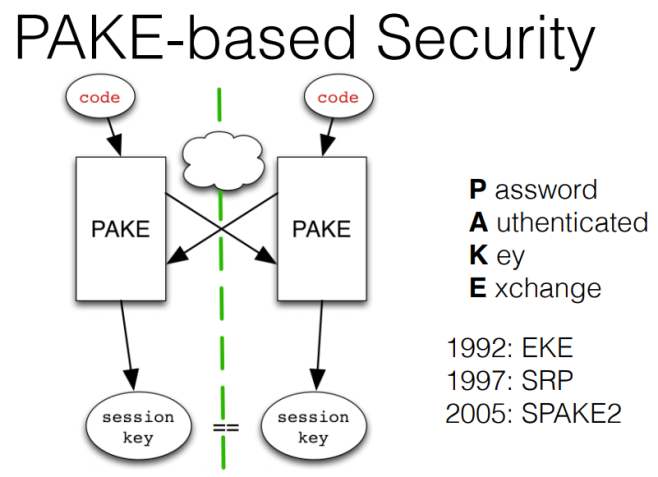
|  |
| --- |
| **과제 보고서**  보고서 및 논문 윤리 서약  1. 나는 보고서 및 논문의 내용을 조작하지 않겠습니다.  2. 나는 다른 사람의 보고서 및 논문의 내용을 내 것처럼 무단으로 복사하지 않겠습니다.  3. 나는 다른 사람의 보고서 및 논문의 내용을 참고하거나 인용할 시 참고 및 인용 형식을 갖추고 출처를 반드시 밝히겠습니다.  4. 나는 보고서 및 논문을 대신하여 작성하도록 청탁하지도 청탁받지도 않겠습니다.  나는 보고서 및 논문 작성 시 위법 행위를 하지 않고, 명지인으로서 또한 공학인으로서 나의 양심과 명예를 지킬 것을 약속합니다.  EMB000034e07094  보고서명 : 공개SW 기능분석 보고서  학 과 : 컴퓨터공학과  과 목 : 공개SW실무  담당교수 : 류연승 교수님  조 : 3조  이 름 : 60165110 송현석  이 름 : 60172186 임준형  이 름 : 60142324 이진영  이 름 : 60172129 김다연  이 름 : 60172210 최선학 |

목차

1. 프로젝트 개요
   1. Magic-Wormhole 소개
   2. Pake
   3. 소스파일 디렉토리 구조
2. wormhole.py
   1. DelegatedWormhole(object) 클래스
   2. DeferredWormHole(object) 클래스
   3. import된 모듈
3. send.py
   1. Send(object) 클래스
   2. import된 모듈
4. input.py
   1. Input(object) 클래스
   2. Helper(object) 클래스
   3. import된 모듈
5. key.py
   1. key(object) 클래스
   2. SortedKey(object) 클래스
   3. function
   4. import된 모듈
6. receive.py
   1. Receive(object) 클래스
   2. import된 모듈
7. 프로젝트 개요

