УДК 004.056

ВОЛОКИТА А.Н.,
КОХАНЕВИЧ И.В.

БЛОЧНОЕ ШИФРОВАНИЕ В CLOUD COMPUTING С ИСПОЛЬЗОВАНИЕМ ДЕРЕВА КЛЮЧЕЙ

Обеспечение безопасности – одна из основных проблем современных облачных вычислений. Именно отсутствие достаточных гарантий безопасности хранения данных становится основным сдерживающим фактором при переходе в «облако». Статья посвящена механизму шифрования данных в cloud computing. Рассматривается распределенная файловая система, в которой файлы разделяются на отдельные блоки с последующим шифрованием на основе бинарного дерева ключей. Также описан способ построения данного дерева. Показаныдальнейшиеперспективыразвитияпредложенногометода.

Providing secure is an important component of cloud computing. Furthermore, the lack of sufficient data security is a main constraint to move to the “cloud”. The article is devoted to data encryption in cloud computing. The author considers a distributed file system in which files are divided into separate blocks. Then he describes a method of encryption of these blocks based on a binary tree of keys. Moreover the author explains a way of constructing of this tree. The paper considers the prospects for further development of the proposed method.

1. Введение

За последние пять лет стремительно набирает обороты новая тенденция в развитии вычислительных технологий – облачные вычисления (англ. Cloud computing). Компания маркетинговых исследований Forrester Research предполагает [1], что суммарная прибыль всех сегментов облачных вычислений (SaaS, PaaS и IaaS) вырастит с 15 миллиардов долларов США в 2010 году до почти 160 миллиардов в 2020, что будет представлять 27% рост в годовом исчислении.

По имеющимся статистическим данным [2], компании в сфере информационных технологий представляютоблачные вычисления будущегов основном как центры обработки данных, которые предоставляют надежную защиту информации, а также возможность динамического роста в виде опций.

Исходя из данных Morgan Stanley Research [3], первое место среди всего списка проблем облачных вычислений занимает проблема обеспечения безопасности. В рамках данного исследования, отсутствие достаточных гарантий безопасности хранения данных было названо самым большим препятствием при переходе в «облако» (24% респондентов), это вдвое больше, чем следующаяпроблема – неочевидность экономической выгоды (12% респондентов).

2. Постановка задачи

При рассмотрении вопроса безопасности данных в облачных вычислениях,рассмотрим более детально следующие пункты: [4, 5, 6]

1. безопасность хранения данных;
2. безопасность сети;
3. локальность данных;
4. целостность данных;
5. сегрегация данных;
6. безопасность доступа к данным;
7. аутентификация и авторизация;
8. уязвимости виртуализации;
9. резервное копирование.

Безопасность данных.

В традиционной модели, при которой приложение развертывается на собственном оборудовании, данные размещаются в пределах границ организации и подчиняются ее политике контроля доступа к информации. Вотличие от традиционной модели, в облачных вычисленияхкорпоративные данные хранятся за пределами организации. Следовательно, поставщик должен принять дополнительные мерыобеспечения безопасности информации и предотвращения несанкционированного доступа из-за уязвимости в приложении или через сотрудника-злоумышленника [7].

Безопасность сети.

Вся информация, которая передается посредством сети, должна быть надежно защищена. Данное условие выполняется с помощью использования механизмов шифрования сетевого трафика, таких как SSLи TLS.

Локальность данных.

В некоторых случаях пользователю необходимо знать, где физически хранятся его данные. Например, в большинстве стран Евросоюза и Южной Америки информация,являющаяся гос-

ударственной тайной, не может покидать пределы государства. В Украине также передача персональных данных иностранным субъектам отношений, связанных с персональными данными, осуществляется лишь при условии обеспечения надлежащей защиты персональных данных, при наличии соответствующего разрешения и в случаях, установленных законом или международным договором Украины, в порядке, установленном законодательством [8].

Целостность данных.

Проблема целостности данных является основной для любой системы. Она решается довольно просто в отдельно стоящей системе с единственной базой данных. Целостность данных в таком случаи поддерживается через транзакции и/или ограничения БД. При этом, транзакции должны следовать принципамACID (atomicity,consistency, isolationanddurability – атомарность, постоянство, изолированность и надежность).Большинство БД поддерживают ACID, тем самым обеспечивая необходимую целостность данных.

