

FIG. 1

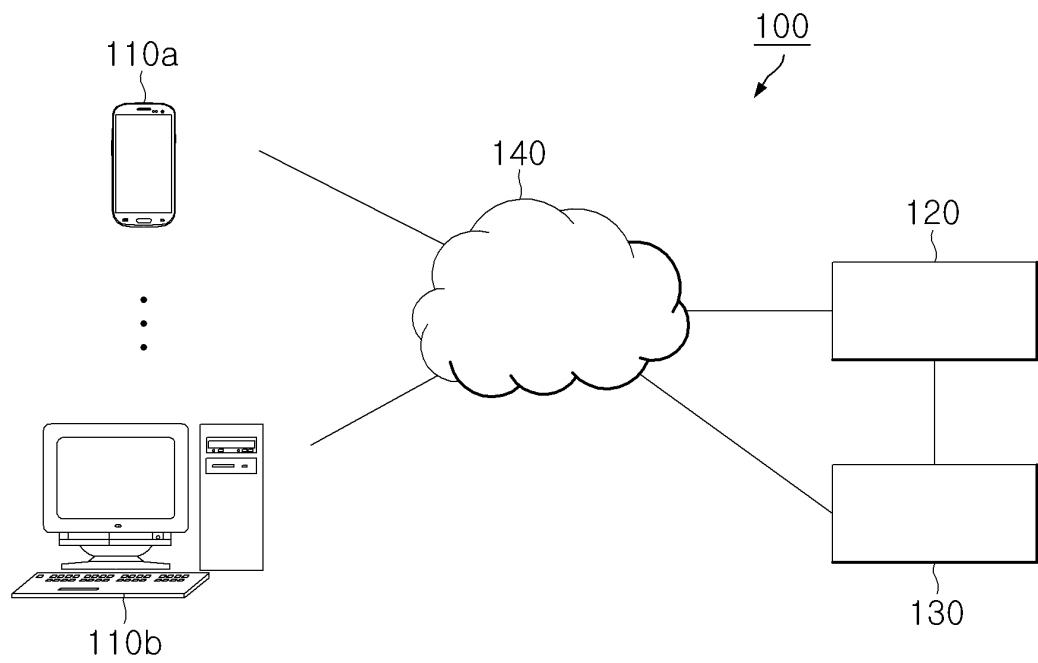


FIG. 2

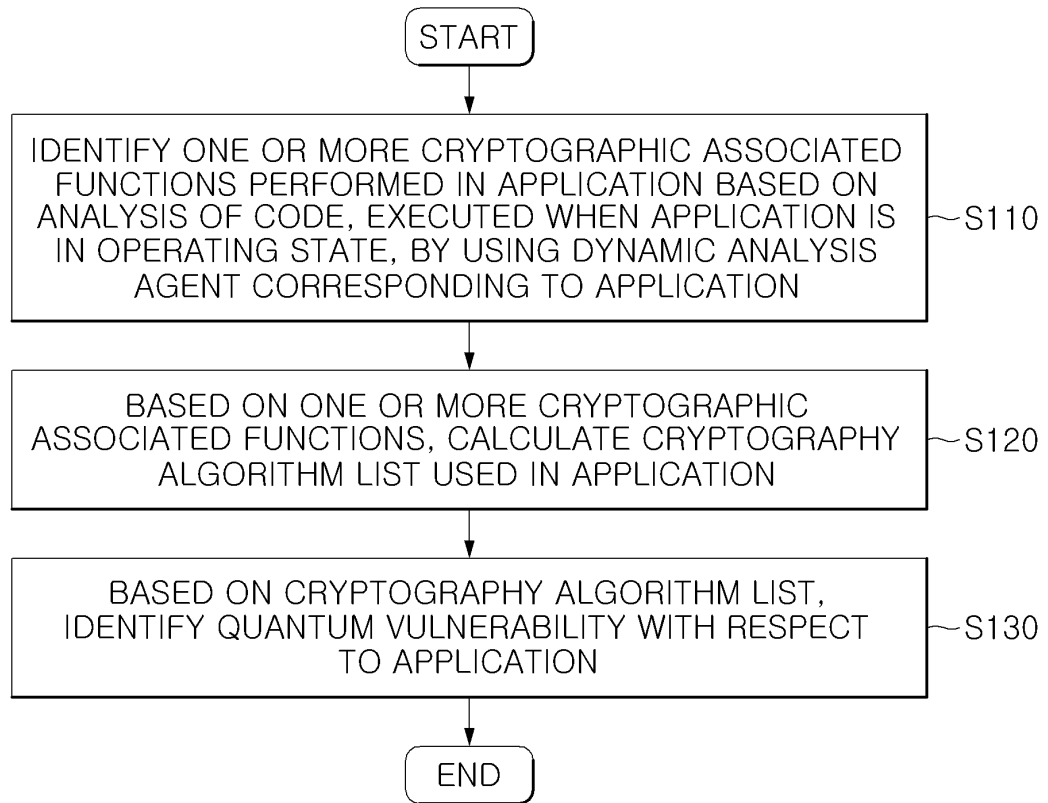


FIG. 3

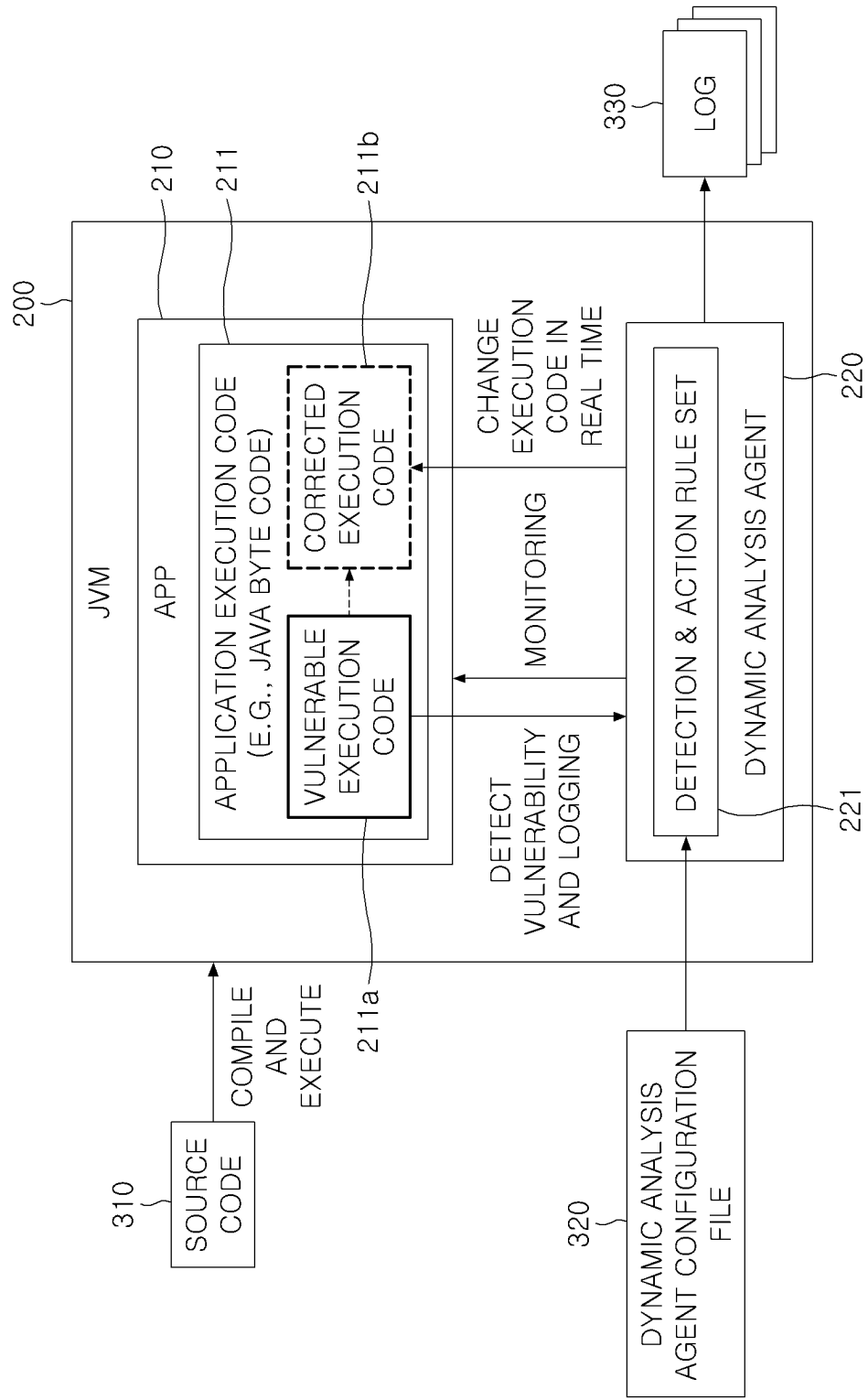


FIG. 4

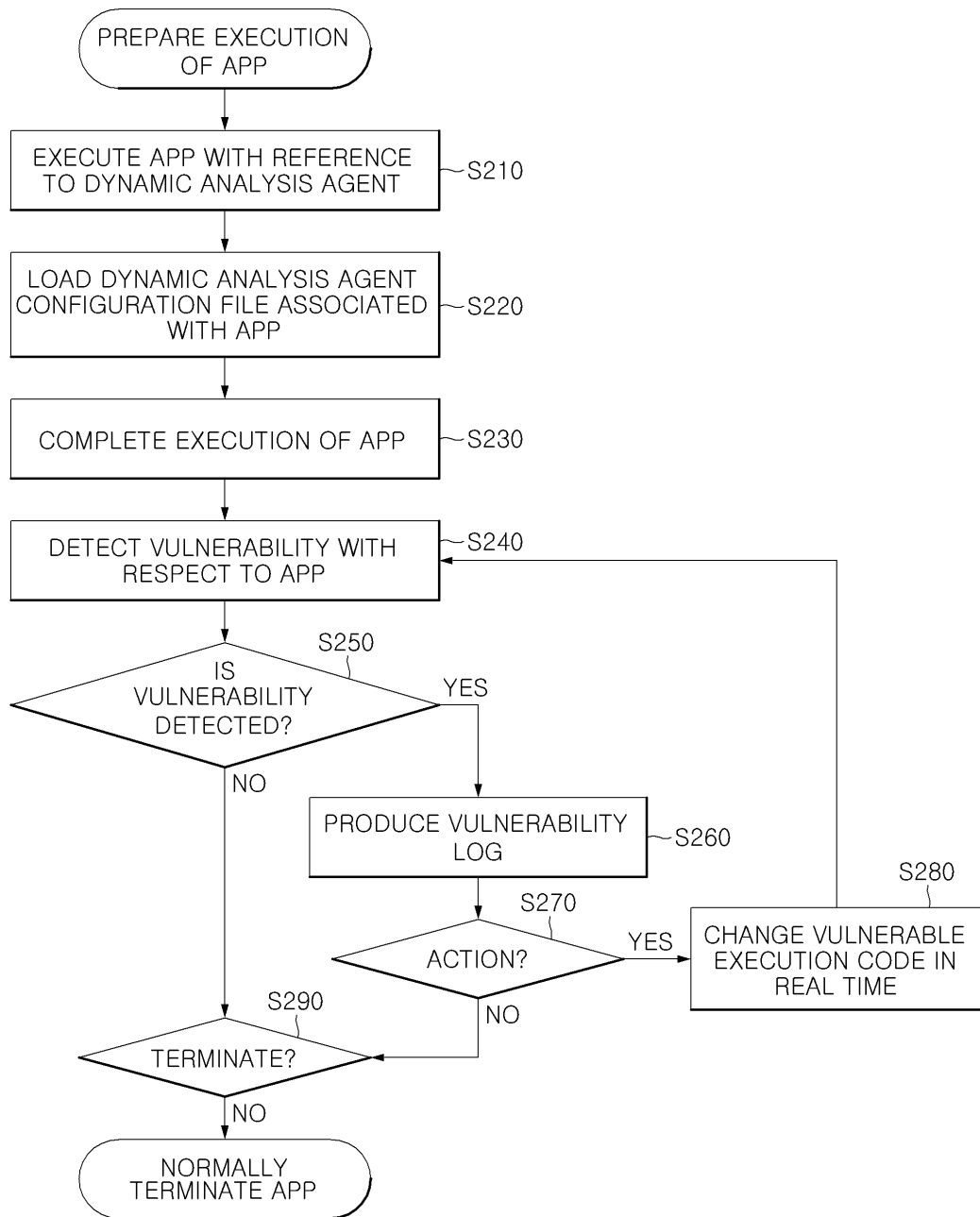


FIG. 5

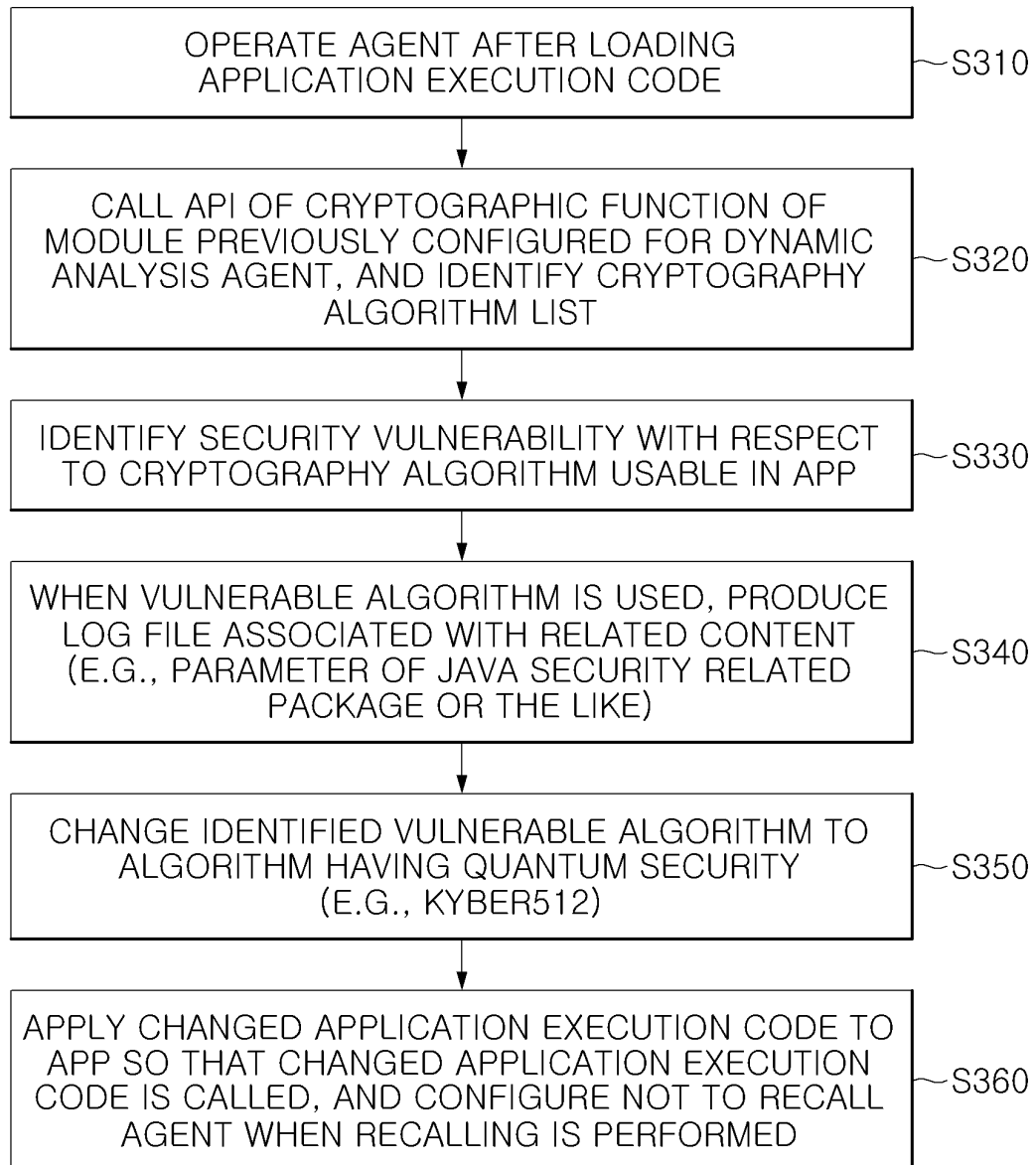


FIG. 6

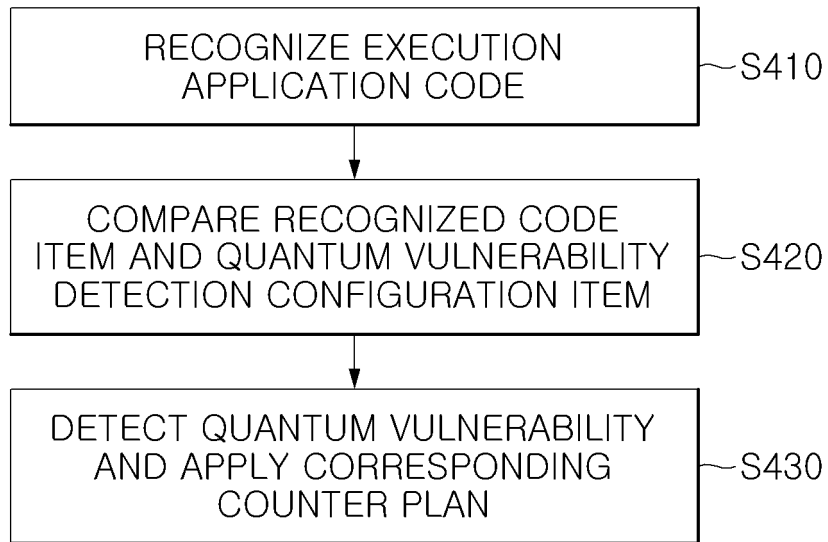


FIG. 7

