A. A. Demyanchuk, N. A. Moldovyan, A. V. Ryzhkov

CHOOSING IDEAL PARAMETERS FOR ZERO-KNOWLEDGE AUTHENTICATION PROTOCOLS AND COMMUTATIVE CIPHERS

It has been proposed a method for commutative encryption and zero-knowledge two-step protocols with a special choice of parameters for this cryptoschemes. Advantage of the method is eliminating weak messages in case of commutative encryption and simplification of zero-knowledge protocols.

Encryption, commutative encryption, zero-knowledge protocol, information security

УДК: 20.53.19, 28.23.13

М. С. Куприянов, И. И. Холод, З. А. Каршиев

Метод построения параллельных алгоритмов интеллектуального анализа данных из потоконезависимых функциональных блоков

Описывается метод построения параллельных алгоритмов интеллектуального анализа данных, опирающийся на представление алгоритма в виде иерархии вложенных функциональных потокобезопасных блоков.

Интеллектуальный анализ данных, параллельные алгоритмы, интеллектуальный анализ распределенных данных

В соответствии с формальной моделью [1] алгоритм интеллектуального анализа данных (ИАД) можно декомпозировать на отдельные потокобезопасные функциональные блоки. Каждый блок должен получать на вход обрабатываемые данные, настройки алгоритма и созданную на предыдущих этапах модель знаний. При этом реализуется следующий принцип: данные и настройки не подлежат изменению – меняться может только модель знаний. Изменение модели знаний можно рассматривать как создание внутри блока новой модели знаний на основе переданных в него параметров: набора данных, настройки алгоритма и исходной модели знаний. Таким образом, каждый блок является «монолитным» с точки зрения основного алгоритма. По сути, он представляет собой мини-алгоритм.

Функциональные блоки являются потоко безопасными, если в процессе выполнения своих вычислений не модифицируют значения глобальных переменных. Для достижения этого требования при разбиении алгоритма на блоки необходимо соблюдать следующие правила:

- каждый блок может представлять собой или некоторую операцию, выполняемую над моделью, или некоторый структурный элемент, характерный для алгоритма ИАД;
- блок, реализующий некоторую операцию, должен изменять модель знаний, поданную на вход таким образом, чтобы она оставалась целостной (т. е. должны выполняться все ограничения, накладываемые на модель);
- внутри функционального блока не должно происходить обращений к внешним переменным, вся работа должна выполняться на основании набора данных, настроек и исходной модели знаний, переданных в функциональный блок.

Функциональный блок может содержать как исполняемый код (неразделимую операцию), так и последовательность других функциональных блоков. В общем случае блок может содержать не одну последовательность.

В результате алгоритм ИАД можно представить как иерархию вложенных функциональных потокобезопасных блоков (рис. 1). Сам алгоритм, являющийся корневым функциональным блоком

в данной структуре, также обладает свойством потокобезопасности.

Свойство потокобезопасности функциональных блоков позволяет выполнять их параллельно. По сути, алгоритмы, состоящие из таких потокобезопасных функциональных блоков, можно распараллеливать (распределять функциональные блоки по параллельным потокам) автоматически, подбирая параллельную структуру с лучшими характеристиками.

