Peuyeum: A Geospatial URL Encrypted Web Framework Using Advance Encryption Standard-Cipher Block Chaining Mode

R M Awangga*

Applied Bachelor Program of Informatics Engineering, Politeknik Pos Indonesia, Bandung 40151, Indonesia

*awangga@poltekpos.ac.id

Abstract. Many researches proposes geospatial web framework over the popularity of the Internet. Based on that, research on securing geospatial web framework is necessary. In this research aimed Peuyeum. Peuyeum is geospatial web framework with Encrypted Universal Resource Locator (URL). Advance Encryption Standard (AES)-Cipher Block Chaining (CBC) chosen as the method in this research. By calculating attack time, the brute force attack will reduce by this approach and resistance time will improve.

1. Introduction

Internet technology allows us to deliver data through a network. All computers around the world connect to the one big Network. The Internet network was built to facilitate the military in sending data, reports, and real-time activities directly to the headquarters command center. With the Internet network, then the actions and military policies can be quickly issued and enforced. Then the growing internet network is used in today's society. Where began emerging various Internet-based applications running as email and World Wide Web (WWW). Email is a new invention of the world of accuracy. Where a person no longer has to go to the post office, buy stamps, write and send letters and wait for his reply [1]. The discovery of this email that makes everyone can send a message easily and quickly until just by using a computer and internet network. Then an email will be directed to all corners of the world with the speed of delivery coming on the spot and can be relied on the spot. Similarly, newspapers are being replaced by the WWW [2]. Everyone can access and read all the news in the world simply by using the www address of a company or individual who owns the domain.

WWW then evolved from its origins using the HTML script language then evolved using the dynamic programming language [3]. Lack of HTML that can only deliver static content, which if you want to update then we have to open and re-coded the script. Developed again with dynamic programming such as PHP. Besides PHP also appears Java, Net Framework, and Python. This dynamic web programming language also requires a framework for the developer to add features quickly or add a particular function on the website. Website structure is now a lot of outstanding and growing in their way. In addition to WWW or better known as the site, also appears web service technology, which is the development of communication between machines, were using the internet intermediary as the communication [4]. This web service is opened and used with particular languages such as XML, and JSON that is grasped by other machines.

The use of web service and website technology for the benefit of geospatial data processing becomes necessary. Web Service is to reduce data processing time and reporting of geospatial data. The use of reporting websites and web services for the benefit of data collection is a quick and accurate recording solution rather than using manual methods that can take days to catalog data and release geospatial analytics. With the internet, network-based apps provide an easy to report in real time, cut the distance and processing time. With the web service and website then the time of cataloging geospatial data can be done in real-time. This efficiency is deemed necessary to continue to develop web-based frameworks, both for the creation of GIS interfaces and the use of web services regarding cataloging geospatial data in real-time [5].

Each web framework has advantages and disadvantages. Regarding this research, there is no particular web framework for geospatial, so it is important to research the development of web framework for geospatial. This context analysis begins with the security of framework [6]. Security on the Internet becomes an important thing to note, especially if these geospatial data contain state secrets security data. Encrypt URL is one of the areas of research conducted in this study. This research to complement and develop the concept of encryption on URL that is still rarely done.

2. Related Works

Research on the development of geospatial-based web services has been done using the Vegscape framework; the research serves The NASA Soil Moisture Active and Passive (SMAP) with Service Oriented Architecture (SOA) VegScpae [7]. Then the web geospatial framework was investigated with the development of the use of Fuzzy C Means. A Method to perform calculations in the spread of disease [8]. There is also a geospatial web service development using Ontology Framework. This research proposes a web service model using the ontology framework of the Open Geospatial Consortium (OGC) standard. They call it the OGC Web service framework [9]. The ontology research development then continues with the development of Sensor Modeling Language using the Semantic sensor Web Environment cite modeling [10]. Also, research on modeling spatial, temporal data on the semantic web [11].