1.1 Magic-Wormhole 소개

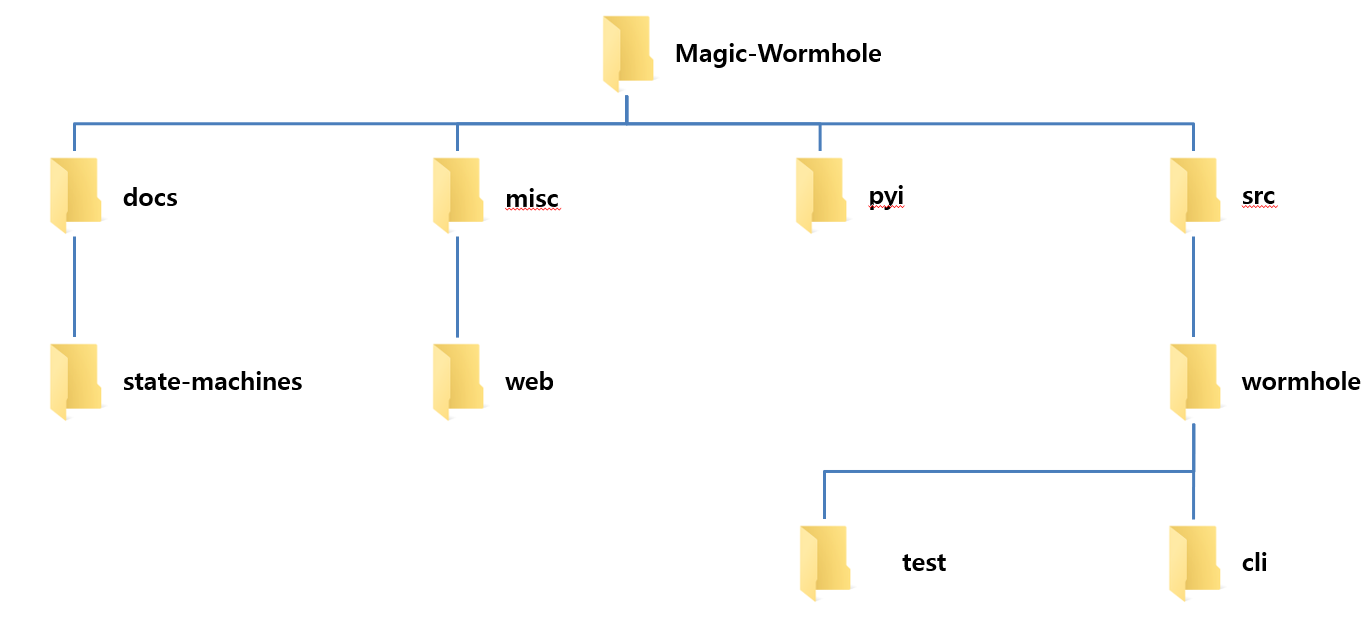
이 패키지는 wormhole이라는 이름의 라이브러리와 명령 줄 도구를 제공하므로 한 컴퓨터에서 다른 컴퓨터로 임의 크기의 파일과 디렉터리(또는 짧은 텍스트)를 가져올 수 있습니다. 두 끝점은 동일한 “wormhole codes”를 사용하여 식별합니다. 파일을 전송하는 컴퓨터에서는 코드를 생성하고 보여줍니다. 이 코드는 파일을 받을 컴퓨터에 입력해야 합니다.



