

## Шифрование и дешифрование сервера Spring Cloud Config с использованием репозитория Git

*Ервлева Регина Викторовна*

*Приамурский государственный университет имени Шолом-Алейхема  
Студент*

*Ервлев Павел Андреевич*

*Приамурский государственный университет имени Шолом-Алейхема  
Студент*

### Аннотация

В данной статье рассмотрен процесс шифрования и дешифрования сообщений. Шифрование и дешифрование происходит с использованием сервера Config Server в Spring Cloud. Итоговым результатом является возможность шифровать и дешифровать сообщения.

**Ключевые слова:** SHA, Java, Spring

## Encrypting and Decrypting Spring Cloud Config Server Using Git Repository

*Eroleva Regina Viktorovna*

*Sholom-Aleichem Priamursky State University  
Student*

*Erolev Pavel Andreevich*

*Sholom-Aleichem Priamursky State University  
Student*

### Abstract

This article discusses the process of encrypting and decrypting messages. Encryption and decryption happen using the Config Server in Spring Cloud. The end result is the ability to encrypt and decrypt messages.

**Keywords:** SHA, Java, Spring

Шифрование сообщений – это значимая часть современных приложений. Оно используется с целью защиты при возможных кражах данных. Также необходимо и дешифровывать данные, когда они поступают человеку, которому их и отправили.

Цель данной статьи – зашифровать и дешифровать сообщения с использованием Config Server.

В своей работе В.П. Великов, К.С. Добрева рассмотрели проблемы автоматизированной генерации ПО [1]. В своей статье А.А. Шейн описал

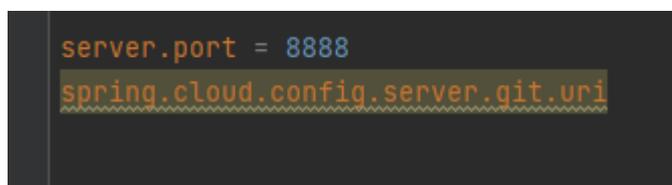
процесс разработки программы, которая автоматически создает наборы классов в виде нативных объектов Java [2]. А.В. Колойдчук рассмотрела методы программирования кредитных рисков внедрения ИКТ с помощью Java, Pascal и Basic [3]. А.А. Симаков описал разработку трассировки для JVM программ в использовании анализа ПО и обратного проектирования [4]. А.Н. Иванов описал процесс реализации безопасного веб-приложения на Java [5].

Будем использовать проект «Spring Cloud Config Server» для предварительной подготовки как симметричной, так и асимметричной криптографии с небольшим количеством конфигураций.

Также будем использовать локальный git в качестве внутреннего репозитория для получения файла конфигурации для приложения Spring. Прежде всего, для приложения «Config Server» можно напрямую загрузить его из «Spring Initializer» с зависимостью «Config Server».

Теперь нужно настроить локальный репозиторий git в качестве внутреннего репозитория для файлов конфигурации Spring. Для этого, создадим новый каталог в файловой системе и с помощью команды «git init» инициализируем репозиторий git.

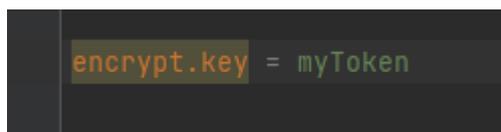
Теперь настроим загруженный проект Spring Boot с помощью этого локального git в качестве репозитория для файлов конфигурации. Для этого: добавим аннотацию «@EnableConfigServer» в класс приложения Spring Boot и добавим свойство «spring.cloud.config.server.git.uri» в файл свойств приложения с каталогом репозитория конфигурации (рис.1).



```
server.port = 8888
spring.cloud.config.server.git.uri
```

Рисунок 1 - Файл конфигурации

Для шифрования и дешифрования с помощью криптографии с симметричным ключом установим свойство «encrypt.key» в файле свойств приложения в пути к классам. Сервер внутренней конфигурации выберет эту переменную среды и будет использовать ее для симметричной криптографии (рис.2).



```
encrypt.key = myToken
```

Рисунок 2 – Добавление свойства

«Config Server» предоставляет 2 конечные точки с путем «/encrypt» и «/decrypt» для шифрования (рис.3) и расшифровки (рис.4) данных соответственно.