Сегрегация данных.

Одной из основных характеристик облачных вычислений является многопользовательский доступ к данным. При этом возникает ситуация, когда на одном и том же устройстве хранится информация различных пользователей. Данная ситуация предоставляет злоумышленнику возможность, например используя SQL-инъекции, получить доступ к чужим данным [9].

Безопасность доступа к данным.

Компании использующие «облако» для своих бизнес процессов, могут использовать собственную политику безопасности, предоставляя сотрудникам разные права доступа к информации. Исходя из этого, модель облачных вычислений должна быть достаточно гибкой, чтобы обеспечить данную возможность.

Аутентификация и авторизация.

Cloud computing должен иметь удобные механизмы аутентификации и авторизации, предоставляя пользователю возможность легко создавать и удалять отдельные учетные записи сотрудников предприятия. При этом возможно использование собственной политики безопасности.

Уязвимости виртуализации.

Виртуализация является одним из главных компонентов облачных вычислений. Данный механизм предоставляет возможность запускать на одном физическом ресурсе набор изолированных друг от друга виртуальных машин. Проблема заключается в том, что современные виртуальные машины не обеспечивают достаточный уровень приватности из-за наличия уязвимостей [10].

Резервное копирование.

Пользователям cloud computing должны предоставляться гарантии быстрого восстановления информации в случаи ее потери. Данное условие выполняется за счет резервного копирования. При этом поставщики услуг облачных вычислений могут хранить около трех копий пользовательских данных.

Описанные выше пункты безусловно влияют на вопрос безопасности данных в cloud computing, при этом, с нашей точки зрения,проблема хранения данных имеет наибольший приоритет.

3. Распределенная файловая система HDFS

При организации облачных вычислений используется достаточно широкий набор современных технологий. В последнее время весьма большую популярность получил свободный Java-frameworkApacheHadoop и одновременно с ним распределенная файловая система HDFS (HadoopDistributedFileSystem). HDFS – это свободный аналог GFS (GoogleFileSystem). Как и GFS HDFS позволяет приложениям легко масштабироваться до уровня тысяч узлов и петабайт данных. На рис. 1 представлена обобщенная структура HDFS [11].

Как видно из рисунка работу всей системы координирует управляющий узел. Данный узел контролирует доступ к данным, а также управляет областью имен файловой системы. Все данные распределяются по узлам данных в виде отдельных блоков, при этом используется механизм репликации.

Использование HDFS в облачных вычислениях позволяет значительно увеличить степень безопасности хранения данных. Основная идея заключается в следующем: каждый файл разбивается на определенное количество блоков, которые хранятся на физически отдельных узлах, затем данные блоки шифруются с использованием отдельных ключей для каждого из них.Рассмотрим данный метод более детально.

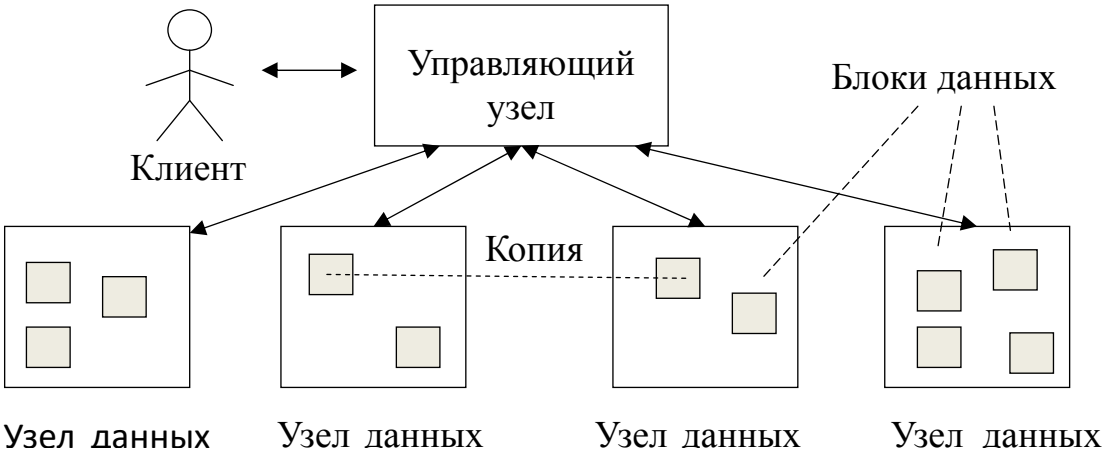

Рис. 1. Обобщенная структура HDFS

4. Модель безопасности хранения данных

Распределенную модель данных представим следующим образом:

$$K_f = f \cdot D_f.$$

f – вектор блоков, из которых состоит файл:

$$f = \{F_1, F_2, \ldots, F_n\}.\ F_i \cap F_j = \varnothing,$$
$$i \neq j; i, j = 1..n.$$