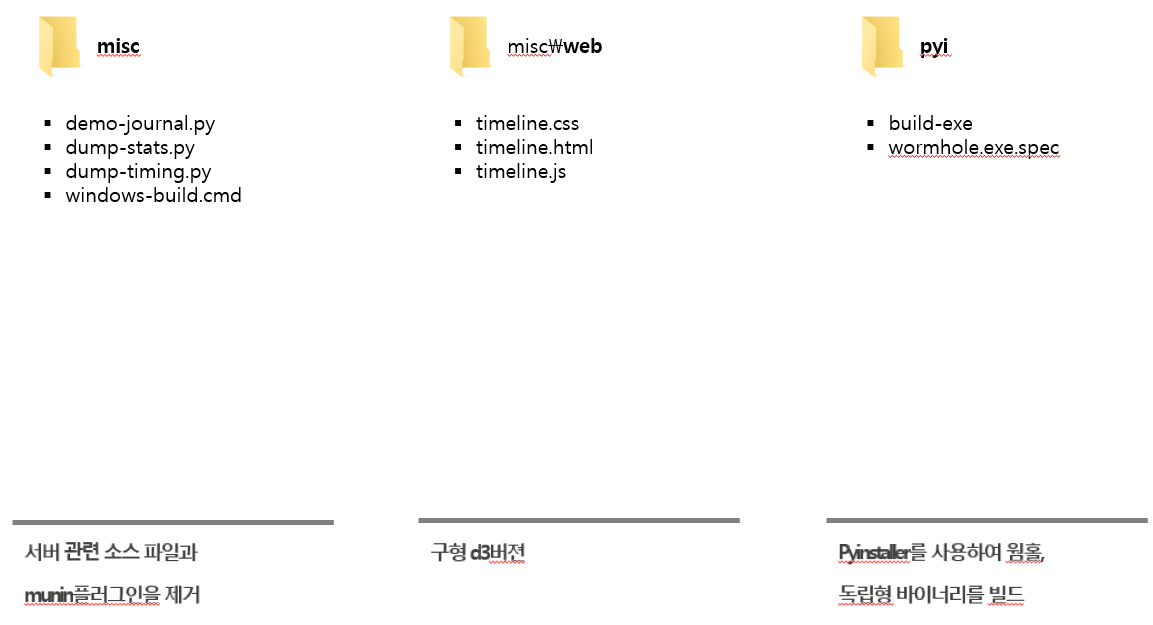
1.2 Pake

Pake는 두 명 이상의 참여자가 패스워드 기반으로 암호화된 채널을 만들어서 서로 통신할 수 있게 해주는 암호학적인 방법을 말한다.

1.3 소스 파일 디렉토리 구조









**2. wormhole.py**

* 1. **DelegatedWormhole(object) 클래스**

|  |  |
| --- | --- |
| 하는 일 | 웜홀에 정보가 있는 경우에 웜홀 제어 및 데이터 전송을 합니다. |
| 보유  함수 | * def \_\_attrs\_post\_init\_\_(self) * def \_set\_boss(self, boss) * def allocate\_code(self, code\_length=2) * def input\_code(self) * def set\_code(self, code) * def send\_message(self, plaintext) * def derive\_key(self, purpose, length) * def close(self) * def debug\_set\_trace(self, client\_name, which="B N M S O K SK R RC L C T", file=sys.stderr) * def got\_welcome(self, welcome) * def got\_code(self, code) * def got\_key(self, key) * def got\_verifier(self, verifier) * def got\_versions(self, versions) * def received(self, plaintext) * def closed(self, result) |

* 1. **DeferredWormHole(object) 클래스**

|  |  |
| --- | --- |
| 하는 일 | 권한이 넘어간 것이 아닌 경우 웜홀 제어 및 데이터 전송 |
| 보유  함수 | * def \_\_init\_\_(self, eq) * def \_set\_boss(self, boss) * def get\_code(self) * def get\_welcome(self) * def get\_unverified\_key(self) * def get\_verifier(self) * def get\_versions(self) * def get\_message(self) * def allocate\_code(self, code\_length=2) * def input\_code(self) * def set\_code(self, code) * def send\_message(self, plaintext) * def derive\_key(self, purpose, length) * def close(self) * def debug\_set\_trace(self, client\_name, which="B N M S O K SK R RC L A I C T", file=sys.stderr) * def got\_welcome(self, welcome) * def got\_code(self, code) * def got\_key(self, key) * def got\_verifier(self, verifier) * def got\_versions(self, versions) * def received(self, plaintext) * def closed(self, result) |

**2.3** **import된 모듈**

|  |  |
| --- | --- |
| import된  모듈 | * from \_\_future\_\_ import absolute\_import, print\_function, unicode\_literals * import os * import sys * from attr import attrib, attrs * from twisted.python import failure * from zope.interface import implementer * from .\_boss import Boss * from .\_interfaces import IDeferredWormhole, IWormhole * from .\_key import derive\_key * from .errors import NoKeyError, WormholeClosed * from .eventual import EventualQueue * from .journal import ImmediateJournal * from .observer import OneShotObserver, SequenceObserver * from .timing import DebugTiming * from .util import bytes\_to\_hexstr, to\_bytes * from .\_version import get\_versions |

