УДК 004.7

DOI: http://dx.doi.org/10.20535/2219-380413201568840

А. Л. Забияка[1], *студент*, **Ю. И. Барынин**[2], *студент*

СПОСОБЫ БЕЗОПАСНОЙ МАРШРУТИЗАЦИИ В *GRID*-СИСТЕМАХ

En The article gives an introduction to distributed computing (distributed computing) - a way to solve time-consuming tasks (computing and data processing tasks), namely, technology development and use of large grid systems, as well as secure routing algorithms between nodes in such systems. The focus of this article is on combining techniques of parallelization and encryption. Encryption can be done in two ways. In the first method of encoding a file will take place before the division into packets, and the second is to first split the file and then encrypt each piece otdelno.S security standpoint faster will be the second method, but the former is more optimized as it requires less resources and time for its implementation. When transferring large volume of data it is advisable to select the first encryption method, if security has priority over the speed of the task.

In the case of transmission through secure communication channels of small

[1] *Национальный технический университет Украины "Киевский политехнический институт", факультет информатики и вычислительной техники*

[2] *Национальный технический университет Украины "Киевский политехнический институт", факультет информатики и вычислительной техники*

volumes of data a second encryption method is better and faster. The smaller packets for transmission, the greater the reliability of this method in multipath routing mode.

Ua У статті дається введення в розподілені обчислення – спосіб вирішення трудомістких завдань (обчислювальних і завдань обробки даних), а саме, технології створення і використання великих грід-систем, а також алгоритми безпечної маршрутизації між вузлами в таких системах. Основна увага в даній статті приділяється комбінування методів розпаралелювання і шифрування.

Введение

Грид – это система, которая координирует распределенные ресурсы посредством стандартных, открытых, универсальных протоколов и интерфейсов для обеспечения нетривиального качества обслуживания (*QoS - Quality of Service*). Основной идеей, заложенной в концепции грид-вычислений, является централизованное удаленное предоставление ресурсов, необходимых для решения различного рода вычислительных задач. Также и в идеологии грид: мы можем запустить любую задачу с любого компьютера или мобильного устройств на вычисление, ресурсы же для этого вычисления должны быть автоматически предоставлены на удаленных высокопроизводительных серверах, независимо от типа нашей задачи. По сути грид – это будущее всех сетевых технологий, но требуется обеспечить безопасную, быструю и простую передачу данных между узлами [1].

В грид системах безопасность передачи данных играет важную роль в работоспособности и быстродействии. Более высокий уровень безопасности нагружает систему в целом, тогда как низкий уровень позволяет легко похитить данные, либо помешать их пересылке на конечный узел системы [2, 3].

Постановка задачи

Рассмотреть различные алгоритмы шифрования, а так же методы распараллеливания. Обеспечить безопасность передачи данных между узлами в грид системах, предотвратить взлом системы и другие хакерские атаки, при минимальных нагрузках на систему.

Подсистема безопасности

Технологии грид включают в себя:

– решения по безопасности, поддерживающие управление сертификацией и политиками безопасности, когда вычисления производятся несколькими организациями;

- протоколы управления ресурсами и сервисами, поддерживающие безопасный удаленный доступ к вычислительным ресурсам и ресурсам данных, а также перераспределение различных ресурсов;
- протоколы запроса информации и сервисы, обеспечивающие настройку и мониторинг состояния ресурсов, организаций и сервисов;
- сервисы обработки данных, обеспечивающие поиск и передачу наборов данных между системами хранения данных и приложениями.
- В этой статье мы проведем анализ алгоритмов шифрования на основе американского конкурса *AES.* Так же попробуем усовершенствовать идею шифрования и передачи данных через узлы в грид-системе.

Существует три способа обеспечения безопасности передачи данных между узлами:

- распараллеливание (формирование множества непересекающихся путей)
- шифрование данных [3]
- комбинированный способ (комбинация шифрования и распараллеливания)

В табл. 1 представлен анализ результатов конкурса *AES* (*Advanced Encryption Standard*).

