УДК 004.056.001.891 (045)

З 988.0-018.4

Харченко В. С. д-р техн. наук
Халин М. Б.

МНОГОВЕРСИОННОСТЬ ДЛЯ ОБЕСПЕЧЕНИЯ ЦЕЛОСТНОСТИ ИНФОРМАЦИИ: ЭКСПЕРИМЕНТЫ И РЕЗУЛЬТАТЫ

Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ»

Предлагаются методы и структурные решения для реализации многоверсионной цифровой подписи с использованием диверсных хэш-функций. Приводятся результаты машинного эксперимента по тестированию таких решений.

Реализация теста стойкости диверсной хэш-функции. Проведем экспериментальную проверку (верификацию) результатов, полученных в предыдущей статье. Целью реализации теста стойкости диверсной хэш-функции является:

- подтверждение или опровержение полученных ранее теоретических зависимостей и свойств;
- сравнение полученных оценок с аналогичными оценками одноверсионной (стандартной) хэш-функции.

Генератор псевдослучайных чисел, предложенный в [2] используется для создания случайных сообщений длиной 128 бит. В качестве алгоритма блочного шифрования используется *RC*5-64/12/8. Алгоритм *RC*5 используется потому, что он имеет гибкую структуру и на его основе можно создать не только генератор псевдослучайных чисел, но и различные хэш-функции. Конкретная версия *RC*5 обозначается *RC*5-*w*/*r*/*b*, где:

1) w – размер слова в битах. *RC*5 шифрует данные блоками длин в 2 слова. Допустимые значения: 16, 32, 64;

2) r – число раундов. Допустимые значения: 0, 1, …, 255;

3) b – число байтов в секретном ключе K. Допустимые значения: 0, 1, …, 255.

Период счетчика N равен 2^{64}. Данное значение взято с большим запасом, т. к. согласно парадоксу задачи о днях рождения при длине дайджеста хэш-функции 32 бита, понадобится перебрать 2^{16} различных случайных сообщений для того, чтобы с вероятностью 0,5 обнаружить хотя бы одну коллизию относительно данной хэш-функции. Учитывая, что тождественные сообщения также можно считать случайными, то понадобится перебрать 2^{16} коллизий для первой хэш-функции, прежде чем будет с вероятностью 0,5 найдена коллизия для диверсной хэш-функции. Таким образом, для диверсной хэш-функции необходимо перебрать 2^{32} сообщений, прежде чем с вероятностью 0,25 будет найдена одна коллизия.

Диверсная хэш-функция на основе метода сцепления блоков. Целый ряд предложений, касающийся функций хеширования, был основан на использовании техники сцепления шифрованных блоков, но без использования секретного ключа. Разделим сообщение M на блоки фиксированной длины M_1, M_2, ..., M_N и используем любую систему блочного шифрования, чтобы вычислить хэш-код G следующим образом:

$$H_0 = \text{начальное значение},$$
$$H_i = E_{M_i}[H_{i-1}], \qquad (24)$$
$$G = H_N.$$

В данной работе в качестве блочного алгоритма шифрования используется *RC*5-16/12/8. Отличие двух алгоритмов хеширования состоит в различном начальном значении H_0 (значения даны в двоичном представлении): $H_0^1 = 0101...01$, $H_0^2 = 1010...10$. Длина H_0 равна 32 би-

там, так как такую же длину имеет дайджест, вырабатываемый обеими хэш-функциями.

Общее описание программы теста. Тест стойкости диверсной хэш-функции заключается в получении опытным путем показателя сильной сопротивляемости коллизиям, т.е. в нахождении любых двух сообщений, которые бы имели одинаковый диверсный дайджест, а также подсчет количества сообщений, которые необходимо перебрать для достижения цели. Так как диверсная хэш-функция состоит из двух стандартных хэш-функций, то нахождение коллизии можно разделить на две подзадачи:

1) нахождение коллизии для первой хэш-функции;

2) проверка на образование коллизии для второй хэш-функции, сообщениями, которые образовали коллизию для первой хэш-функции.