MODULE NAME	METHOD NAME	METHOD PARAMETER TYPE	METHOD RETURN TYPE	RESPONSE TYPE AFTER DETECTION
moduleA (ex. Verifier)	methodB (ex. verify)	paramC paramD ... (ex. String, bytes)	value (ex. bool)	CHANGE CODE IN REAL TIME AFTER LOGGING
moduleF (ex. Signer)	methodG (ex. sign)	paramH paramI ... (ex. String, int)	value (ex. bytes)	CHANGE CODE IN REAL TIME AFTER LOGGING

(A)

FIG. 8

MODULE NAME	METHOD NAME	METHOD PARAMETER VALUE	METHOD RETURN VALUE	RESPONSE TYPE AFTER DETECTION
moduleF (ex. Signer)	methodG (ex. sign)	paramH paramI ... (ex. RSA σ / 2048)	valueJ (ex. bytes)	APPLY CHANGING CODE AFTER LOGGING ※ CHANGE RSA/2048 PARAMETER TO KYBER/512 AND APPLY
moduleF	methodG (ex. sign)	paramH paramI ... (ex. kyber σ / 512)	valueJ (ex. bytes)	APPLY CHANGING CODE AFTER LOGGING ※ REASON: QUANTUM SAFETY ALGORITHM (KYBER/512) IS ALREADY APPLIED

(A)

(B)

FIG. 9A

```
SecretKey aesKey = keyGenerator.generateKey();  
Cipher cipher = [Cipher.getInstance("RSA");]  
cipher .init(1, bobPublicKey);  
byte[] encryptKey = cipher.doFinal(aesKey.getEncoded());
```

910

FIG. 9B

```
SecretKey key = kg.generateKey();  
Cipher w1 = [Cipher.getInstance("KYBER", "BCPQC");]  
w1.init(3, kp.getPublic(), (AlgorithmParameterSpec)ktsParameterSpec);  
byte[] data = w1.wrap(key);
```

920

FIG. 10