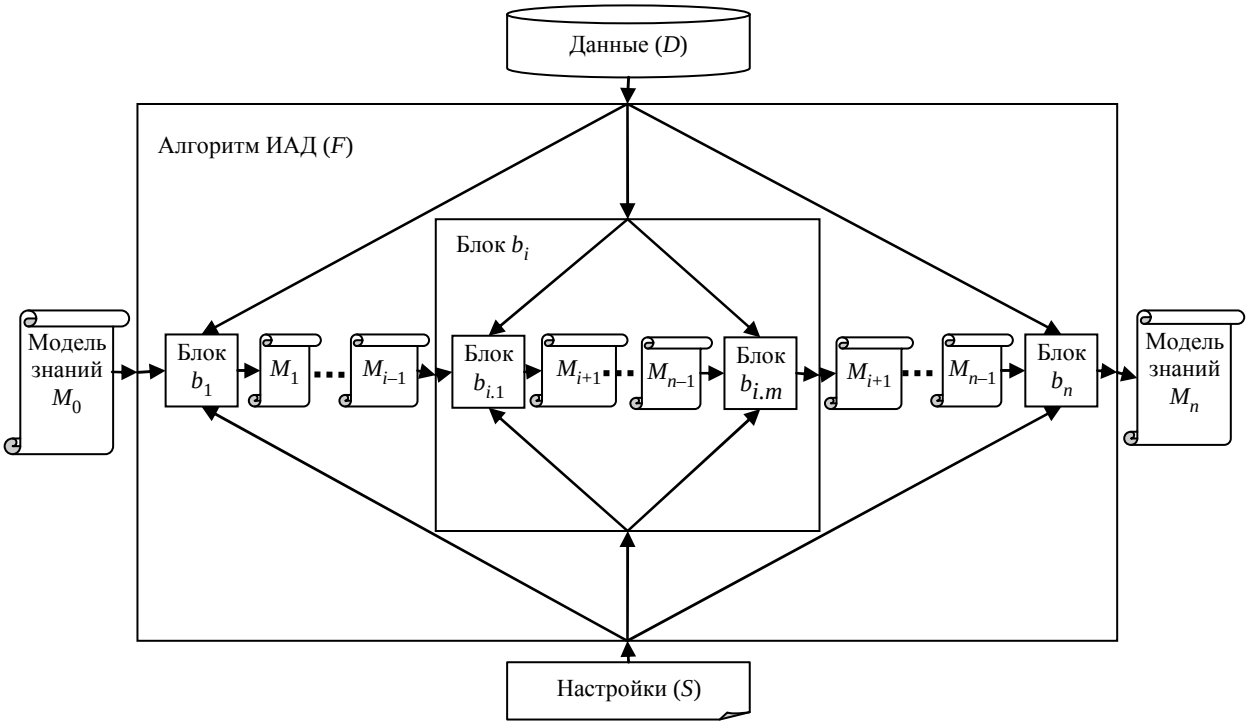

Рис. 1

Рассмотрим функциональные блоки, из которых могут состоять алгоритмы ИАД.

Линейный блок выполняет некоторую операцию над моделью знаний: добавление, удаление или изменение ее элементов. Не содержит вложенных блоков.

Блок цикла содержит вложенную последовательность блоков и определяет операции, выполняемые при инициализации цикла, до и после выполнения каждой итерации, а также условие выхода из цикла. В программной реализации данные функции должны выполняться в следующем порядке:

1) инициализация цикла;

2) проверка условия выхода из цикла;

3) функция, выполняемая перед итерацией;

4) выполнение вложенной последовательности блоков;

5) функция, выполняемая после итерации.

Блок решения содержит проверяемое условие и две последовательности блоков. По результатам проверки условия выполняется та или иная вложенная последовательность блоков.

Как уже отмечалось, одним из основных преимуществ разделения алгоритма ИАД на потоконезависимые функциональные блоки является возможность их параллельного выполнения. Для этих целей должны быть добавлены функциональные блоки, обеспечивающие распараллеливание алгоритмов ИАД. Рассмотрим, какие блоки необходимы для реализации различных видов параллельного выполнения.

Параллельный блок может содержать одну или больше последовательностей блоков. В случае параллелизации по данным он содержит одну последовательность блоков, которая должна клонироваться на несколько параллельных обработчиков. В случае параллелизации по задачам он содержит несколько последовательностей блоков (по одной на каждую ветвь алгоритма). Кроме того, данный блок может содержать вложенную последовательность блоков для диспетчеризации параллельных блоков. Унифицированный интерфейс данного блока должен описывать выполнение операций: разделения на потоки (split) и объединения (join). При этом порядок выполнения должен быть следующий:

1) выполнение операции разделения на параллельные блоки (split);

2) запуск блока диспетчера (если он есть);

3) запуск параллельных блоков;

4) выполнение операции объединения (join).

Блок-отправитель включает в себя вложенный блок, выполняющий подготовку данных для их отправки в другие, параллельно выполняющиеся параллельные блоки. Унифицированный интерфейс должен описывать операцию отправки данных. Последовательность выполнения должна быть следующая:

– выполнение вложенного блока;

– выполнение операции отправки.

Блок-получатель включает в себя вложенный блок, выполняющий обработку данных после их получения от других параллельно выполняющихся блоков. Унифицированный интерфейс должен описывать операцию получения данных. Последовательность выполнения должна быть следующая:

– выполнение операции получения;

– выполнение вложенного блока.

При декомпозиции алгоритма ИАД в иерархическую структуру потокобезопасных функциональных блоков, представленную на рис. 1, необходимо выполнить следующие действия:

1) декомпозировать алгоритм ИАД на элементарные операции;

2) сгруппировать в функциональные блоки действия, выполняющие дискретное изменение модели знаний;

3) выделить структурные элементы и вложенные в них последовательности функциональных блоков;

4) добавить в структуру функциональные блоки для параллельного выполнения.

На первом шаге структура алгоритма должна быть декомпозирована на элементарные операции, выполняющиеся над моделью знаний: добавление, удаление и изменение отдельных элементов модели знаний. При этом в структуре алгоритма должны быть явно выделены основные управляющие операторы: циклы, условные переходы и др. Декомпозиция должна выполняться с учетом следующих условий:

– после выполнения каждой операции сохраняется целостность модели знаний;

– количество переменных, используемых различными операциями, должно быть минимизировано и локализовано (переменные должны использоваться только для следующих друг за другом операций).