D_f – матрица размером nxL, где L – количество узлов данных. Каждый элемент матрицы может принимать значение 1 или 0, при этом 1 указывает на то, что F_i часть файла f размещена на данном узле, i= 1..n.

K_f – вектор состояния распределенного файла.

Для повышения степени безопасности необходимо использовать механизм шифрования, тогда данная модель будет иметь следующий вид:

$$D'_f = M \ and \ D_f,$$
$$K_f = E(f) \cdot D'_f.$$

M – матрица доступа данного пользователя.

Матрица D'_f – это матрица D_f с учетом прав доступа данного пользователя.

$E(f)$ – вектор зашифрованных блоков файла.

На рис. 3 представлено графическое отображение полученной модели в виде трех уровней защиты.

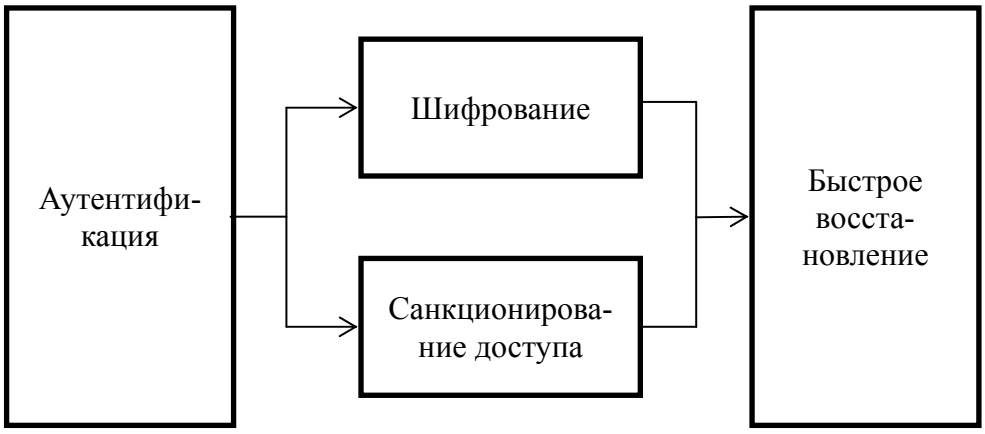

Рис. 2. Модель трехуровневой защиты данных

Первый уровень отвечает за аутентификацию пользователя путем проверки его цифрового сертификата. На втором уровне происходит шифрование данных и определение прав

доступа данного пользователя. На третьем уровне – быстрое восстановление информации в случае ее повреждения или потери.

5. Генерация ключей с использованием бинарного дерева

Как было показано выше, в облачных вычислениях часто используются распределенные файловые системы, в частности HDFS, при этом возможно использовать отдельное симметричное шифрование для каждого блока данных. Однако данный подход имеет существенный недостаток. Поскольку файл может быть разделен на значительное количество блоков, то пользователю также необходимо иметь соответствующее количество ключей. Для решения данной задачи предлагается использовать иерархию ключей.

Для каждого блока $\{F_1, F_2, \ldots, F_n\}$ $(i = 1..n)$ генерируется собственный уникальный ключ, при этом используется следующий принцип: следующий ключ в иерархии может быть получен как комбинация предыдущего ключа с определенной публичной информацией. При этом порождение следующего ключа происходит через хэш-функцию, которая удовлетворяет требование необратимости.

Существует множество способов построения дерева ключей, но для уменьшения количества вычислительных операций целесообразно использовать бинарное дерево. Предположим, что файл разбивается на n блоков $\{D_1, D_2, \ldots, D_n\}$, при этом $2^{p-1} \le n < 2^p$. Следовательно, необходимо построить бинарное дерево с высотой равной p (рис. 3).