***Таблица* 1.**

Сравнительные результаты

№	Критерии	*Serpend*	*Twofish*	*Mars*	*RC6*	*Rijindael*
1	Криптостойкость	+	+	+	+	+
2	Запас криптостойкости	++	++	++	+	+
3	Стойкость шифрования	-	+-	+-	+	+
4	Скорость расширения ключа	+-	-	+-	+-	+
5	Смарт-карты с большим обьемом ресурсов	+	+	-	+-	++
6	Аппаратная реализация	+	+	-	+-	+
7	Защита от атак на процедуру расширения ключа	+	+-	-	-	+
8	Защита от атак по потреблемой мощности	+	+	-	+-	+
9	Защитаот атак по времени выполнения и мощности	-	+-	-	-	+

№	Критерии	*Serpend*	*Twofish*	*Mars*	*RC6*	*Rijindael*
10	Возможность расширения ключа «на лету»	+	+	+-	+-	+-
11	Возможность параллельных вычислений	+-	+-	+-	+-	+

Криптостойкость всех этих алгоритмов достаточная, в процессе исследований не было обнаружено каких-либо практически реализуемых атак. Однако, эксперты *NIST* предупреждают, что данная оценка является весьма поверхностной и не может быть значимой при выборе алгоритма – победителя конкурса, но, тем не менее, отметили, что запас криптостойкости у *Rijndael* и *RC*6 несколько ниже, чем у остальных алгоритмов.

Лучше других поддерживают возможность параллельных вычислений алгоритмы *Rijndael* и *RC*6,а значит подходят для грид-систем.

Структура алгоритма *RC*6 облегчает анализ метода шифрования. Это самый быстрый из алгоритмов на 32-битных платформах. Процессы шифрования и расшифровывания в алгоритме *RC*6 практически идентичны. Скорость шифрования при программной реализации сильно зависит от того, поддерживает ли платформа 32-битное умножение и вращение на переменное число битов. *RC*6 сложно реализуем аппаратно и в условиях ограниченных ресурсов. Достаточно сложно защищается от атак по времени выполнения и потребляемой мощности. Распараллеливание вычислений при шифровании алгоритмом *RC*6 реализуемо с ограничениями, поэтому победителем конкурса *AES* стал именно алгоритм *Rijndael*.

Алгоритм *Rijndael* [4] позволяет шифровать данные не только 128-битными блоками, но и блоками по 192 или 256битов. Обрабатываемые данные могут представляться не только в виде массива размером 4 × 4, но и 4 × 6 или 4 × 8 для 192 и 256битных блоков соответственно, количество битов сдвига строк таблицы также зависит от размера блока. Поскольку для 192 и 256битного блоков увеличивается количество столбцов массива данных до 6 и 8 соответственно, в операции *AddRoundKey* участвуют уже 6 или 8 слов расширенного ключа вместо четырех. Следовательно, в *r*-м раунде алгоритма выполняется наложение слов расширенного ключа 3** ...*WW rNbrNb* + , где *Nb* — количество столбцов массива данных. В связи с вышесказанным, изменяется и процедура расширения ключа, однако модификация состоит лишь в том, что эта процедура должна выработать *RNb* +)1(* , а не *R* +)1(*4 слов расширенного ключа (что, впрочем, остается справедливым для 128-битного блока).

Сейчас при организации передачи информации в распределенных компьютерных системах широко используется многопутевая маршрутизация, которая объединяет несколько физических каналов в один многоканальный виртуальный путь. Суть комбинированного способа обеспечения безопасности состоит в том, чтобы «разбивать» зашифрованные пакеты в

точке отправки, с последующей их дешифровкой и сборкой в конечном узле.

Шифрование может осуществляться двумя способами. По первому способу, кодировка файла будет происходить перед разбиением на пакеты, а по второму – сначала разбиение файла, а потом шифрование каждой части отдельно. С точки зрения безопасности более надежным будет второй способ, но первый более оптимизирован, так как требует меньше ресурсов и времени на его реализацию.

