УДК 004.738

БЕЗОПАСНАЯ ОБРАБОТКА ЗАПРОСОВ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ НА ОСНОВЕ CRYPTOGRAPHIC MESSAGE SYNTAX

Р.Э. Асратян
Институт проблем управления им. В.А. Трапезникова РАН
Россия, 117997, Москва, Профсоюзная ул., 65
E-mail: rubezas@yandex.ru

Ключевые слова: распределенные системы, Интернет, web-технологии, web-сервисы, защита данных, криптосистема, безопасный сетевой канал.

Аннотация: Рассмотрены принципы реализации сетевой службы - Protected Message Service (PMS) - предназначенного для обработки защищенных запросов в распределенных информационных системах. Отличительной особенностью PMS является тесная интеграция функций аутентификации и защиты данных с функциями сетевого обмена информацией и обработки данных. Суть подхода заключается в применении стандарта Cryptographic Message Syntax (CMS) в качестве основы защищенного представления данных в сети. Рассматриваются принципы использования CMS для построение защищенных сетевых каналов для web-сервисов.

1. Введение

Интернет-стандарт Cryptographic Message Syntax (CMS), анонсированный в документах RFC 3852 и RFC 5653 [1, 2], еще в нулевые годы привлек внимание разработчиков распределенных информационных систем. Он описывает структуру данных, представляющую собой своего рода защищенный «контейнер» для хранения электронных документов, обеспечивающий аутентификацию и конфиденциальность данных [3]. Самое важное свойство CMS заключается в том, что он предоставляет стандартный программный интерфейс для формирования и проверки электронных подписей, а также для шифрования и расшифровки данных, который можно использовать для работы не с определенной криптосистемой, но с множеством криптосистем, поддерживающих этот стандарт (в число последних входит и ряд криптосистем российских провайдеров). Если Средства защиты информации (СЗИ) в информационной системе опираются на CMS, то появляется возможность оперативно переключаться с одной криптосистемы на другую без какой-либо доработки программных модулей системы (см. рис. 1). Очевидно, что это свойство особенно ценно в распределенных системах, в которых удаленные узлы могут администрироваться независимо.

Тем не менее, в литературе до сих пор отсутствуют сведения о конкретных системах, построенных на основе CMS, о методах применения этого стандарта в разработках распределенных систем и об исследованиях их эффективности. По-видимому, это объясняется нехваткой в CMS сетевых свойств, адаптирующих его условиям Интернета. Данная работа представляет собой попытку «восполнить» этот пробел. В ней рассматриваются два альтернативных подхода к использованию CMS:

- создание сетевой службы, и сетевого протокола, специально ориентированных на применение CMS для защиты данных в сети;
- построение безопасного сетевого канала для обслуживания уже существующих сетевых протоколов (например, HTTP), основанного на инкапсуляции информационных запросов и ответов в защищенную структуру CMS [4].

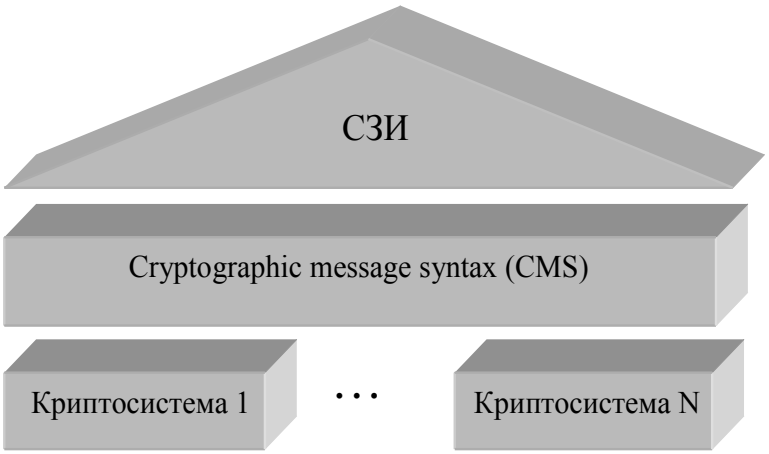

Рис. 1. Архитектура средств защиты данных на основе CMS.

2. Сетевая служба защищенных сообщений на базе CMS

Служба защищенных сообщений PMS (Protected Message Service) предназначена для организации безопасных взаимодействий в распределенных системах на основе использования CMS [5]. Основная идея ее создания заключается в тесном объединении средств защиты данных и средств сетевого взаимодействия в общей системе программных классов, основными из которых являются:

- класс PMSmessage, описывающий защищенное сообщение (представляющий собой «надстройку» над главным классом CMS «signedCMS»);
- класс PMSconnection, описывающий сетевое соединение.