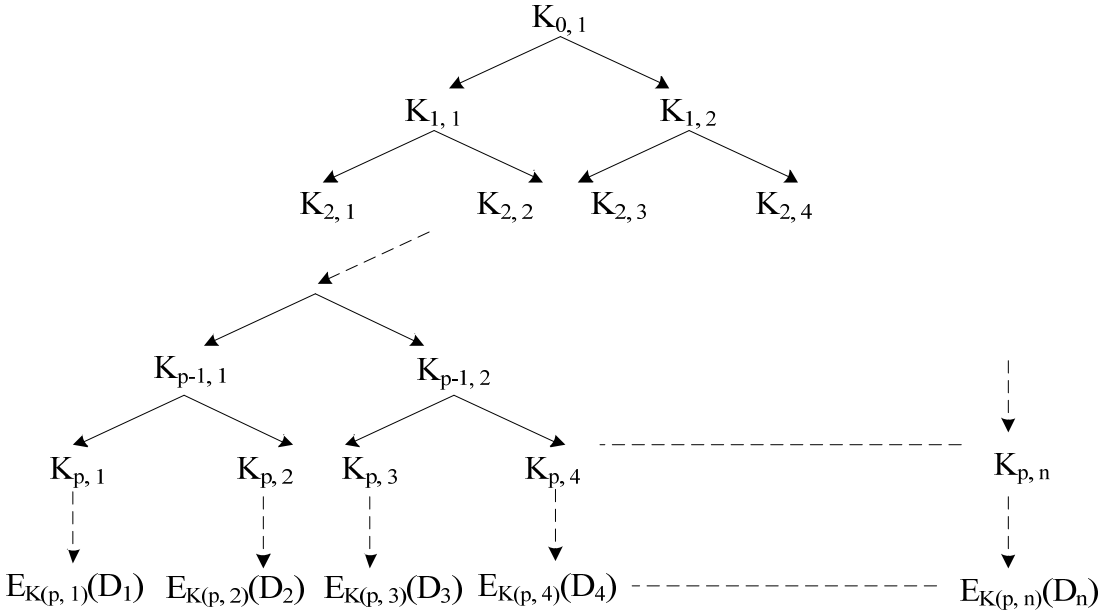

Рис. 3. Бинарное дерево ключей.

Корневой ключ $K_{0,1}$ генерируется пользователем, при этом существует возможность использовать любой алгоритм симметричного шифрования, например AES или IDEA. Первый индекс ключа определяет номер уровня в иерархи, второй – порядковый номер на данном уровне. Также пользователь выбирает порождающую хэш-функцию, которая должна удовлетворять требованию необратимости. Левый и правый производный ключ генерируются следующим образом:

$$K_{i+1,2\cdot j-1} = hash(K_{i,j} \,||\, (2\cdot j\text{-}1) \,||\, K_{i,j})$$

$$K_{i+1,2\cdot j} = hash(K_{i,j} \,||\, (2\cdot j) \,||\, K_{i,j})$$

Аргумент хэш-функции рассчитывается в результате двойной конкатенации ключа-родителя и номера текущего узла.

Процесс сохранения данных происходит следующим образом: пользователь сообщает «облаку» корневой ключ и хэш-функцию, менеджер доступа генерирует дерево, после чего полученные ключи с нижнего яруса иерархии используются в распределенной файловой системе для шифрации данных. Считывание данных происходит аналогичным способом.

6. Анализ вычислительных затрат

В качестве алгоритма хеширования будем использовать MD5, т.е. размер ключа для шифрования блоков будет составлять 128 бит. Сложность дешифрования данного ключа (B_1), при использовании механизма полного перебора, составит $O(2^{127})$ ($B_1 = 2^{127}$). Затраты на вычисление одного ключа –$O(64)$($H_1 = 64$).Для проведения анализа введем понятие эффективности шифрования:

$$E = \frac{B}{H},$$

где B – общая сложность дешифрации файла, а H – суммарная сложность вычисления ключей. В случае, когда весь файл кодируется одним ключом $B = B_1, H = 0$ (поскольку корневой ключ не вычисляется, а предоставляется пользователем).Значение эффективности при этом будет стремиться к бесконечности, по этому, данный случай является вырожденным и далее учитываться не будет.