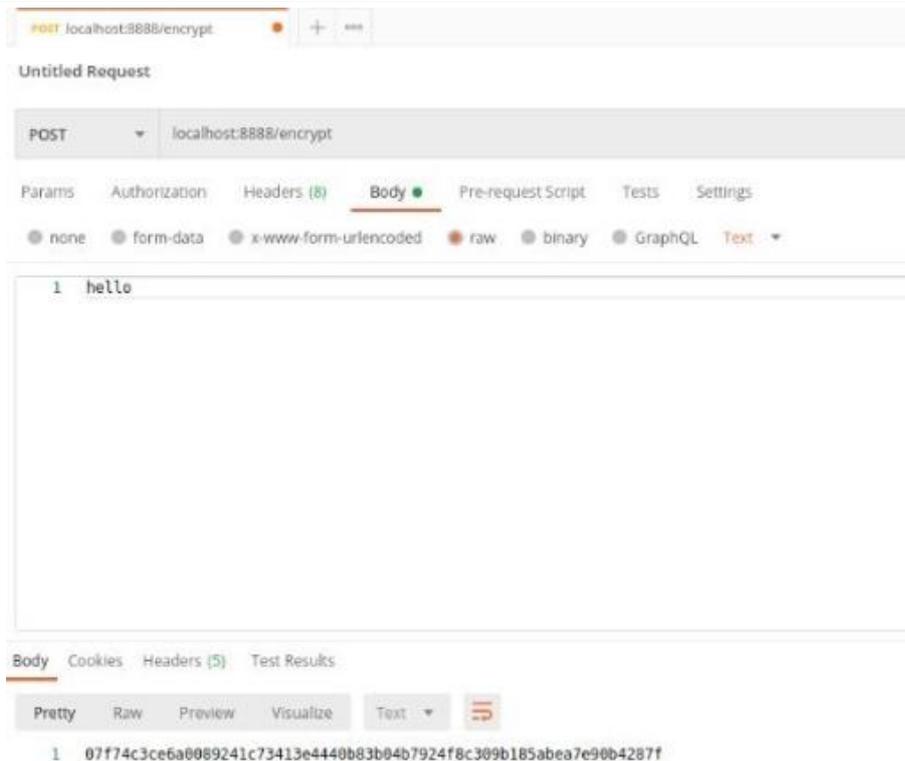


Рисунок 3 – Симметричное шифрование

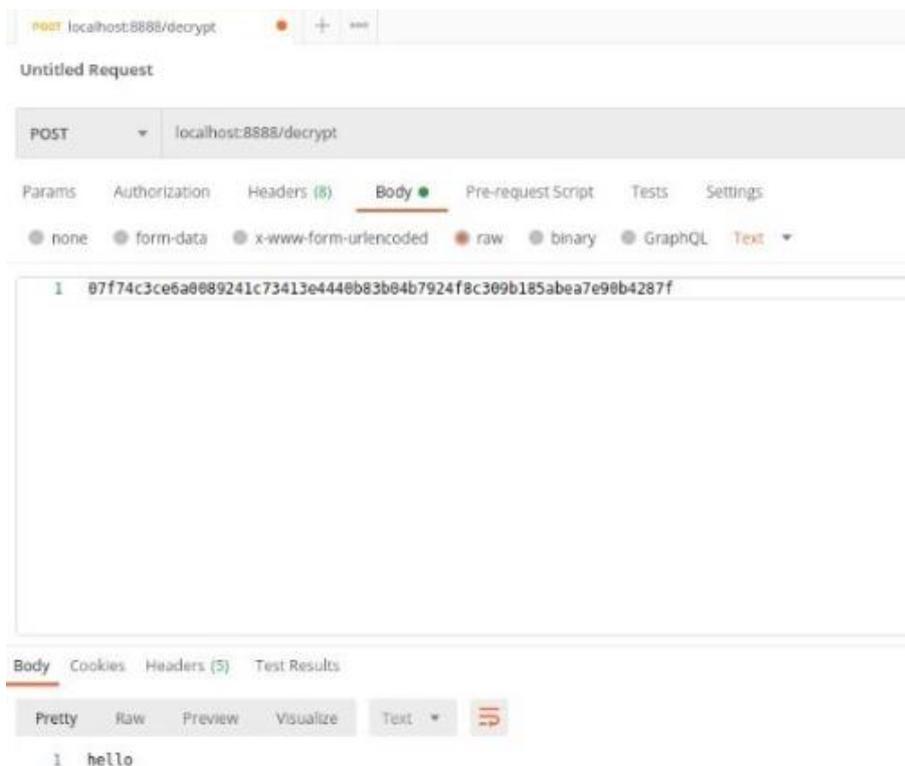


Рисунок 4 – Симметричное дешифрование

Для асимметричной криптографии нужны пары закрытых и открытых ключей для выполнения шифрования и дешифрования. Можно использовать программу «keytoo»<sup>1</sup> для генерации пар ключей для асимметричной

криптографии. «Keytool» — это утилита командной строки, похожая на «openSSL», с помощью которой можно создавать пары криптографических ключей, цепочки сертификатов X.509 и самозаверяющие сертификаты.

Сначала создадим пару криптографических ключей для выполнения асимметричного шифрования и дешифрования.

Можно использовать следующую команду для создания пар ключей RSA в формате jks. Это создаст пару закрытого и открытого ключей в указанном файле хранилища ключей (рис.5).

```
> -keyalg RSA \  
> -dname "cn=Config-Serv, ou=Java, o=Spring, c=US" \  
> -alias configserv \  
> -keypass nopass \  
> -keystore simple.jks \  
> -storepass nopass  
Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 90 days  
for: CN=Config-Serv, OU=Java, O=Spring, C=US
```

Рисунок 5 – Генерация ключей

Перед запуском приложения необходимо убедиться, что этот файл доступен в пути к классам. Затем необходимо установить свойства в файле «application.properties» для выполнения асимметричного шифрования и дешифрования (рис.6).

```
server.port = 8888  
  
spring.cloud.config.server.git.uri = uri  
  
encrypt.keyStore.location=classpath://simple.jks  
  
encrypt.keyStore.password=nopass  
  
encrypt.keyStore.secret=nopass  
  
encrypt.keyStore.alias=configserv  
  
encrypt.keyStore.type=jks
```

Рисунок 6 – Файл конфигурации

Как только эти свойства установлены, можно запустить приложение и оно будет использовать пары ключей из файла jks и выполнять шифрование (рис.7) и дешифрование (рис.8).

