파일 송수신을 위한 안정적인 시스템 구성

[1] 요구 사항

1. 서버간 파일 PUSH/PULL 기능

-

2. 파일 정합성

- Buff

3. 파일 암복호화 처리 구현

- AES 알고리즘을 이용한 데이터 바이트어레이 암호화하여 전송

4. 콘솔을 통한 효율적인 관리 기능 구현

[2] 시스템 구성

[3] 주요 개발내용 및 소스코드

<개발환경>

- Server, Client for Managing : Eclipse(JAVA)

<구현된 개발내용>

1. 파일 자체 암복호화 처리

-자바 제공 암호화 라이브러리 이용

(javax.crypto.Cipher 클래스 – AES 암호화 알고리즘 사용)

2. 서버 프로그램의 통신 구현 방식

- NIO Selector를 통한 넌블로킹 방식으로 클라이언트와 원격서버로 소켓 통신을 수행한다.

Selector select 부분

데이터 송신 부분

3. 서버와 클라이언트와 환경설정 정보 저장

클라이언트과 관리할 서버의 IP주소 및 Port 주소를 환경설정파일(clientconfig.properties)에 저장하고 있고 서버 또한 원격으로 파일을 전송할 서버의 IP주소 및 Port 주소를 저장하고 있다. 환경 설정 파일이 존재하지 않을 경우 프로그램은 자동으로 종료된다.

실제로는 암호화 처리되어 있으며, 예시로 비암호화 처리된 포맷을 첨부하였다.

4. 서버 프로그램 코드에서부터 시작 및 종료 부분

- 종료를 위해 뭘 했는지, 환경설정을 읽는 부분 및 Thread 생성부터 실행하게 된다.

5. 암/복호화 구현 방법 – AES 알고리즘

- 서버간 전송될 파일은 AES 암호화 알고리즘을 사용하여 암호화/복호화를 수행한다. (javax.crypto.Cipher)

암호화 초기 과정에서 KEY 및 IV(Initial Vector 첫 block 암호화에 사용됨) 값을 SHA-1으로 Hash

처리하게 되며, 암호화에 사용된다. 최종의 암호화된 결과물은 Base64 String으로 출력된다.

복호화는 역으로 Base64 String을 먼저 디코딩(decoding)하게 되고 AES 복호화를 수행하게 된다.

AES 암호화 알고리즘은 기존의 DES 암호화 알고리즘을 대체하여 미국 표준기술연구소(NIST)의 미

국 연방 표준 암호화 알고리즘으로 선정되었으며 안정성이 검증되었다.

[4] 미구현 기능에 대한 적용방안