Предположим, что размер пользовательского файла составляет 1 ГБ. Исходя из того, что размер блоков HDFS64 МБ, получаем 16 блоков ($n = 16$). Учитывая, что $2^{p-1} \le n < 2^p$ высота дерева будет равняться 5 ($p = 5$). Общая сложность дешифрации файла, в этом случаи, составит $B = 16 \cdot B_1$.Суммарная сложность вычисления ключей $H = 30 \cdot H_1$.Как результат получим следующее значение эффективности $E \approx 0.53 \cdot R$.Где $R = \frac{B_1}{H_1}$(данный коэффициент зависит от выбранного алгоритма хеширования).

Общие формулы имеют следующий вид:

$$B = n \cdot B_1;$$

$$H = (2^{p-1} - 2 + n) \cdot H_1.$$

Проведем расчеты, варьируя значениемколичества блоков. Результаты вычислений представлены в табл. 1.

Табл. 1. Значения параметра эффективности

N	***P***	***B/B1***	***H/H1***	***E/R***
2	2	2	2	1.00
3	3	3	5	0.60
4	3	4	6	0.67
5	4	5	11	0.45
6	4	6	12	0.50
7	4	7	13	0.54
8	4	8	14	0.57
9	5	9	23	0.39
10	5	10	24	0.42
11	5	11	25	0.44
12	5	12	26	0.46
13	5	13	27	0.48
14	5	14	28	0.50
15	5	15	29	0.52
16	5	16	30	0.53
17	6	17	47	0.36
18	6	18	48	0.38
19	6	19	49	0.39
20	6	20	50	0.40
21	6	21	51	0.41
22	6	22	52	0.42
23	6	23	53	0.43
24	6	24	54	0.44
25	6	25	55	0.45
26	6	26	56	0.46
27	6	27	57	0.47
28	6	28	58	0.48
28	6	28	59	0.49
30	6	30	60	0.50
31	6	31	61	0.51
32	6	32	62	0.52

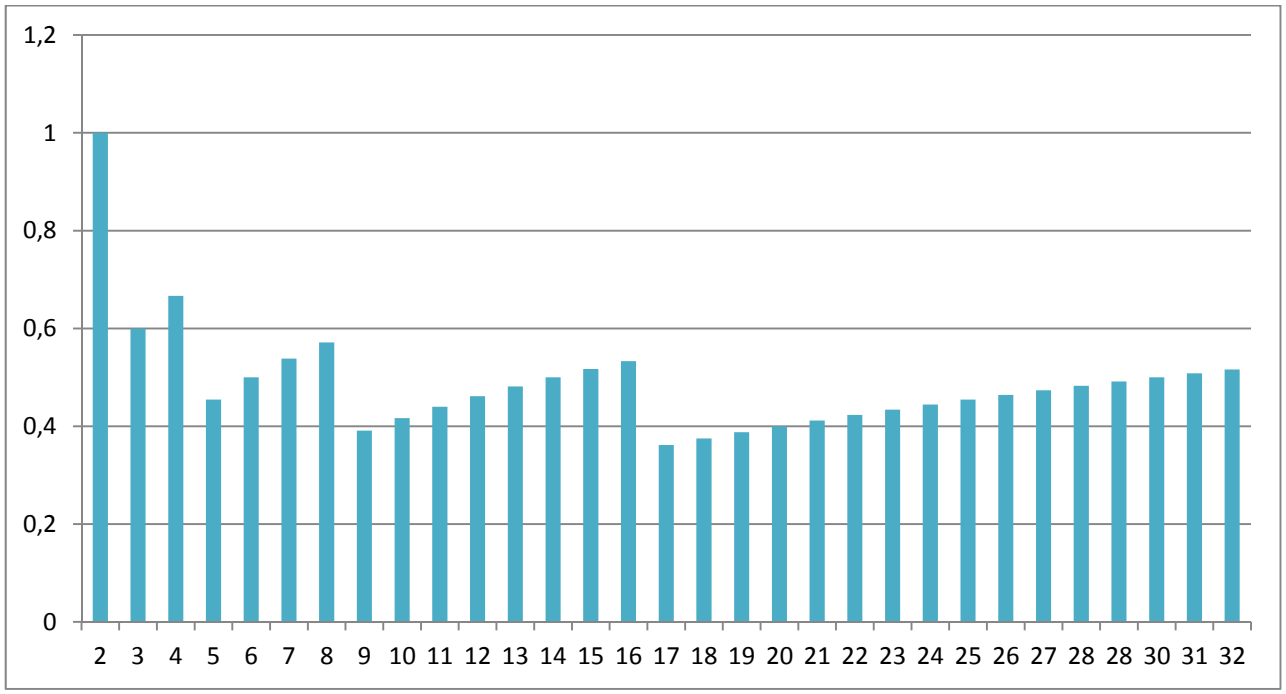

Рис. 4.Гистограмма эффективности шифрования.

