**Solution I:**

***Series of events for communication between LATE and APL:***

1. APL gets the machine information and construct a challenge string
2. APL sets some package information in that string.
3. APL encrypts the CS (challenge string) with APL’s private key.
4. LATE gets the serial number and the CS
5. LATE decrypts the CS using APL’s public key.
6. LATE validates the relation between serial number and CS and checks for its validity.
7. LATE parses the CS and adds more meta information in it. A response string (RS) is constructed.
8. LATE encrypts the RS using its private key.
9. APL gets the encrypted RS
10. APL decrypts the RS using LATE’s public key
11. APL gets the information out of it and compares the properties with that of original CS.
12. APL returns pass or failure on the basis of the comparison.

**NOT POSSIBLE DUE TO LARGE LENGTH OF ENCRYPTED TEXT**

***Drawbacks:***

1. Encryption of original CS will have to be followed by base 32 encoding which means a 60% increase in length. So, 16 digit CS string will become about 25 digit long number.

**Solution II:**

***Series of events for communication between LATE and APL:***

1. APL computes the machine id along with some randomness in it.
2. APL encodes the machine id on a primitive level
3. LATE encrypts it using the symmetric key
4. LATE sends back random 10 characters from the encrypted text and the position of these random characters is encoded in a 3 byte long block. Rest 3 bytes will tell the time for which the package is to be activated.
5. There will be a maximum limit of the package activation.

***Contract between LATE and APL for CS and RS***

APL constructs the original CS which is a 16 digit number. 10 digits are processed on the digit level and 4 digits are on the bits level. Remaining 2 digits are used for future purpose for extra information.