Как и всякая сетевая служба, PMS включает две основные компоненты: сервер PMS, в котором хранятся библиотеки сервисных функций, и клиентская библиотека PMS, открывающая клиентскому модулю доступ к определениям классов [6,7]. Рассмотрим фрагмент кода клиентского модуля на языке C#, в котором проиллюстрирован вызов сервисной функции MyFunc из библиотеки MyLib на сервере MyServer и передачу ей информационного запроса (например, в формате XML). В коде имеются строки комментариев, начинающиеся с двух слешей.

```
//создание защищенного сообщение и загрузка в него документа запроса
PmsMessage Query = new PmsMessage("<query> … </query>");
//формипрвание двух удостоверяющих электронных подписей
PmsCertList SenderCerts = PmsCertList(new string [] {"Иванов", "Петров"});
Request.AddSignatures(SenderCerts);
//открытие сетевого соединения с сервером MyServer
PmsConnection MyConn = new PmsConnection();
MyConn.Connect("MyServer");
//Получение сертификата открытого ключа от сервера
PmsCertList ReceiverCert = MyConn.GetServerCertificate();

//Вызов сервисной функции с шифрацией документа ключем сервера и загрузкой
//результата в защищенное сообщение Reply
PmsMessage Reply=Query.Process (MyConn, "MyLib.MyFunc", ReceiverCert);
```

Использованные во фрагменте кода методы классов PmsMessage и PmsConnection (AddSignatures, Process и т.п.) опираются на методы основных классов CMS (рис. 2).

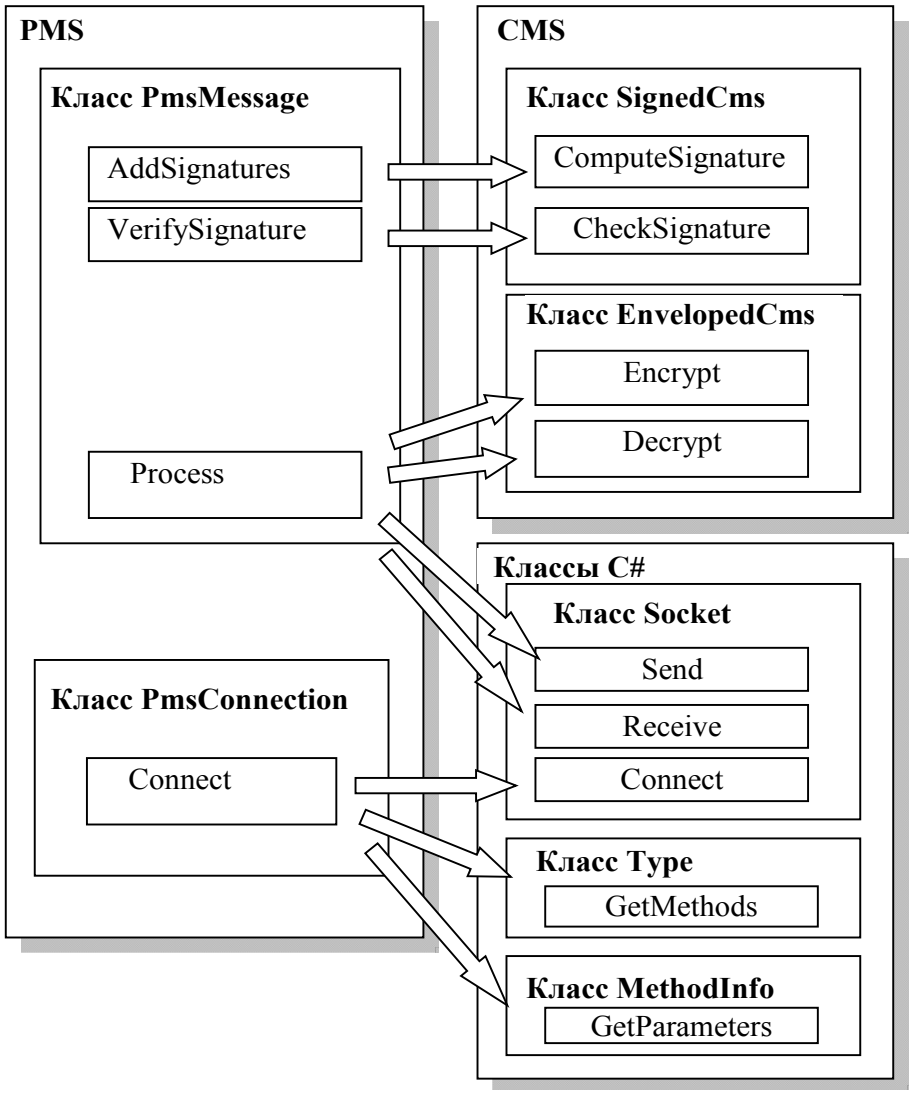

Рис. 2. Соотношение методов классов PMS и CMS.