Данная реализация атаки на диверсную хэш-функцию основана на парадоксе задачи о днях рождения. В свою очередь, для того, чтобы выполнить условия, описанные в задаче о днях рождения, необходимо реализовать случайную генерацию сообщений подаваемых на вход диверсной хэш-функции.

Описание версий программной реализации теста. В процессе тестирования (верификации) были учтены аспекты, связанные с ограничением числа и мощности аппаратных средств в постановке задачи и реализации данного эксперимента, а также выявлены ошибки в программной реализации теста диверсной хэш-функции, и. Это привело к последовательному созданию нескольких версий программной реализации теста, описание которых приводится далее.

1. Версия 1.0. Реализованы основные функции:

а) генератор псевдослучайных сообщений на основе алгоритма *RC*5-32/12/8. Таким образом генерация одного сообщения происходит не за один такт работы счетчика, как предполагалось при использовании криптоалгоритма *RC*5-64/12/8, а за два. Переход к алгоритму *RC*5-32/12/8 связан со сложностью реализации *RC*5-64/12/8 стандартными средствами *Microsoft Visual C++* 6.0;

б) две хэш-функции на основе алгоритма *RC*5-16/12/8. Они отличаются только одним параметром – начальным вектором H_0;

в) функция нахождения коллизии диверсной хэш-функции. Все найденные коллизии как первого, так и второго уровня записываются в файлы по выбору пользователя.

2. Версия 1.1:

а) оптимизирован расход оперативной памяти компьютера необходимой для создания базы данных дайджестов первой хэш-функции, что привело к увеличению количества обрабатываемых сообщений в секунду в два раза;

б) улучшен пользовательский интерфейс программы.

3. Версия 1.2: несмотря на увеличение скорости работы программы, перебрать необходимое количество сообщений для нахождения конкретного значения показателя сильной сопротивляемости коллизиям не представлялось возможным при имеющихся аппаратных ресурсах. Поэтому было принято решение об использовании нестандартной модификации алгоритма *RC*5-8/12/8.

4. Версия 1.3:

а) исправлена ошибка обнаруженная при тестировании предыдущей версии программы. Коллизии, найденные с использованием широкого диапазона сообщений, не включают в себя коллизии, найденные с использованием более узкого диапазона. Это происходило из-за того, что при нахождении n сообщений с одинаковыми дайджестами первого уровня, программа сигнализировала о нахождении n коллизий, а не $\sum_{i=1}^{n}(i-1)$ коллизий. Данный факт существенно повлиял на результаты эксперимента, например, при переборе 400 тысяч сообщений программа обнаружила 7 и 17 коллизий второго уровня для версий 1.2 и 1.3 соответственно;

б) добавлена возможность отказа от сохранения коллизий первого уровня в файле, что уменьшает время на проведение эксперимента.

Анализ результатов эксперимента. Результаты, полученные для первых двух версий теста, не отличаются: для диверсной хэш-функции на основе криптоалгоритма *RC5*-16/12/8 не было найдено ни одной коллизии второго уровня при переборе 80 млн. сообщений. Следует учесть, что было найдено 400 тысяч коллизий первого уровня. Однако, по результатам этого эксперимента нельзя судить о стойкости диверсной хэш-функции, т. к. как следует из формулы (23) для его логического завершения необходимо перебрать около 4 млрд. сообщений. Причем данный перебор необходимо повторить несколько десятков раз с различными сообщениями для набора статистики достаточной для вычисления точного значения показателя сильной сопротивляемости коллизиям.

Результаты эксперимента приведены в табл.1. В данном эксперименте диапазон сообщений от 1 до 10^7 сообщений был разделен на 10 равных поддиапазонов, в каждом из которых были найдены двумя способами коллизии первого уровня. Из табл. 1 видно, что количество найденных коллизий без обнуления базы данных (БД) дайджестов в девять раз больше, чем количество найденных коллизий с обнулением БД дайджестов. Данный результат свидетельствует о невозможности параллельного проведения эксперимента на независимых компьютерах.