Simulation of the framework was conducted to get calculation and diagnosis on the Geospatial web framework. The Web service put in a distributed network. Some parameters included in this calculation is Task arrival situation. Computing Resource Usage Simulation, Task Execution Status Simulation, Execution Result Simulation, Statistic and Evaluation of Execution Result [12]. Analysis using cloud framework is done to measure the efficiency of earth satellite imagery [13]. Another research that created web content aggregation from Digital Earth geospatial framework [14].Web service for Raster Processing was developed to process satellite image data quickly and high resolution and real-time [15].

Research on security did among others try to do brute force password on MSP430-BSL. The study reduces brute force time from 32 years to several days using sample password approach. With conditions of experiments with passwords are incorrect and correct there is no pause time. Reduce is also done by reducing 254-bit password to 40-bit password. Can reduce to 9.66E + 67 to 128 years only. Moreover, by setting the transfer rate baud rate to the IOT device can be reduced by only 32 years at the rate of 38400. Can be reduced again by creating a password dictionary or password example [16]. AES-CBC protection using Interrupt Vector (IV) Table, so is easy to predict from the method of randomization [17].

Implementation of AES-CBC requires a high-security application such as on satellite [18]. The study of security to build Honeypot with the calculation of the attacks is also done. The calculation consists of an inefficient ratio and a formula for comparing the login experiment with the password dictionary [19]. In the other side, Research on encryption on web service and website URL has not conducted. Research for securing URL is necessary to do.

3. Methods

3.1. Hyper text transport protocol secure

Hyper Text Transport Protocol Secure (HTTPS) is the development of security module of HyperText Transport Protocol (HTTP). HTTP is a communication protocol between the server and the client with web server software from the server side and the web browser from the user side. The URL of this framework is recommended using HTTPS to have the URL encapsulated.

3.2. Advance encryption standard

AES CBC is an encryption algorithm also called Rijndael algorithm. The US National Institute of Standards and Technology (AES) accepted this algorithm as a standard [18]. The encryption and decryption process performed by AES CBC is using two-way encryption, meaning there will be encryption method to encode data and there is decryption process to read data. Depicted in figure 1 and 2.

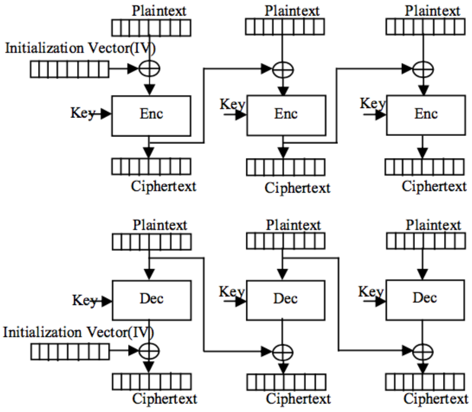


Figure 1. Cbc block diagram.

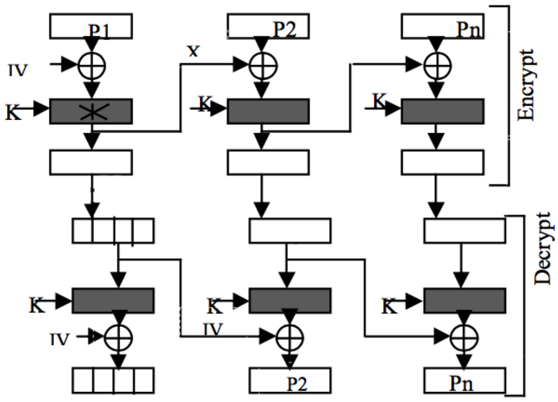


Figure 2. Cbc encryption description process.

4. System Design

In this research framework development that implemented through several steps in the form of URL Design, API Design, Logic Design, Build Library, Brute Force Calculation depicted in figure 3.

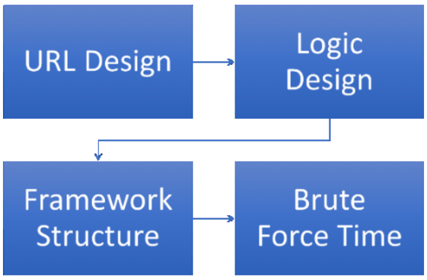


Figure 3. Method of this research.