При выполнении декомпозиции крайними случаями являются:

– монолитный алгоритм – когда декомпозировать алгоритм ИАД на составные потокобезопасные функциональные блоки не удалось и алгоритм представляется одним потокобезопасным функциональным блоком;

– слабосвязанный алгоритм – когда каждую операцию алгоритма можно представить отдельным потокобезопасным функциональным блоком.

На данном шаге допускается наличие элементарных операций, не изменяющих модель знаний.

На втором шаге действия группируются в функциональные блоки, выполняющие дискретное изменение модели знаний. Группировка должна выполняться исходя из набора правил:

1. В функциональный блок группируются операции, связанные между собой общими локальными переменными. Базовыми в таких блоках являются операции, изменяющие состояние модели знаний, к которым добавляются предшествующие и/или последующие операции, не меняющие модель. Группировка операций, не меняющих модель, со смежными базовыми операциями выполняется исходя из их логической принадлежности к ним. Признаком такой логической связи может служить использование промежуточных (локальных) переменных. Такие переменные должны быть локализованы внутри создаваемого функционального блока.

2. Выделяемый функциональный блок должен целиком входить в управляющий оператор (условие, цикл и др.). Например, в одном функциональном блоке не должно быть операций, выполняющихся как в цикле, так и вне цикла.

На третьем шаге формируются структурные блоки. Управляющие операторы (циклы, условные переходы и т. п.) не изменяют состояния модели, однако они должны быть выделены в отдельные функциональные блоки. В них должны быть включены функциональные блоки, операции которых выполняются внутри управляющих операторов.

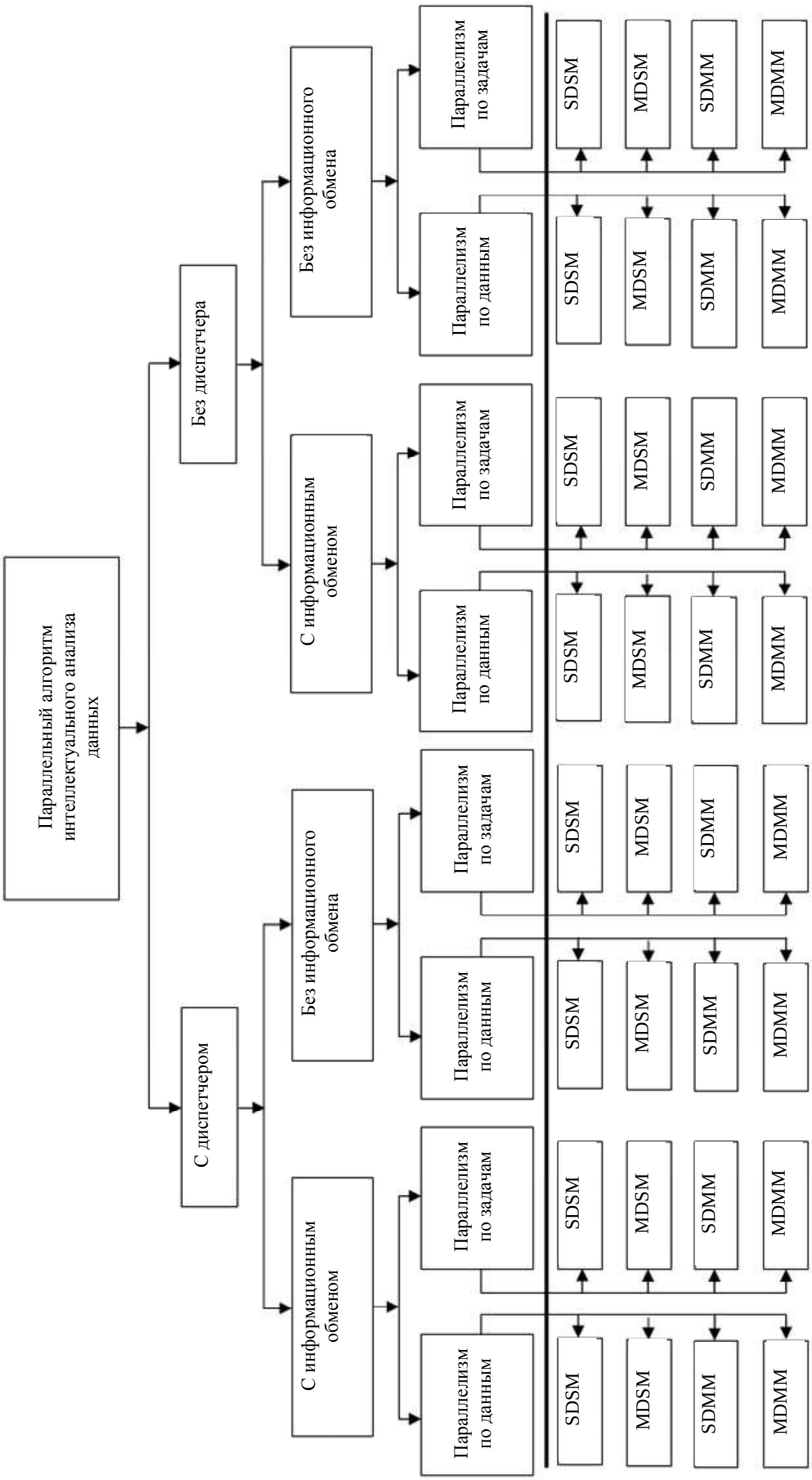
Параллельный алгоритм интеллектуального анализа данных
С диспетчером
Без диспетчера
С информационным обменом
Без информационного обмена
Параллелизм по данным
Параллелизм по задачам
SDSM
MDSM
SDMM
MDMM