Таблица 1. Результаты эксперимента с использованием версий 1.0 и 1.1

№	Кол-во коллизий без обнуления БД	Кол-во коллизий, найденное за диапазон	Кол-во коллизий с обнулением БД	Кол-во потерянных коллизий на каждом этапе
1	133	133	133	0
2	464	331	100	231
3	1033	569	91	478
4	1806	773	116	657
5	2890	1084	110	974
6	4144	1254	123	1131
7	5685	1541	129	1412
8	7393	1708	106	1602
9	9275	1882	100	1782
10	11514	2239	116	2123

Наиболее полные и точные результаты были получены при использовании программы теста версии 1.3. Для получения численного значения показателя сильной сопротивляемости коллизиям было найдено 30 коллизий диверсной хэш-функции. Среднее количество сообщений, которые необходимо перебрать для нахождения одной коллизии второго уровня, примерно равно $9 \cdot 10^4 \approx 2^{16.5}$, что соответствует показателю сильной сопротивляемости коллизиям для одноверсионной хэш-функции, которая имеет дайджест длиной 32 бита.

Так как примерно равны показатели сильной сопротивляемости коллизиям для одноверсионной и диверсной хэш-функций при условии равенства длины дайджестов, вырабатываемых в первом и втором случае, то и мощность множеств тождественных сообщений относительно этих хэш-функций равны. Соответственно верна и формула (23).

Варианты реализации одно- и многоверсионной цифровой подписи. Многоверсионная цифровая подпись (МЦП) может генерироваться как с использованием различных версий самого алгоритма ЦП, так и с использованием многоверсионной хеш-функции (МХФ). Приведем примеры использования многоверсионной ЦП:

1. При передаче информации по одному каналу возможны различные варианты применения одно- и многоверсионной ЦП, показанные на рис. 1. Первый вариант (ОЦП, ОХФ-1) – это стандартная одноверсионная ЦП, использующая стандартную одноверсионную хэш-функцию. Ее показатель сильной сопротивляемости коллизиям равен $2^{\frac{m}{2}}$, а в остальных примерах он в $2^{\frac{(n-1)m}{2}}$ раз больше. Среди примеров 2-5 самым нестойким к аналитическим атакам является третий. Появление одной аналитической атаки или на алгоритм хэш-функции, или на алгоритм ЦП существенно понизит стойкость всей ЦП. Второй пример является более стойким, далее следуют четвертый и пятый. Таким образом, чем больше версий алгоритмов хэш-функций и ЦП используется для МЦП тем она менее подвержена влиянию новых аналитических атак.

При передаче информации по нескольким каналам возможны различные варианты применения одно- и многоверсионной ЦП, показанные на рис. 2. Первый вариант (рис. 2) соответствует стандартной схеме резервирования. Во втором примере показан вариант использования принципа многоверсионного алгоритма ЦП и многоверсионной хэш-функции одновременно. Стойкость такой ЦП в $2^{\frac{(n-1)m}{2}}$ раз больше, чем стойкость ЦП в первом примере. Стоит также обратить внимание, что количество передаваемых данных по каналам при этом не увеличивается. Чтобы достигнуть той же стойкости с помощью резервирования, необходимо создать ЦП, показанную в третьем примере. Используя такую технологию, необходимо передавать ЦП, которая в n раз больше, чем в первом и втором примерах. Структура (МЦП, МХФ) меньше подвержена влиянию аналитических атак на один из используемых в ней алгоритмах. Кроме того, при успешной атаке криптоаналитика на $\left\lfloor \frac{n-1}{2} \right\rfloor$ алгоритмов или хэш-функции, или ЦП возможно определить, на какие именно алгоритмы производилась атака и заменить их на более стойкие или же сменить ключи.