4.1. Universal resource locator design

The Universal Resource Locator (URL) designed and proposed in this study is divided into two parts for two purposes. Design URL for website access address and map URL for access to Application User Interface (API) address. The URL Design proposed in this study is as follows:

4.1.1. https://domainname/urlencrypted. The encrypted URL design proposed in this study includes three development segments. Https as the medium that connects the server and client is a standard protocol for security (Hyper Text Transport Protocol). domain name is a domain name used by the Internet such as Top Level Domain (TLD) or Country Code Top Level Domain (ccTLD). The urlencrypted segment is the model proposed in this study. The part is subdivided to identify the use of the framework within the website and API. Inside the part is divided by using a separator. The Separator proposed in this research is a % sign. This splitter splits the identification and location of the classes and methods of the framework. The example of URL segmentation before encryption process as follows:

- Identifier%class%method%param
  The identifier is the identification of the web service function for API or web site. This identifier is a string of 3 characters long to identify the role of the URL. The class is the class of the called framework. The method serves to call methods that exist within the class. Param is the parameter used as input from the process.

4.2. Logic design

In building the framework, it takes logic to translate URLs entered by the user into a web browser. This logic serves to provide services by the URL entered. The logic design of the inserted URL steps is as follows.

- URL Decryption.
- Reading URL and split into a segment.
- If the identifier is website then go to site logic process.
- If the identifier is API then go to API logic block.
- Else, return error report.

API and website logic block consisting of:

- Read segment class as controller and make sure controller exist and do instantiation from class.
- Read segment method and param then calls the method with parameter param from class instantiation.
- Method of programmer Algorithm running
- The user gets HTML as a return.

4.3. Framework structure design

The framework should consist of a modular structure and easy to learn. The model is built using the Model, View, and Controller (MVC). MVC is a common structure used by most website frameworks. By using, the MVC structure is expected programmer can quickly learn and build an application on it. The Encrypt URL function is embedded in the library form in the Peuyeum Framework with the primary features that must exist, among others:

- URL Encrypt
- URL Decrypt

4.4. Brute force time

Calculates the approximate time of a URL's experiment by learning the URL structuring before being encrypted. Time is computed from the API and website URL cracking tests with possible timings.