3. Защищенный сетевой канал для web-сервисов на базе CMS

Интерес к средствам защиты данных в Интернете растет с каждым годом [8-10]. Сегодня почти все значимые сайты переключились на защищенный протокол HTTPS. Пожалуй, единственной областью, где обычный HTTP еще востребован, остаются распределенные системы, построенные на основе web-сервисов [11]. Однако, такие сервисы требуют защищенных каналов взаимодействия для безопасной работы в сети [12, 13]. Один из возможных подходов к построению такого канала основан на использовании CMS для временной инкапсуляции запросов к web-серверам (HTTP/SOAP-запросов) и ответов в структуру защищенного сообщения, осуществляемой с помощью серверов-посредников – клиентского и серверного шлюзов (рис. 3). Главное преимущество такого подхода целиком «наследуется» от CMS: возможность гибкого выбора крипто-средств без переделки клиентских и серверных модулей.

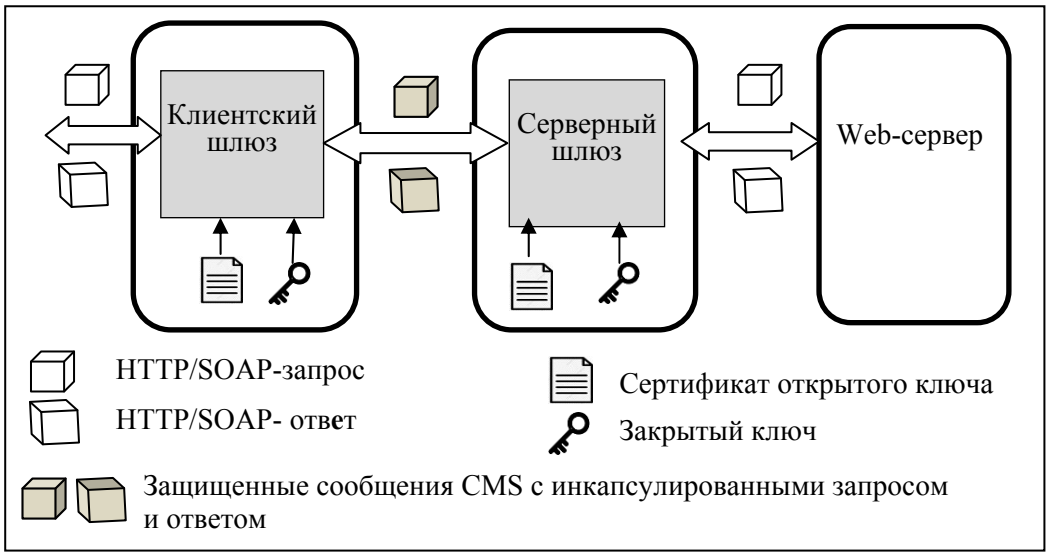

Рис. 3. Структура защищенного канала на основе CMS.

4. Оценка быстродействия

Применение CMS подразумевает введение дополнительного звена между СЗИ и крипто-средствами (рис. 1). Это с неизбежностью ставит вопрос об оценке дополнительной задержки, которое вносит это звено. На рис. 4 приведены результаты сравнительного экспериментального исследования двух реализаций PMS: с жесткой привязкой к конкретной криптосистеме (PMS-Crypto) и с использованием CMS для гибкого выбора криптосистемы (PMS-CMS-Crypto). В данном эксперименте измерялась полное время обращения к быстрой серверной функции со временем выполнения 100 мс при различных величинах сетевого трафика (2, 50 и 100 Кб). В качестве криптосистемы использовалась «КриптоПро 3.6». Как видно из рисунка, использование CMS внесло заметную задержку, но в пределах 16%.