Рис. 2

На последнем шаге для построения параллельного алгоритма ИАД в полученную структуру могут быть введены структурные блоки, обеспечивающие параллельное выполнение:

– параллельный блок;

– блок-отправитель;

– блок-получатель.

Учитывая свойство потокобезопасности полученных функциональных блоков, параллельный блок может быть добавлен в любое место иерархии. При этом должны быть определены функциональные блоки, входящие в параллельный блок как вложенная последовательность. Основным условием такого вхождения является то, что все функциональные блоки, входящие в параллельный блок, должны относиться к одному уровню иерархии.

Параллельный блок может включать в себя и функциональные блоки самого верхнего уровня иерархии, т. е. в этом случае параллельно будет выполняться весь алгоритм. При этом блоки, находящиеся под ним, будут вложены в параллельный блок.

Блоки отправителя и получателя должны добавляться совместно в параллельные последовательности функциональных блоков, вложенных в параллельный блок, если требуется обмен информацией между этими последовательностями.

Типы структур параллельных алгоритмов ИАД, которые можно построить предложенным методом, классифицированы как показано на рис. 2 [2]. При этом способы параллелизации верхнего уровня связаны со структурными изменениями алгоритмов – перемещением параллельных блоков в структуре функциональных блоков алгоритмов ИАД. Типы параллелизации нижнего уровня можно изменить за счет настройки алгоритмов ИАД.

Таким образом, описанный метод построения параллельных алгоритмов ИАД позволяет получить различные типы параллельных структур алгоритмов ИАД [2]:

– с параллелизацией по задачам/по данным;

– с/без взаимодействия между параллельными ветвями для систем с распределенной памятью;

– с/без диспетчера.

Для построения алгоритма ИАД с диспетчером необходимо выделить в соответствующую последовательность параллельного блока или существующие функциональные блоки алгоритма, или добавить новые функциональные блоки, решающие задачи диспетчеризации. При отсутствии диспетчерской последовательности функциональных блоков в параллельном блоке будет получен параллельный алгоритм без диспетчера.

При формировании в параллельном блоке последовательностей функциональных блоков из одинаковых блоков (клонирование блоков) и разделения между ними набора данных будет получена параллельная структура алгоритма ИАД с параллелизацией по данным. Если последовательности параллельного блока будут содержать разные функциональные блоки, обрабатывающие одни и те же данные, будет получена структура с параллелизацией по задачам.

Структуры параллельных алгоритмов ИАД с взаимодействием и без могут быть получены соответственно добавлением или нет функциональных блоков отправителя и получателя.

Данная статья и проведенная научная работа поддержаны в рамках Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг. (контракт 14.B37.21.0589).

СПИСОК ЛИТЕРАТУРЫ

1. Куприянов М. С., Каршиев З. А. Формальная модель выполнения алгоритмов интеллектуального анализа //Изв. СПбГЭТУ «ЛЭТИ». 2012. Вып. 9. С. 60–68.

2. Холод И. И., Каршиев З. А. Методика распараллеливания алгоритмов интеллектуального анализа данных // Изв. СПбГЭТУ «ЛЭТИ». 2013. Вып. 3. С. 38–45.

M. S. Kupriyanov, I. I. Kholod, Z. A. Karshiyev

METHOD OF CREATING PARALLEL ALGORITHMS OF INTELLECTUAL ANALYSIS OF DATA USING FLOW-INDEPENDENT FUNCTIONAL BLOCKS

The paper describes a new method of creating parallel algorithms of intellectual analysis of data. The method is based on representing the algorithm using the hierarchy of embedded functional flow-safe blocks.

Intellectual analysis of data, parallel algorithms, intellectual analysis of distributed data