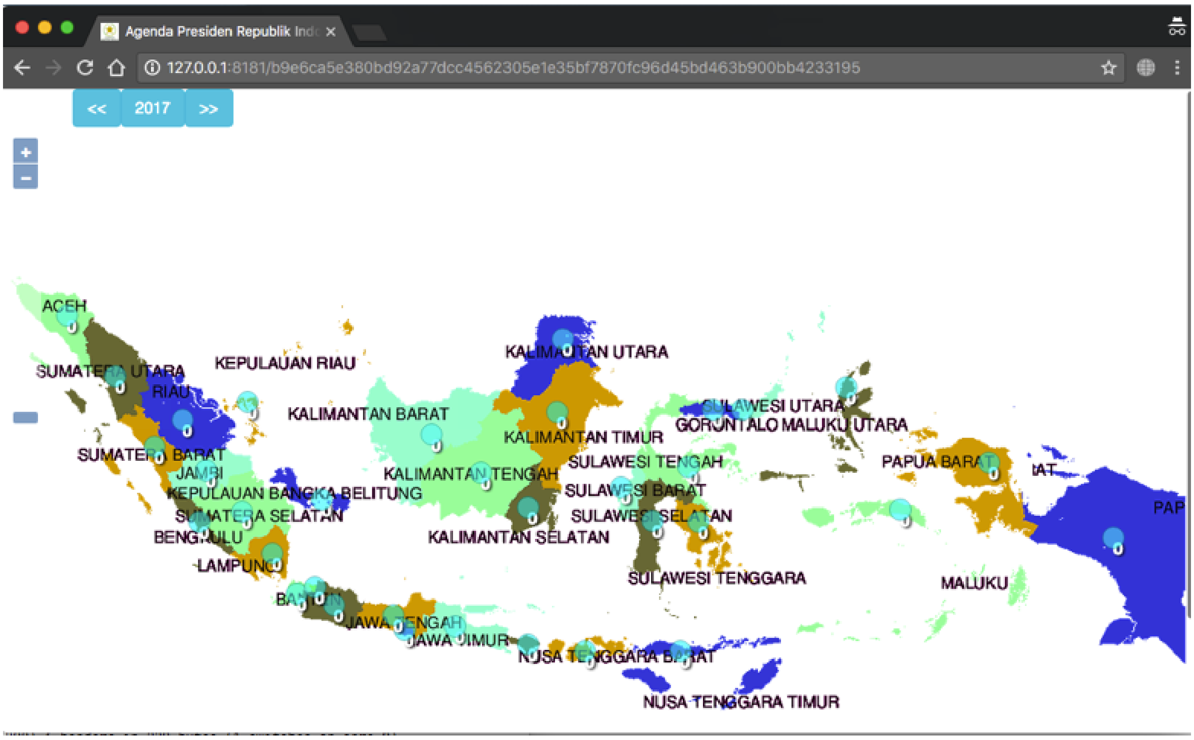
5. Results and Discussion

Experiment with building an Open layer-based app by displaying the map of Indonesia and with the base map of the Map Tile Service Web protocol. Table 1 explains the identification of website and API functions.

Table 1. URL information format.

Function	Class/Controller	Method	Parameter	Format	URL
WEB	peta	home	2017	WEB%peta%home%2017	*(1)
API	agenda	get List	5101	API%agenda%getList%5101	*(2)

(1) WEB: https://localhost/269bb14bfe28b304729da71b067b77cbf2dcbe7df75e58d9dfe8fc0c8984f152
(2) API: https://localhost/aabbace64a6acfebefaf6d7c248945a0d88ed89b061e66cab787e2340ca048a2