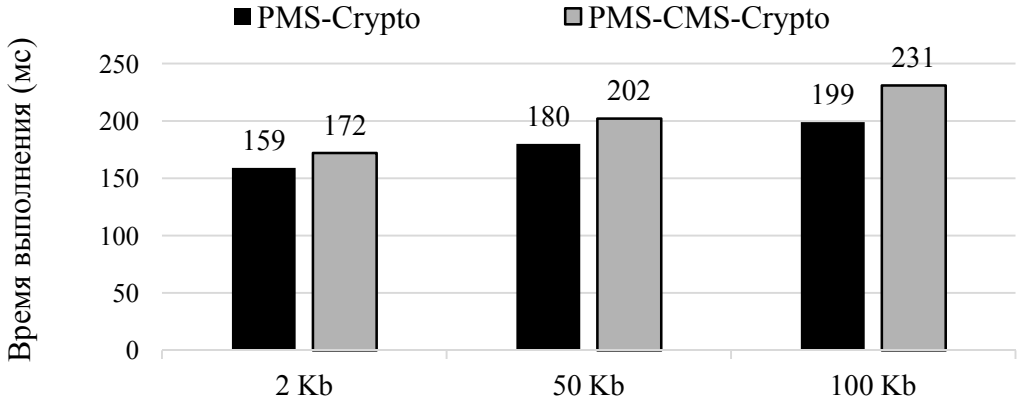

Рис. 4. Сравнение быстродействия двух реализаций PMS.

6. Заключение

Оба рассмотренных метода использования CMS (сетевая служба и защищенный канал) в полной мере «наследуют» его главное свойство – возможность гибкого выбора используемых крипто-средств без переделки программных модулей. В процессе экспериментов автор намеренно использовал на серверах и рабочих станциях разные криптосистемы и «взаимодействующие стороны» прекрасно понимали друг друга. Разумеется, стандартный высокоуровневый интерфейс к крипто-средствам, предоставляемый CMS, значительно упрощает разработку СЗИ, так как избавляет от необходимости изучать программный интерфейс конкретной криптосистемы.

Программная поддержка CMS для языка программирования C# имеется как в Windows (среда разработки MS Visual Studio), так и в Linux (среда разработки MonoDevelop) [14]. Экспериментальные реализации службы PMS и защищенного канала для Windows и Linux были выполнены с использованием этих средств.

Список литературы

1. https://www.protokols.ru/WP/rfc5652 (дата обращения 27.11.2023).
2. https://www.protokols.ru/WP/rfc3852 (дата обращения 27.11.2023).
3. Козлов А.Д., Орлов В.Л. Методы и средства обеспечения информационной безопасности распределенных корпоративных систем. М.: ИПУ РАН, 2017. 156 с.
4. Шапошников И.В. Web-сервисы Microsoft .NET. СПб: БХВ-Петербург, 2002. 336 с.
5. Асратян Р.Э. Интернет-служба защищенной обработки информационных запросов в распределенных системах // Программная инженерия. 2016. № 11. С. 490-497.
6. Снейдер Й. Эффективное программирование TCP/IP. Библиотека программиста. СПб.: Символ-Плюс, 2002. 320 с.
7. Хант К. TCP/IP. Сетевое администрирование. СПб.: Питер, 2007. 816 с.
8. Салимова Ш.А. Кибербезопасность в России: актуальные угрозы и пути обеспечения в современных условиях // Достижения вузовской науки 2021: сборник статей XVII Международного научно-исследовательского конкурса. Пенза, 20 января 2021 года. Пенза: Наука и Просвещение, 2021. С. 207-214.
9. Жаранова А.О., Птицына Л.К. Анализ влияния распределенности на качество функционирования комплексных систем защиты информации // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2020): Сборник научных статей IX Международной научно-технической и научно-методической конференции. СПб: СПБГУТ, 2020. С. 324-327.
10. Згоба А.И., Маркелов Д.В., Смирнов П.И. Кибербезопасность: угрозы, вызовы, решения // Вопросы кибербезопасности. 2014. № 5. С. 30-38.
11. Мак-Дональд М., Шпушта М. Microsoft ASP.NET 3.5 с примерами на C# 2008 и Silverlight 2 для профессионалов. М.: Вильямс, 2009. 1408 с.
12. Акушуев Р.Т. Принцип работы VPN и его особенности // Modern Science. 2020. № 7. С. 312-314.
13. Baka P, Schatten J. SSL/TLS under lock and key: a guide to understanding SSL/TLS cryptography. Keyko books, 2020. 132 p.
14. Negus C. Linux Bible. Wiley, 2015. 912 p.