**3. send.py**

* 1. **Send(object) 클래스**

|  |  |
| --- | --- |
| 하는 일 | Send는 queue 형태로 파일을 전달해줍니다 |
| 보유  함수 | * def \_\_attrs\_post\_init\_\_(self) * def wire(self, mailbox) * def S0\_no\_key(self) * def S1\_verified\_key(self) * def got\_verified\_key(self, key) * def send(self, phase, plaintext) * def queue(self, phase, plaintext) * def record\_key(self, key) * def drain(self, key) * def deliver(self, phase, plaintext) * def \_encrypt\_and\_send(self, phase, plaintext) |

* 1. **import된 모듈**

|  |  |
| --- | --- |
| import된  모듈 | * from \_\_future\_\_ import absolute\_import, print\_function, unicode\_literals * from attr import attrib, attrs * from attr.validators import instance\_of, provides * from automat import MethodicalMachine * from zope.interface import implementer * from . import \_interfaces * from .\_key import derive\_phase\_key, encrypt\_data |

**4. input.py**

* 1. **Input(object) 클래스**

|  |  |
| --- | --- |
| 하는 일 | 리스트에 있는 웜홀 코드 중 생성된 코드를 입력받는다. |
| 보유  함수 | * def \_\_attrs\_post\_init\_\_(self) * def set\_debug(self, f) * def \_debug(self, what) * def wire(self, code, lister) * def when\_wordlist\_is\_available(self) * def S0\_idle(self) * def S1\_typing\_nameplate(self) * def S2\_typing\_code\_no\_wordlist(self) * def S3\_typing\_code\_yes\_wordlist(self) * def S4\_done(self) * def start(self) * def got\_nameplates(self, all\_nameplates) * def got\_wordlist(self, wordlist) * def refresh\_nameplates(self) * def get\_nameplate\_completions(self, prefix) * def choose\_nameplate(self, nameplate) * def \_choose\_nameplate(self, nameplate) * def get\_word\_completions(self, prefix) * def choose\_words(self, words) * def do\_start(self) * def do\_refresh(self) * def record\_nameplates(self, all\_nameplates) * def \_get\_nameplate\_completions(self, prefix) * def record\_all\_nameplates(self, nameplate) * def record\_wordlist(self, wordlist) * def notify\_wordlist\_waiters(self, wordlist) * def no\_word\_completions(self, prefix) * def \_get\_word\_completions(self, prefix) * def raise\_must\_choose\_nameplate1(self, prefix) * def raise\_must\_choose\_nameplate2(self, words) * def raise\_already\_chose\_nameplate1(self) * def raise\_already\_chose\_nameplate2(self, prefix) * def raise\_already\_chose\_nameplate3(self, nameplate) * def raise\_already\_chose\_words1(self, prefix) * def raise\_already\_chose\_words2(self, words) * def do\_words(self, words) |

* 1. **Helper(object) 클래스**

|  |  |
| --- | --- |
| 하는 일 | input 클래스 외부에서 필요한 내용을 다룬다. |
| 보유  함수 | * def \_\_attrs\_post\_init\_\_(self) * def refresh\_nameplates(self) * def get\_nameplate\_completions(self, prefix) * def choose\_nameplate(self, nameplate) * def when\_wordlist\_is\_available(self) * def get\_word\_completions(self, prefix) * def choose\_words(self, words) |

* 1. **import된 모듈**

|  |  |
| --- | --- |
| import된  모듈 | * from \_\_future\_\_ import absolute\_import, print\_function, unicode\_literals import threading * from attr import attrib, attrs * from attr.validators import provides * from automat import MethodicalMachine * from twisted.internet import defer * from zope.interface import implementer * from . import \_interfaces, errors * from .\_nameplate import validate\_nameplate |