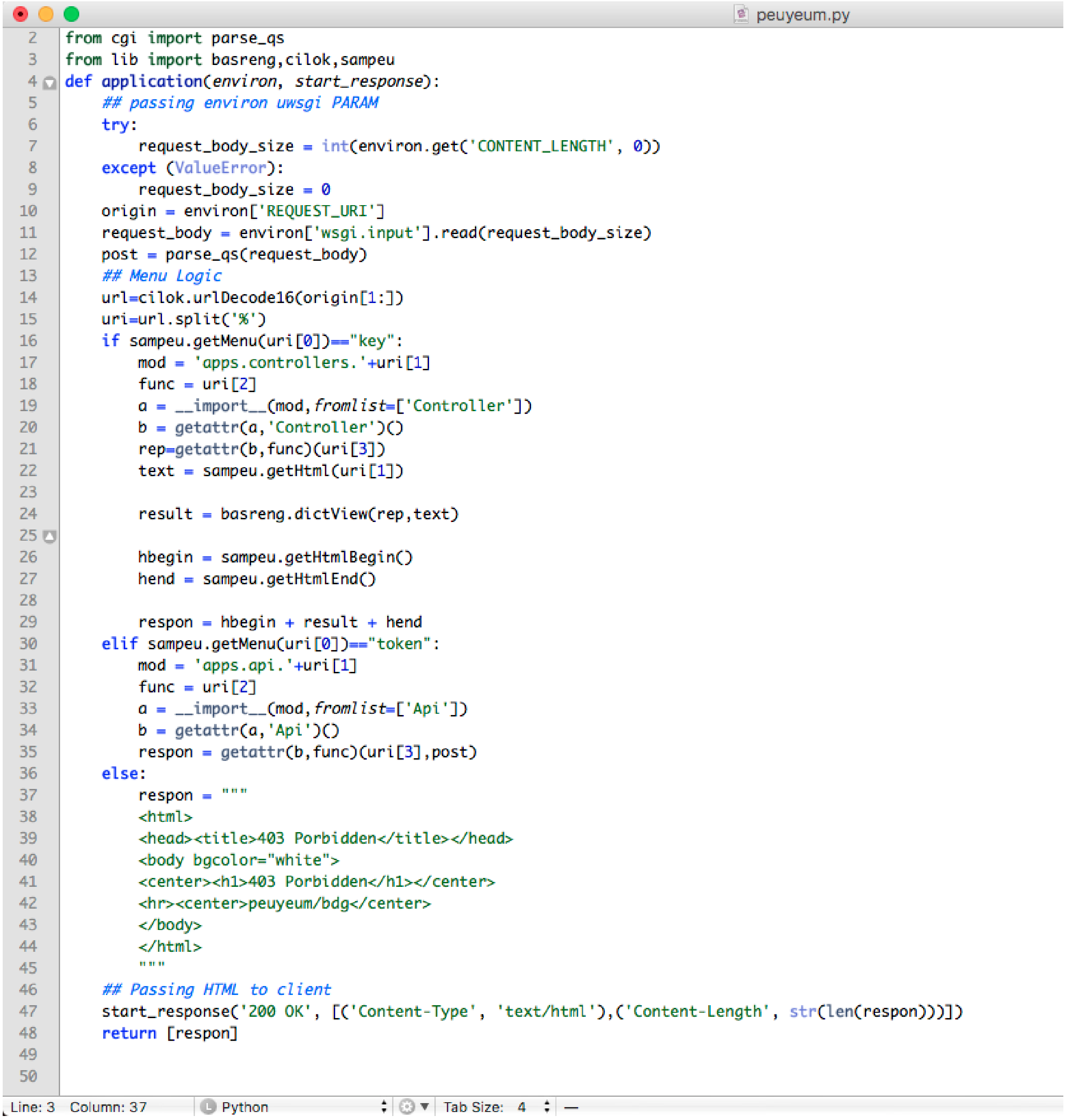


Agenda Presiden Republik Ind

127.0.0.1:8181/b9e6ca5e380bd92a77dcc4562305e1e35bf7870fc96d45bd463b900bb4233195

Figure 4. Indonesia map on Peuyeum framework.

Framework development using python uWSGI module. Framework logic explain as follow:

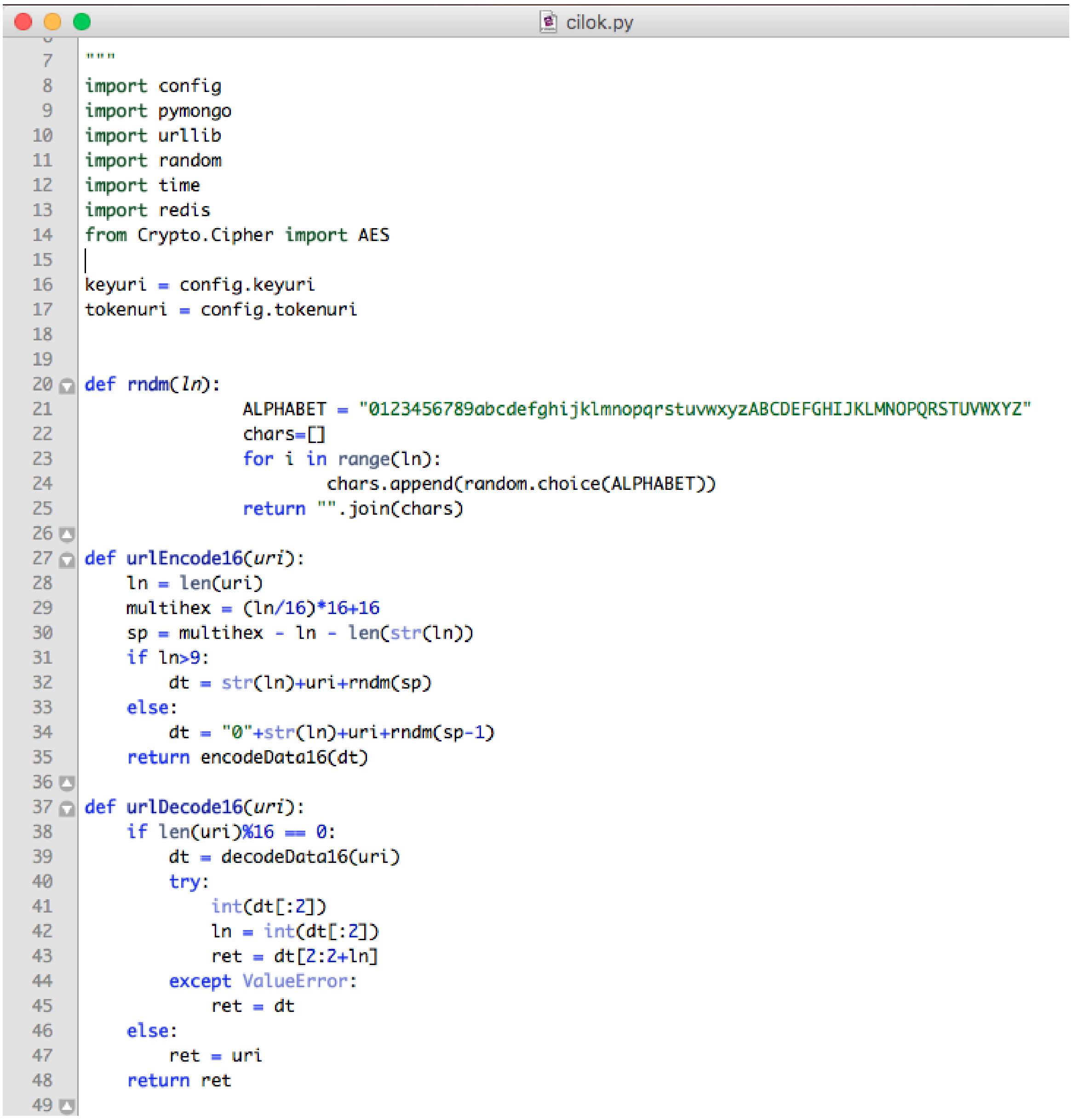


```python
from cgi import parse_qs
from lib import basreng,cilok,sampeu
def application(environ, start_response):
    ## passing environ uwsgi PARAM
    try:
        request_body_size = int(environ.get('CONTENT_LENGTH', 0))
    except (ValueError):
        request_body_size = 0
    origin = environ['REQUEST_URI']
    request_body = environ['wsgi.input'].read(request_body_size)
    post = parse_qs(request_body)
    ## Menu Logic
    url=cilok.urlDecode16(origin[1:])
    uri=url.split('%')
    if sampeu.getMenu(uri[0])=="key":
        mod = 'apps.controllers.'+uri[1]
        func = uri[2]
        a = __import__(mod,fromlist=['Controller'])
        b = getattr(a,'Controller')()
        rep=getattr(b,func)(uri[3])
        text = sampeu.getHtml(uri[1])

        result = basreng.dictView(rep,text)

        hbegin = sampeu.getHtmlBegin()
        hend = sampeu.getHtmlEnd()

        respon = hbegin + result + hend
    elif sampeu.getMenu(uri[0])=="token":
        mod = 'apps.api.'+uri[1]
        func = uri[2]
        a = __import__(mod,fromlist=['Api'])
        b = getattr(a,'Api')()
        respon = getattr(b,func)(uri[3],post)
    else:
        respon = """
        <html>
        <head><title>403 Porbidden</title></head>
        <body bgcolor="white">
        <center><h1>403 Porbidden</h1></center>
        <hr><center>peuyeum/bdg</center>
        </body>
        </html>
        """
    ## Passing HTML to client
    start_response('200 OK', [('Content-Type', 'text/html'),('Content-Length', str(len(respon)))])
    return [respon]
```

Line: 3 Column: 37 Python Tab Size: 4

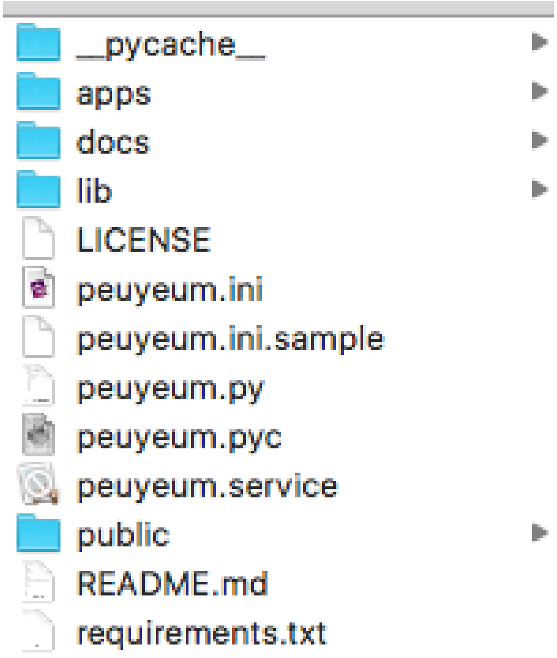
Figure 5. Source code of Peuyeum framework.