Систему, показанную во втором примере, можно применять, если необходимо повысить стойкость СЗИ, интегрированных в критические системы с высокой отказоустойчивостью. Понятие «критические системы» обычно связывают с такими техническими комплексами как морские и воздушные суда, аэрокосмические системы, АЭС и другими, отказ которых ведет к авариям, катастрофам и экологическим бедствиям. В последнее время к числу критических (бизнес-критических) стали относить системы, используемые в биржевой и банковской деятельности, поскольку отказ и сбой в них приводит к большим денежным потерям [4].

2. Технология *STRATUS* (фирма *Hewlett Packard*) является частным случаем примера 2 при n=4, причем каналы при этом попарно связаны между собой функцией, которая пропускает информацию с 2 каналов, если она идентична, и не пропускает в противном случае. При исправности всех каналов показатель сильной сопротивляемости коллизиям равен 2^{2m}. Если же сработала только одна пара каналов, то показатель сильной сопротивляемости коллизиям равен 2^{m}.

Выводы. Полученные результаты говорят о несомненном превосходстве многоверсионной цифровой подписи, которая вырабатывается на основе МХФ, перед n-ированной. Если одноверсионная дублированная хэш-функция имеет длину дайджеста m бит для каждого канала передачи информации, то для МХФ необходимо перебрать в $2^{\frac{(n-1)m}{2}}$ раз больше сообщений для нахождения коллизии между двумя случайными сообщениями. Также повышается сопротивляемость новым аналитическим атакам. Предположим, что одна из многоверсионных хэш-функций использует тот же алгоритм, что и одноверсионная, а другая – алгоритм, базирующийся на математике иного вида. Тогда при появлении аналитической атаки на одноверсионную хэш-функцию, показатели стойкости последней понизятся больше, чем аналогичные показатели МХФ.

При передаче информации по одному каналу показатели сильной сопротивляемости коллизиям равны для одноверсионной и МХФ с равными длинами дайджестов. Если предположить, что первая из диверсных хэш-функций использует тот же алгоритм, что и одноверсионная, а вторая – алгоритм, базирующийся на математике иного вида, то при появлении аналитической атаки на одноверсионную хэш-функцию, показатели стойкости последней понизятся больше, чем аналогичные показатели диверсной хэш-функции.

Дальнейшие исследования целесообразно проводить в направлении системного анализа всех возможных вариантов применения многоверсионности для защиты информации (многоверсионность как средство и многоверсионные системы как объект защиты), а также реализации принципа построения гарантоспособных систем из негарантоспособных элементов как обобщения результатов [7].

Список литературы

1. *S. Garfinkel, G. Spafford.* WebSecurity, Privacy, and Commerce, Beijing, Cambridge, Farnham, Köln, Paris, Sebastopol', Taipei, Tokio, 'Reilly, 2002. – 756 p.

2. *A. Avižienis, J-C. Lapri, B. Randel.* Fundamental Concepts of Dependability, UCLA CSD Report no.010028, LAAS Report no.01-145, Newcastle University Report no. CS-TR-739, 2002. – 20 p.

3. *Многоверсионные* системы, технологии и проекты. Под ред. В. С. Харченко, Минобразования и науки Украины, НАКУ «ХАИ»: Харьков, 2003. – 486 с.

4. *B. Littlewood, L.Strigini.* Redundancy and diversity in security http://www.csr.city.ac.uk/people/lorenzo.strigini/ls.papers/_03_FTsecurity/.

5. *Харченко В. С., Скляр В. В., Сидоренко Н. Ф.* Многоверсионные технологии в системах защитымнформации и отказоустойчивых web-приложениях// Информационно-управляющие системы на железнодорожном транспорте, 2003, № 5. – С. 43-45.

6. *Харченко В. С., Халин М. Б.* О многоверсионной цифровой подписи//Материалы МНТК «Информационные технологии в машиностроении», НАКУ «ХАИ»: Харьков, 2003. – С. 24.

7. *J. E. Dobson, B. Randell.* Building Reliable Secure Computing Systems out of Unreliable Insecure Components, Proceedings of the Conference on Security and Privacy, Oakland, USA, IEEE, 1986, pp. 187-193.