**5. key.py**

* 1. **key(object) 클래스**

|  |  |
| --- | --- |
| 하는 일 | got\_code와 got\_pake로 code와 pake를 받아와 SortedKey로 전송합니다 |
| 보유  함수 | * def \_\_attrs\_post\_init(self) * def wire(self, boss, mailbox, receive) * def S00(self) * def S01(self) * def S11(self) * def got\_code(self, code) * def got\_pake(self, body) * def stash\_pake(self, body) * def deliver\_code(self, code) * def deliver\_pake(self, body) * def deliver\_code\_and\_stashed\_pake(self, code) |

* 1. **SortedKey(object) 클래스**

|  |  |
| --- | --- |
| 하는 일 | SortedKey는 받아온 키 값들을 올바른 순서대로 정렬하고, 암호화시킵니다 |
| 보유  함수 | * def wire(self, boss, mailbox, receive) * def S0\_know\_nothing(self) * def S1\_know\_code(self) * def S2\_know\_key(self) * def S3\_scared(self) * def got\_code(self, code) * def got\_pake(self, body) * def got\_pake\_good(self, msg2) * def got\_pake\_bad(self) * def build\_pake(self, code) * def scared(self) * def compute\_key(self, msg2) |

* 1. **function**

|  |
| --- |
| * def HKDF(skm, outlen, salt=None, CTXinfo = b””) |
| * 헤시 기반 메시지 인증 코드(HMAC)를 기반으로 하는 간단한 키 유도 함수 |
| * def derive\_key(key, purpose, length=SecretBox.KEY\_SIZE) |
| * 받아온 키가 정상적으로 받아온 것인지 확인하는 함수 |
| * def derive\_phase\_key(key, side, phase) |
| * isInstance로 받아온 것이 항상 참이라 가정하고, side와 phase를 ascii로 인코딩한 후에 derive\_key에 전송해준다. |
| * def decrypt\_data(key, encrypted) |
| * 받아온 데이터를 해독하는 함수 |
| * def encrypt\_data(key, plaintext) |
| * 데이터를 받아와 암호화 시키는 함수 |

* 1. **import된 모듈**

|  |  |
| --- | --- |
| import된  모듈 | * from \_\_future\_\_ import absolute\_import, print\_function, unicode\_literals * from hashlib import sha256 * import six * from attr import attrib, attrs * from attr.validators import instance\_of, provides * from automat import MethodicalMachine * from hkdf import Hkdf * from nacl iport utils * from nacl.exceptions import CryptoError * from nacl.secret import SecretBox * from spake2 import SPAKE2\_Symmetric * from zope.interface import implementer * from . import \_interfaces * from .util import (bytes\_to\_dict, bytes\_to\_hexstr, dict\_to\_bytes, hexstr\_to\_btes, to\_bytes) |

**6. receive.py**

* 1. **Receive(object) 클래스**

|  |  |
| --- | --- |
| 하는 일 | receive는 인증된 키가 맞는지 아닌지 확인하고 맞으면 전달된 파일을 받아옵니  다 |
| 보유  함수 | * def \_\_attrs\_post\_init\_\_(self) * def wire(self, boss, send) * def S0\_unknown\_key(self) * def S1\_unverified\_key(self) * def S2\_verified\_key(self) * def S3\_scared(self) * def got\_message(self, side, phase, body) * def got\_message\_good(self, phase, plaintext) * def got\_message\_bad(self) * def got\_key(self, key) * def record\_key(self, key) * def S\_got\_verified\_key(self, phase, plaintext) * def W\_happy(self, phase, plaintext) * def W\_got\_verifier(self, phase, plaintext) * def W\_got\_message(self, phase, plaintext) * def W\_scared(self) |

* 1. **import된 모듈**

|  |  |
| --- | --- |
| import된  모듈 | * from \_\_future\_\_ import absolute\_import, print\_function, unicode\_literals * from attr import attrib, attrs * from attr.validators import instance\_of, provides * from automat import MethodicalMachine * from zope.interface import implementer * from . import \_interfaces * from .\_key import CryptoError, decrypt\_data, derive\_key, derive\_phase\_key |