HW 3 - CNS Sapienza

Daniele Oriana 1457625

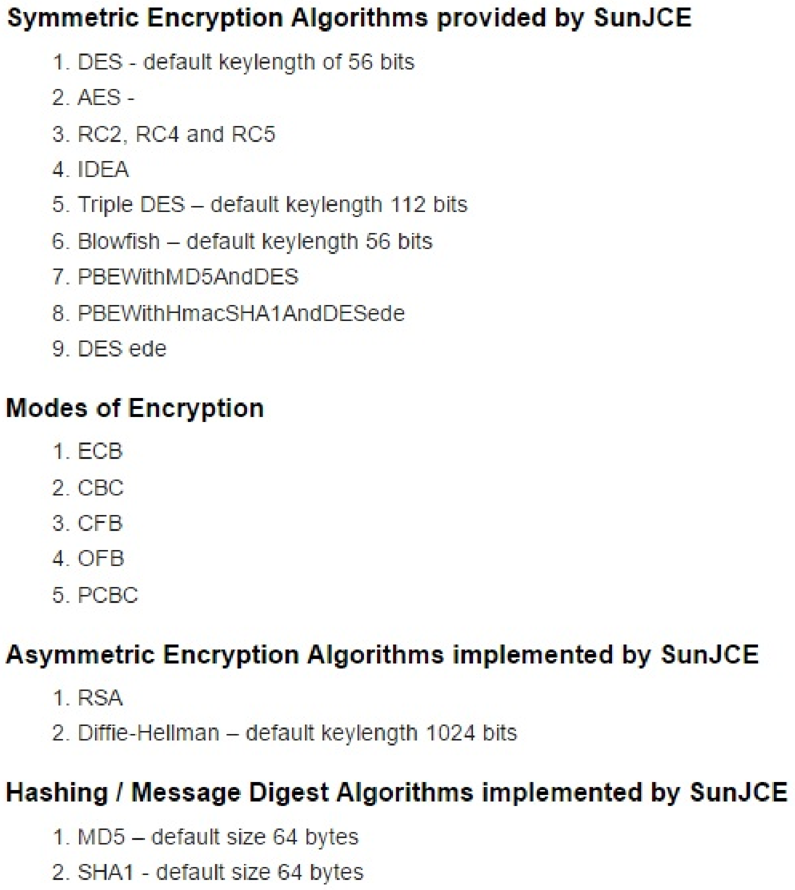
10 Nov 2016

With reference to two languages (to be arbitrarily chosen among C, C++, Java, Javascript, Phyton, Perl, PHP, Ruby, bash, Lisp etc. - either compiled or interpreted languages):

- **locate libraries publicly available that can offer implementations of hash functions**
- **Given two different contracts, write a program in which you apply some insignificant changes in order to obtain the same hash for the two contracts**

Answer (First Library)

The Java Cryptography Extension (JCE: http://www.oracle.com/technetwork /java/javase/downloads/jce8-download-2133166.html) is an officially released Standard Extension to the Java Platform and part of Java Cryptography Architecture. JCE was an optional package to JDK v 1.2.x and 1.3.x. JCE has been integrated into JDK v1.4. It offers the following services:



Symmetric Encryption Algorithms provided by SunJCE

1. DES - default keylength of 56 bits
2. AES -
3. RC2, RC4 and RC5
4. IDEA
5. Triple DES – default keylength 112 bits
6. Blowfish – default keylength 56 bits
7. PBEWithMD5AndDES
8. PBEWithHmacSHA1AndDESede
9. DES ede

Modes of Encryption

1. ECB
2. CBC
3. CFB
4. OFB
5. PCBC

Asymmetric Encryption Algorithms implemented by SunJCE

1. RSA
2. Diffie-Hellman – default keylength 1024 bits

Hashing / Message Digest Algorithms implemented by SunJCE

1. MD5 – default size 64 bytes
2. SHA1 - default size 64 bytes

Figure 1: Offered Services.

In order to use these services you have to download the "Jurisdiction Policy Files" from the Oracle web site, here you can find one of the latest version of these files http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html. You can download the files only after you have accepted the *Java SE BCL License Agreement* which defines the conditions of use. Once you have downloaded the files, you have to add them in the correct folder:

- <java-home>/lib/security for UNIX systems
- <java-home>\lib\security for WINDOWS systems

All the documentation that allows to interact with the interfaces offered by these services is available at this URL: http://docs.oracle.com/javase/8 /docs/technotes/guides/security/crypto/CryptoSpec.html, where you can find the *Java Cryptography Architecture (JCA) Reference Guide*. On the Oracle web site you can find also all the news about supporting and communities in the section “Support” and in the subsection “Community”. (You can find an example of usage of SHA1 hash function in the implementation of the birthday attack in the second part of the homework).

Answer (Second Library)

Python Cryptography Toolkit (pycrypto: https://pypi.python.org/pypi/pycrypto) is a collection secure hash functions and various encryption algorithm available for different versions of Python: from Python version 2.1 through 3.3, Python 1.5.2 is not supported. Is possible to find all the documentation at the following URL: http://pythonhosted.org/pycrypto/. Is possible to collaborate and to report bugs: in order to take part to the collaboration a registration is required, after the login you can propose new versions of the current libraries that will be analyzed by the pycrypto team (only new versions, because the pycrypto team has decided to mantain the actual version also for the future, fixing only possible bugs), while you can report bugs using this URL https://launchpad.net/pycrypto/+bugs.
The installation can be done in two ways:

1 Download the .tar.gz file from the pycrypto web site. Extract the files on your pc. Run from your prompt “python setup.py buil” to build the package and after that run python “setup.py instal” to install it.

2 Write on your prompt the command “pip install pycrypto” (it will work only with the latest versions of Python).

With regard to hash function, here is a small example of usage of SHA256 hash function:

```
    >>> from Crypto.Hash import SHA256
>>>
>>> h = SHA256.new()
>>> h.update(b’Hello’)
>>> print h.hexdigest()
```

Note that pycrypto is completely open source.

Birthday attack on a weak SHA1

In order to can use a birthday attack you have to face an hash function that produces an output with length less than 160bits. So, in my code, i have used a SHA1 function reduced to 24 bits. I have tested until 32 bits, and in that case the program requires more or less 5 minutes to find two contract with a collision on the hash function. The final code presents the SHA1 function with 24 bits in order to test the code quickly, but you can change the length of the hash string in one of the last rows of code: *returning = returning.substring(0, 6);*, simply changing the value "6" with the length that you want (please note that the two contracts have some different salaries, but the code allows to find the same hash function anyway).
How to run the program:

- Put in the same folder of the .java file two files: ContractGood1457625.txt (in which you have one version of the contract) and ContractBad1457625.txt (in which you have the other version of the contract).
- Run the java program and follow the istruction on the terminal.
- See the result in Official1457625.txt and Unofficial1457625.txt.

Note that in order to run the program you have to install the JCE (follow the instructions in the relative answer of the homework).