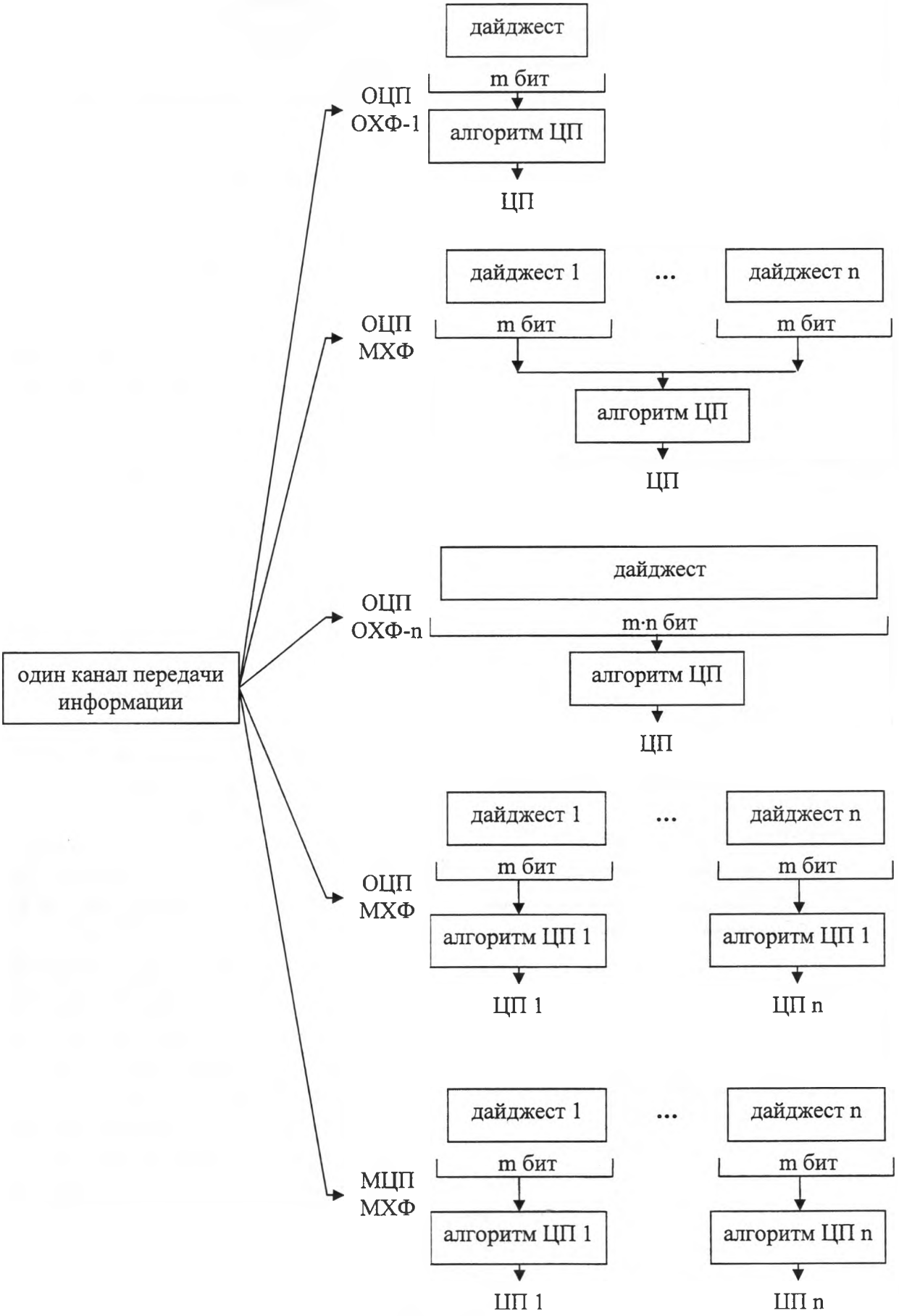

Рис. 1. Варианты применения одно- и многоверсионной ЦП при передаче информации по одному каналу