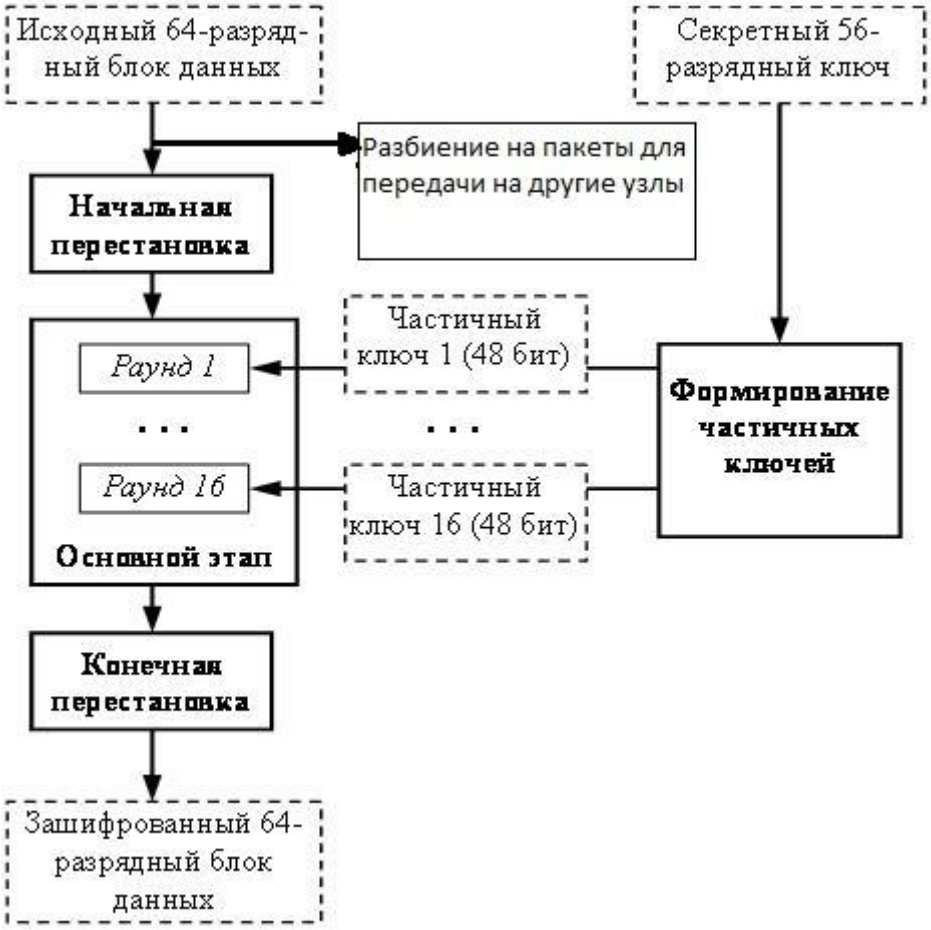

Рис. 1. Шифрование с предварительным разбиением [1]

На рис. 1 представлен алгоритм шифрования *Rijindale* с разбиением данных перед кодировкой. Это значит, что каждая отдельная часть будет зашифрована случайным образом и отправлена на другой узел согласно алгоритму многупутевой маршрутизации с непересекающимися путями. В конечной точке они будут дешифрованы и собраны в изначальный файл.

Преимущества данного способа – обеспечение более высокого уровня безопасности, так как все пакеты зашифрованы отдельно друг от друга.

Недостатки – кодировка всех пакетов по отдельности занимает больше времени и системных ресурсов.

На рис. 2 видно, что разбиение на пакеты происходит после шифрования блока данных. Другими словами, все части по-отдельности зашифрованы по одному принципу.

Преимущества: скорость шифрования гораздо выше, чем в первом способе, так как делается один раз. Так же этот способ снижает нагрузку на системные ресурсы.