For the framework to process HTTP POST then use the parse_qs module to perform the POST data. Cilok is a library for URL encryption support function. This library placed in the Library folder along with other libraries required by the framework. The controller is the default class name for the website, while the default class name of the API is called API. So the structure of the directory of the Peuyeum Framework is illustrated in figure 7, and the MVC theory is in the structure in the app folder as shown in figure 8. Encryption and decryption algorithms using AES-CBC as follows.



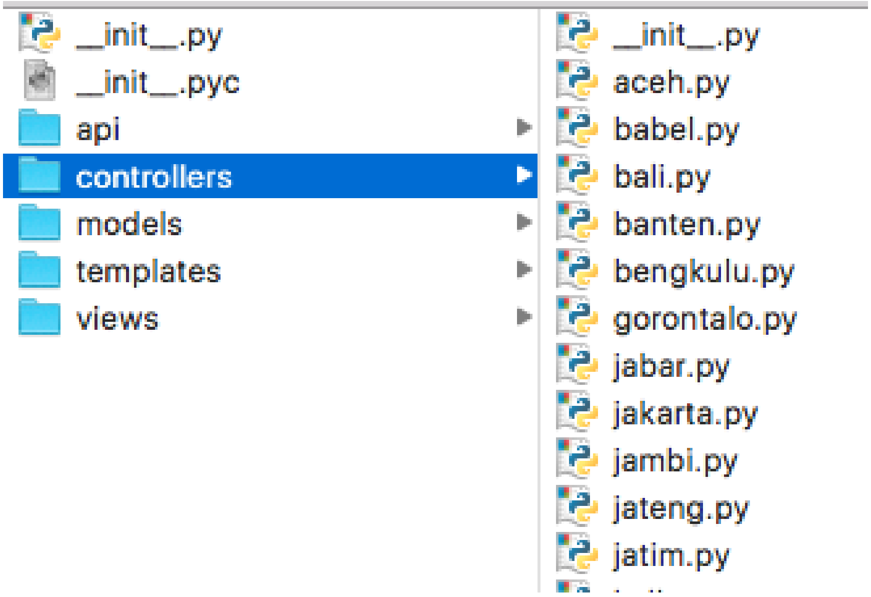
```python
"""
import config
import pymongo
import urllib
import random
import time
import redis
from Crypto.Cipher import AES

keyuri = config.keyuri
tokenuri = config.tokenuri


def rndm(ln):
               ALPHABET = "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
               chars=[]
               for i in range(ln):
                       chars.append(random.choice(ALPHABET))
               return "".join(chars)

def urlEncode16(uri):
    ln = len(uri)
    multihex = (ln/16)*16+16
    sp = multihex - ln - len(str(ln))
    if ln>9:
        dt = str(ln)+uri+rndm(sp)
    else:
        dt = "0"+str(ln)+uri+rndm(sp-1)
    return encodeData16(dt)

def urlDecode16(uri):
    if len(uri)%16 == 0:
        dt = decodeData16(uri)
        try:
            int(dt[:2])
            ln = int(dt[:2])
            ret = dt[2:2+ln]
        except ValueError:
            ret = dt
    else:
        ret = uri
    return ret

```

Figure 6. Source code of encryption and decryption process.



```
__pycache__
apps
docs
lib
LICENSE
peuyeum.ini
peuyeum.ini.sample
peuyeum.py
peuyeum.pyc
peuyeum.service
public
README.md
requirements.txt
```

Figure 7. Main directory structure of Peuyeum.



```
__init__.py        __init__.py
__init__.pyc       aceh.py
api                babel.py
controllers        bali.py
models             banten.py
templates          bengkulu.py
views              gorontalo.py
                   jabar.py
                   jakarta.py
                   jambi.py
                   jateng.py
                   jatim.py
```

Figure 8. Inside apps folder of Peuyeum Framework.