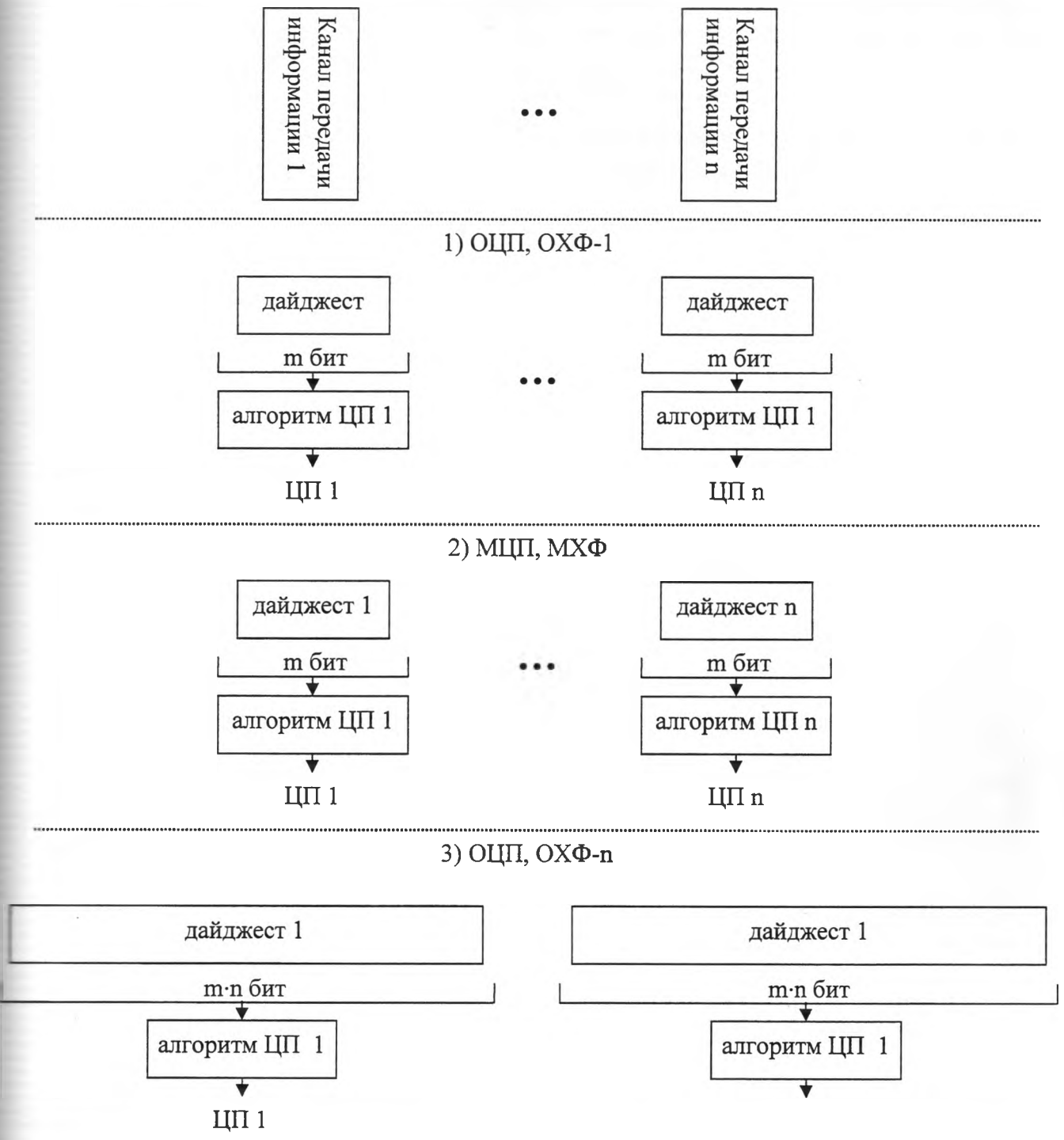

Рис. 2. Варианты применения одно- и многоверсионной ЦП при передаче информации по нескольким каналам