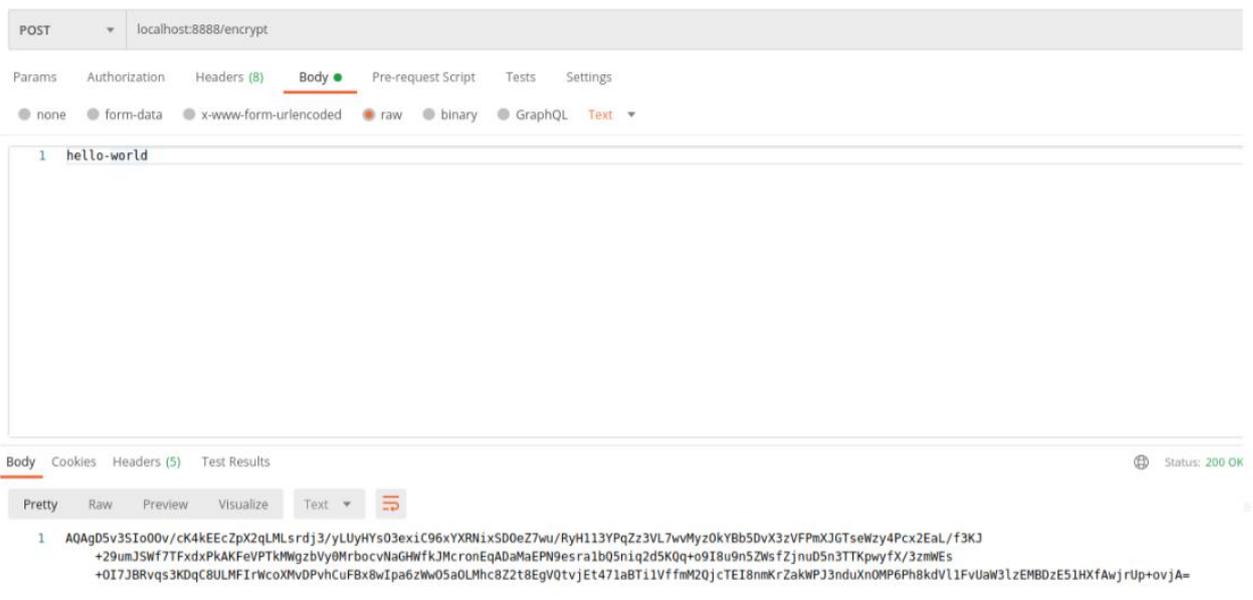


Рисунок 7 – Ассиметричное шифрование

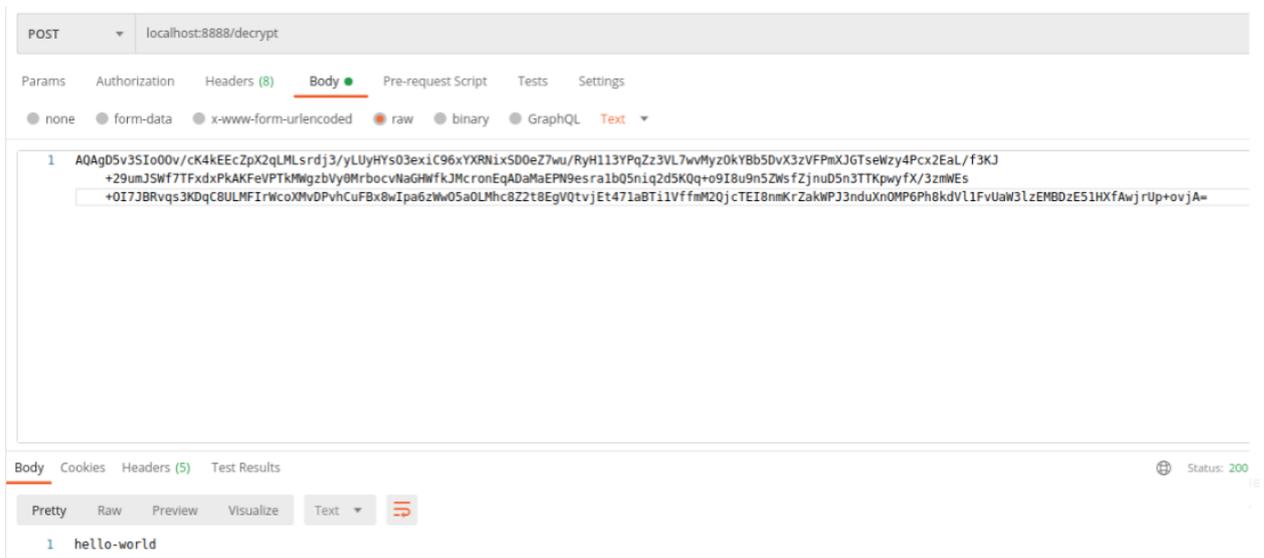


Рисунок 8 – Ассиметричное дешифрование

Вот как можно выполнять как симметричное, так и асимметричное, шифрование и дешифрование с использованием Spring Cloud Config Server.

### Библиографический список

1. Великов В.П., Добрева К.С. Генератор из диаграмм классов java в исходный код java // Информационные системы и технологии: управление и безопасность. 2014. №3. С. 14-23.
2. Шейн А.А. Генератор исходного кода на языке java по описанию бортовых компонентов decode // Современные проблемы науки и образования. 2016 №8. С. 18-25.
3. Колойдчук А.В. Программирование инвестиционных и кредитных рисков

- 
- введение икт с помощью pascal и visual basic // Формирование рыночных отношений в Украине. 2020 №4(227). С. 97-101.
4. Симаков А.А. Java tracer. Программное средство для трассировки java программ // Заметки по информатике и математике. 2019. №3. С. 51-57.
  5. Иванов А.Н. Разработка приложения с использованием веб-технологий java // Ученые заметки ТОГУ. 2020. №4. С. 261-266.