The number of characters in the CBC (Encrypted String) should be 16 characters and multiples, so in the algorithm, if the character URL is less than a multiple of 16 then the random Alphabet is added. The data encryption combined head that contains the number of characters used as data from the multiples of 16 characters sent. This action performs to decrypt encrypted string.

Therefore, for brute force attack calculation, here we use 16 characters for AES-Key and 16 characters for IV on AES CBC usage. With 26 uppercase letters, 26 lowercase letters and 10 characters plus characters with 33 characters. Then we get an experiment to get access to URL by brute force is $9516 * 9516$ times trial.

6. Conclusions

Peuyeum Framework can develop in other feature of programming. For the security, a process should be considered to do IP blocking which often tries to find a security hole. Then from the framework development side can be added development features for computing efficiency. In the future research, web service composition can be interesting topics.

References

[1] Al Mazrouei S A S, Dahalan N and Faiz M H 2015 *Researchers World* **6** 32.

[2] Seufert M, Egger S, Slanina M, Zinner T, Hobfeld T and Tran-Gia P *IEEE Communications Surveys & Tutorials* **17** pp. 469–492.

[3] Akhawe D M 2014.

[4] Sheng Q Z, Qiao X, Vasilakos A V, Szabo C, Bourne S and Xu X 2014 *Information Sciences* **280** pp. 218–238.

[5] Alarabi L, Eldawy A, Alghamdi R and Mokbel MF 2014 *Proceedings of the 2014 ACM SIGMOD international conference on Management of data* (ACM) pp. 897–900.

[6] Peretti G, Lakkundi V and Zorzi M 2015 *Communication Systems and Networks (COMSNETS), 2015 7th International Conference on (IEEE)* pp. 1–6.

[7] Yang Z, Hu L, Yu G, Shrestha R, Di L, Boryan C and Mueller R 2016 *Geoscience and Remote Sensing Symposium (IGARSS), 2016 IEEE International (IEEE)* pp. 3624–3627.

[8] Di Martino F, Mele R, Sessa S, Barillari U E and Barillari M R 2014 *Intelligent Networking and Collaborative Systems (INCoS), 2014 International Conference on (IEEE)* pp. 112–118.

[9] Shanming W and Jianjing S 2008 *Information Science and Engineering, 2008. ISISE'08. International Symposium on* (IEEE) **2** pp. 107–110.

[10] Meng X, Wang Y and Wu Y 2014 *Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on (IEEE)* pp. 1090–1095.

[11] Andrejev A, Misev D, Baumann P and Risch T 2015 *Data Science and Data Intensive Systems (DSDIS), 2015 IEEE International Conference on (IEEE)* pp. 38–45.

[12] Xiang B, Li X, Zhang M, Lu L, Li F, Zhao B and Gui Z 2015 *Geoinformatics, 2015 23rd International Conference on (IEEE)* pp. 1–5.

[13] Patterson M T, Anderson N, Bennett C, Bruggemann J, Grossman R L, Handy M, Ly V, Mandl D J, Pederson S, Pivarski J et al. 2016 *Big Data Computing Service and Applications (BigDataService), 2016 IEEE Second International Conference on (IEEE)* pp. 156–165.

[14] Xu Y, Weng J, Sharma A R and Yussupov D 2011 *Geoinformatics, 2011 19th International Conference on (IEEE)* pp. 1–5.

[15] Karantzalos K, Bliziotis D and Karmas A 2015 *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **8** pp. 4665–4674.

[16] Tellez M, El-Tawab S and Heydari H M 2016 *Systems and Information Engineering Design Symposium (SIEDS), 2016 IEEE (IEEE)* pp. 72–77.

[17] Tellez M, El-Tawab S and Heydari M H 2016 *Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on (IEEE)* pp. 182–187.

[18] Vaidehi M and Rabi B J 2014 *Current Trends in Engineering and Technology (ICCTET), 2014 2nd International Conference on (IEEE)* pp. 499–502.

[19] Fraunholz D, Zimmermann M, Anton S D, Schneider J and Schotten H D 2017 *Cloud Computing, Data Science & Engineering-Confluence, 2017 7th International Conference on (IEEE)* pp. 416–421.