывает, что наиболее компромиссным вопросом организации хеш-поиска при аутентификации пользователей является выбор объема страниц, которыми производится обмен между основной и кэш-памятью. Очевидно, что скорость поиска существенно зависит от времени загрузки адресуемой хеш-адресом страницы основной хеш-памяти в кэш-память, которое, в свою очередь, зависит от размера страницы. Следовательно, с точки зрения достижения высокой скорости хеш-поиска, целесообразно уменьшать размер страницы, в то время как, с позиций снижения времени на подбор хеш-преобразования, согласно (5) оправдано увеличение размера страницы. Решение указанного компромисса может быть найдено исходя из заданной интенсивности обновления массива ключей: чем чаще изменяются ключи, тем меньшим должно быть время подбора хеш-преобразования и тем большим должна быть величина δ. Разрешение этих противоречивых требований можно достичь либо за счет увеличения размера страницы, либо за счет снижения коэффициента α загрузки основной хеш-памяти.

Организация хеш-поиска

Сущность организации хеш-поиска идентификатора пользователя состоит в следующем. Основной массив идентификаторов, в виде хеш-сверток и связанной с ними информации сохраняется по хеш-адресам в основной памяти. Подбором хеш-преобразования для квазипостоянного массива ключей обеспечивается распределение записей по страницам основной памяти с соответствии с хеш-адресами.

В кэш-памяти помещается активная страница основной памяти, а также область записей, которые добавляются в процессе регистрации новых пользователей и для которых не хватает места в соответствующих их идентификаторам страницах основной памяти.

Кроме того, в кэш-памяти может быть организовано хранение кодов хеш-преобразований, наиболее часто используемых записей.

Перед началом сеанса работы системы, либо в специально выделенное время обновления системы для множества Ω идентификаторов-ключей m пользователей подбором находится хеш-преобразование $H_K(X)$, которое распределяет записи пользователей по s страницам хеш-памяти таким образом, что в каждой из них имеется свободный объем, позволяющих

записать в каждую страницу не менее $w\cdot(1-\alpha-\delta)$ записей. Найденное подбором хеш-преобразование $H_K(X)$, в виде кода K ключа фиксируется и используется при хеш-поиске. Таким образом, множество Ω идентификаторов-ключей разделяется на s подмножеств, каждое из которых включает не более $w\cdot(\alpha+\delta)$ элементов. Это означает, в каждой странице сохраняется не более $w\cdot(\alpha+\delta)$ записей пользователей. Кроме того, шифроблоком формируется $\log_2 w$–разрядный хеш-адрес $U_K(X)$, определяющий адрес записи, соответствующей ключу X внутри страницы. Код хеш-свертки $S_K(X)$ вместе с хеш-адресом $A_K(X)$ страницы и хеш-адресом $U_K(X)$ внутри страницы, в силу однозначности преобразований, реализуемых шифроблоком, однозначно определяют идентификатор X пользователя. Внутри страницы записи размещаются по их хеш-адресам $U_K(X)$ с допустимостью возникновения коллизий, которые разрешаются с использованием известных технологий [6], в частности, пробинга.

Каждая из записей содержит код хеш-свертки $S_K(X)$ идентификатора, код пароля, коды прав доступа, другая информация, относящаяся к пользователю. Для сокращения длины записи, и, соответственно, увеличения w, вместо всех кодов, кроме $S_K(X)$ может храниться код адресной ссылки на место их хранения.

После сохранения в основной хеш-памяти записей всех, зарегистрированных на момент обновления пользователей, система их аутентификации готова к работе.

При удалении записи X производится ее исключение из страницы путем обнуления занимаемой ею памяти с учетом перемещения цепочки коллизий.

При добавлении новой записи X производится перемещение адресуемой $A_K(X)$ страницы из основной памяти в кэш-память. Если страница не заполнена, то по адресу $U_K(X)$ производится попытка дописать новую запись X. Если соответствующая область страницы занята, то выполняется пробинг до нахождения свободной области. Если страница, адресуемая $A_K(X)$ оказывается полностью занятой, то запись X, адресуемая частью разрядов хеш-адреса $A_K(X)$ помещается в специально отведенную область кэш-памяти, являющуюся, по-существу областью переполнения. Если адресуемая ячейка занята, то осуществляется разрешение коллизий с использованием одного из видов пробинга. При этом запись, помещаемая в область переполнения содержит не хеш-свертку идентификатора X, а его полный код. Если выделенная область переполнения оказывается полностью занятой, то необходимо произвести цикл обновления.