Графическое представление результатов изображено на рис. 4. Ось абсцисс – количество блоков. Ось ординат – значение эффективности разделенное на *R*.

Общая сложность дешифрации файла при увеличении количества блоков увеличивается линейно. При этом значение эффективности шифрования меняется в пределах от 0.3*R* до 0.6*R* (при $n > 4$).

Исходя из данных результатов, можно сделать вывод, что при $n > 4$ соотношение общей сложности дешифрации файла к суммарной сложности вычисления ключей варьируется в пределах определенных границ (от 0.3*R* до 0.6*R)*. То есть значение количества блоков должно выбираться лишь с учетом ограничений на задержку процесса считывания или записи, размер блоков, а также размер корпоративных данных, которые требуют дополнительной защиты. В данной ситуации наибольший интерес представляют локальные максимумы, которые стремятся к 0.5*R*. Локальные максимумы достигаются при $n = 2^k$ ($k \in Z$).

7. Заключение

На примере распределенной файловой системы HDFSбыл предложен метод шифрования отдельных блоков файла с использованием бинарного дерева ключей,атакже был описан способ построения данного дерева.

Было показано, что данный метод позволяет увеличить сложность дешифрования файла пропорционально количеству блоков. При этом соотношение общей сложности дешифрования файла к суммарной сложности вычисления ключей меняется в пределах определенных границ. Локальные максимумы достигаются при $n = 2^k$ ($k \in Z$).

В некоторых случаях, например в распределенных файловых системах, размер блоков фиксирован и данный размер достаточно большой (в HDFS64 МБ), поэтому предложенный метод подходит лишь для случаев, когда жизненно важные корпоративные данные также достигают значительных размеров (несколько гигабайт).

В будущем, для уменьшения количества вычислительных операций возможно использование механизма кеширования, что позволит не вычислять целую иерархию ключей при каждой операции считывания или записи. Также возможно рассмотрение других вариантов иерархий ключей.

Список литературы

1. James Staten, Simon Yates, Frank Gillett, WalidSaleh, and Rachel A. Dines. Is Cloud Computing Ready For The Enterprise? //Forrester. – Cambridge, USA.– March 7, 2008.
2. Harry Katzan.Cloud Computing, I-Service, And IT Service Provisioning. //Journal of Service Science. – Savannah State University, USA – 2008 – Vol.5. – P.57-64.
3. Adam Holt, Keith Weiss, CFA1, Katy Huberty, CFA1, Ehud Gelblum. Cloud Computing Takes Off. Market Set to Boom as Migration Accelerates. //Morgan Stanley Research. – May 23, 2011.
4. Weichao Wang, Rodney Owens, Zhiwei Li, Bharat Bhargava. Secure and Efficient Access to Outsourced Data //CCSW'09 – Chicago, USA.– November 13 2009.
5. Jay Heiser, Mark Nicolett. Assessing the Security Risks of Cloud Computing. //Gartner Researche. – 3 June, 2008.
6. Yanpei Chen, Vern Paxson, Randy H. Katz. What's New About Cloud Computing Security? //Electrical Engineering and Computer Sciences, University of California at Berkeley. – January 20, 2010.
7. Ann Cavoukian. Privacy in the clouds. //Identity Journal. – 2008.
8. Закон України «Про захистперсональнихданих», редакція від 01.06.2010.
9. S. Subashini, V.Kavitha. A survey on security issues in service delivery models of cloud computing. // Anna University. – Tirunelveli, India. – 2010.
10. Lisa J. Sotto, Bridget C. Treacy, and Melinda L. McLellan. Privacy and Data Security Risks in Cloud Computing. //Electronic Commerce & Law Report. – February 3, 2010.
11. Dai Yuefa, Wu Bo, GuYaqiang, Zhang Quan, Tang Chaojing. Data Security Model for Cloud Computing. //International Workshop on Information Security and Application. – Qingdao, China.– November 21-22, 2009.