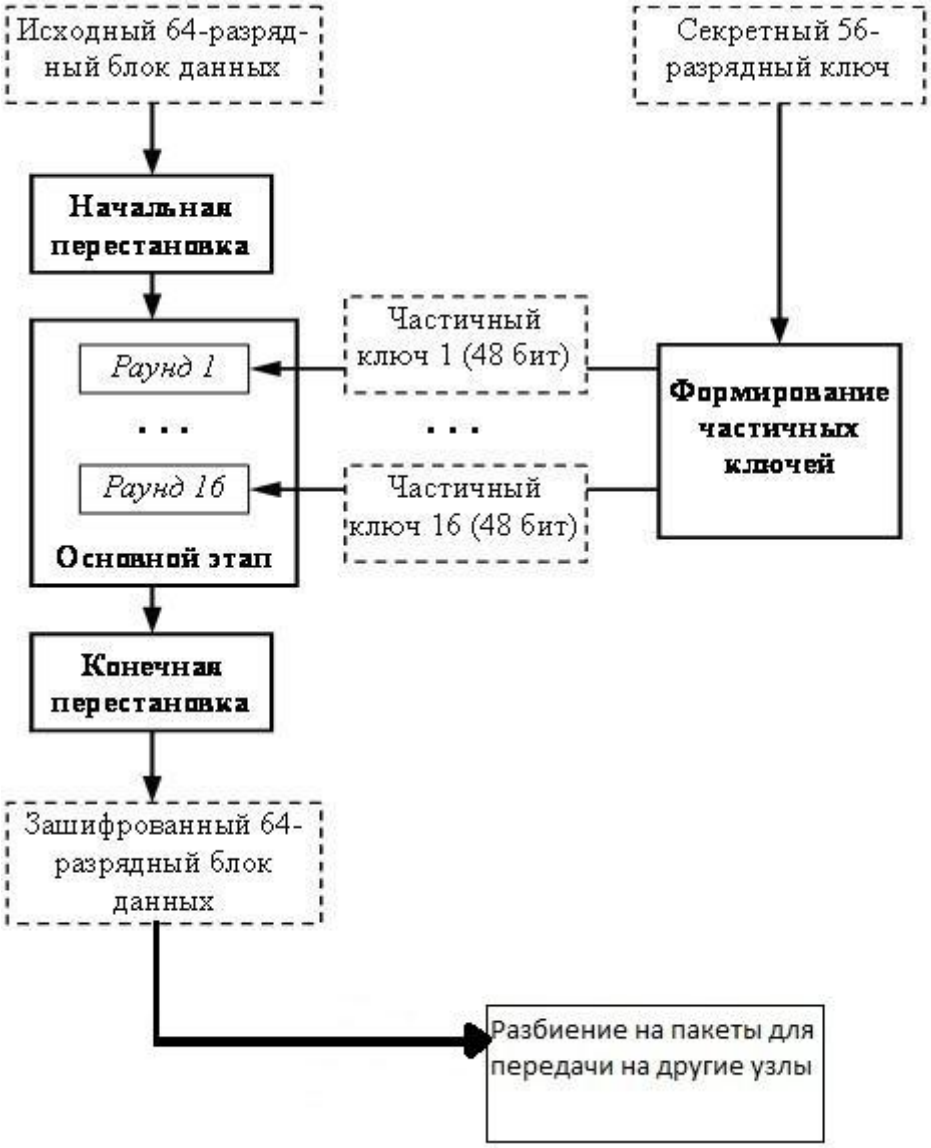

Рис. 2. Шифрование с последующим разбиением [1]

Недостатки: Безопасность данного метода ниже, так как пакеты зашифрованы одинаково и при вероятной хакерской атаке расшифровка одной части приводит к расшифровке всего файла целиком.

Выводы

На основе анализа предложенных алгоритмов шифрования можно сделать вывод, что для параллельных вычислений подходят не все методы

кодировки файлов. Некоторые работают с ограничениями. Алгоритм *Rijindael* больше всего подходит для работы в грид и других системах, где требуется многопутевая маршрутизация.

При передаче большого объёма данных целесообразно выбрать шифрование с предварительным разбиением, если безопасность имеет приоритет перед скоростью выполнения задачи.

В случае передаче по надежным каналам связи небольших объёмов данных шифрование с последующим разбиением лучше и быстрее. При этом, чем меньше пакеты для передачи, тем больше надежность данного метода в режиме многопутевой маршрутизации.

Список использованных источников

1. *Демичев А. П.* Введение в грид-технологии / А. П. Демичев – Москва: Препринт НИИЯФ МГУ - 2007 - 11/832, 2007.
2. *Kulakov Y.* The method of plurality generation of disjoint paths using horizontal exclusive scheduling /Y. Kulakov, A. Kogan // The science advanced. – 2014. – issue 9. – p.16-18.
3. *Lemeshko A. V.* Research on Tensor Model of Multipath Routing in Telecommunication Network with Support of Service Quality by Greate Number of Indices / A. V. Lemeshko, O. Yu. Evseeva, S. V. Garkusha/ 2015. P.
4. *Диброва М. А.* Способ конструирования трафика в Grid системах / М. А. Диброва, А. В. Коган, В. А. Куценко// Вісник Національного технічного університету України “КПІ”: Інформатика, управління та обчислювальна техніка. – К.: ТОВ “ВЕК+”, 2015. – Вип. 62. – С.65–69.
5. RFC 3036 LDP Specification. L. Andersson, P. Doolan, N. Feldman, A. Fredette, B. Thomas. January 2001. P.