Технологически описанная процедура размещения в хеш-память новой записи может производиться двумя способами, отличающихся приоритетом при размещения записи в одной из двух областей. При регистрации новых пользователей в период между циклами обновления системы аутентификации, соответствующие записи могут быть помещены:

- в область переполнения относительно небольшого объема, локализованной в кэш-памяти;
- в основной хеш-памяти, занимая свободные зоны страниц, адресуемых хеш-адресом идентификатора вновь регистрируемого пользователя.

Теоретически можно рассмотреть два варианта организации дописывания:

1. Вначале записи дописываются в свободные зоны соответствующих страниц основной памяти, а при отсутствии в таких зон – в область переполнения кэш-памяти.

2. Вначале в область переполнения кэш-памяти, а по заполнении последней – в свободные зоны соответствующих страниц основной хеш-памяти.

Очевидно, что первый вариант обеспечивает большее количество новых записей, которые могут быть помещены в хеш-память до ее обновления. Преимуществом второго варианта является более эффективное использование объема области переполнения кэш-памяти, что имеет следствием большую скорость аутентификации пользователей.

При поступлении n-разрядного идентификатора X пользователя шифроблоком с использованием в качестве ключа кода K выполняется вычисление h-разрядного хеш-адреса $A_K(X)$ страницы, хеш-адреса внутри страницы $U_K(X)$ и кода хеш-свертки $S_K(X)$. Формируется также хеш-адрес $Q_K(X)$ записи с идентификатором X в области переполнения кэш-памяти. Адрес $Q_K(X)$ представляет собой подмножество разрядов кодов $A_K(X)$, $U_K(X)$ и $S_K(X)$. По адресу $Q_K(X)$ производится хеш-поиск в обрасти переполнения кэш-памяти. Если поиск завершается успешно, то есть в области переполнения найдена запись с идентификатором X, то пользователю разрешается доступ к ресурсам системы в пределах выделенных прав.

Одновременно с поиском в области переполнения кэш-памяти, страница, адресуемая кодом $A_K(X)$ загружается из основной памяти в кэш-память. Затем по адресу $U_K(X)$ из страницы считывается запись, ее хеш-свертка сравнивается с кодом $S_K(X)$. В случае совпадения указанных кодов выполняется сравнение паролей и осуществляется предоставление прав доступа. В случае несовпадения кодов хеш-сверток выполняется пробинг до нахождения совпадения либо пустой записи. Последнее означает, что запись с ключом X не хранится в памяти. Время t_1 поиска по ключу-идентификатору X пользователя определяется суммой времен: t_H – выполнения хеш-преобразования с использованием шифроблока, t_S - времени свопинга адресуемой $A_K(X)$ страницы из основной памяти в кэш-память, t_X -временем хеш-поиска в выбранной странице кэш-памяти: $t_1 = t_H + t_S + t_X$. Учитывая, что $t_S > t_H >> t_X$, время аутентификации, в основном, определяется временем, затрачиваемым на свопинг страницы из основной памяти в кэш-память.

Выводы

Итогом проведенных исследований, направленных на повышение эффективности аутентификации удаленных абонентов многопользовательских систем за счет ускорения поиска по идентифицирующему коду являются следующие результаты:

1. Выявлены специфические особенности поиска по идентифицирующему коду удаленного пользователя.

2. Разработана модель хеш-поиска в квазистатических массивах идентификационных кодах пользователей, учитывающая многоуровневую организацию памяти.

3. На основе разработанной модели предложена организация хеш-поиска в квазипостоянных массивах идентифицирующих пользователей кодов. Доказано, что время аутентификации определяется обращением не более, чем одной странице памяти нижнего уровня.

Предложенная организация хеш-памяти может быть эффективно использована для повышения скорости аутентификации пользователей как компьютерных систем коллективного пользователей, так и абонентов сетей.

Список использованной литературы

1. Столлингс В. Криптография и защита сетей: принципы и практика. М.: Изд.дом.”Вильямс”.- 2001-621 с.
2. Соколов А.В., Шаньгин В.Ф. Защита информации в распределенных корпоративных сетях и системах. М.: ДМК.-2002.- 655 с.
3. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М.:”Триумф”. - 2002.-816 с.
4. Stafford E. Preventing Weak Password Choices // Computers and security.-1992.-№ 3.- P.46-53.
5. Мнацаканов А.В. Организация хеш-поиска в системах с виртуальной памятью // Матеріали VII Міжнародної науково-технічної конференції ”Системний аналіз та інформаційні технології” (13-16 вересня 2006 р., м.Київ). К.: НТУУ ”КПІ”.-2006.- С.208-211.
6. Салех Ибрагим Аль-Омар. Использование генераторов булевых функций для повышения эффективности хеш-памяти. // Вісник Національного технічного університету України ”КПІ”. Інформатика, управління та обчислювальна техніка.- 2003.- № 40.- С.131-140.