1. If symmetric key is compromised

If symmetric key is compromised the person with the key can read the message sent. But he may not be able to change content of message as it is signed by the sender. If he tries to do that the receiver will know that message was changed and would not consider it valid.

2. Techniques to share symmetric key.Improve security of this process

1. Diffie Hellman key exchange

2. Steganography

3. Transporting the key physically

3. All security measures required. Example key storage, key recovery

**1. KeyStorage**

Keystores :Keystores are files containing your public/private keypair. Unlike the locally-stored OS and browser keystores, these files can be stored virtually anywhere, including remote servers and are always password protected (meaning any time you want to use your private key, you have to enter a STRONG password).  Another appeal is that since these are ultimately just files, you can easily distribute copies if you have multiple people who need to use the certificate. On other hand, since they are just files, they are susceptible to being distributed insecurely.  With rapid advances in password cracking algorithms, be sure to [create a sufficiently long, random password](https://www.globalsign.com/en/blog/how-to-create-a-strong-password/) if you use this method.

Cryptographic Tokens and Smart Cards: storing your private key on hardware can offer increased security. However, there is a big difference between using cryptographic tokens or smart cards and standard flash or thumb drives. With cryptographic hardware, the key is generated on the hardware itself and is not exportable. This means the private key never leaves the device, making it much more difficult for someone to access and compromise. Using a crypto token will also prompt for a password each time you want to use your certificate. This means even if someone gets a hold of your token, they would still need your password before being able to use it. Storing your key on a token means you can also securely use the same certificate across multiple machines without having to make multiple copies and going through the export/import process. Cryptographic hardware can also help meet [FIPS compliance](http://csrc.nist.gov/groups/STM/cmvp/standards.html), which is required for some industry and government regulations.

### 2. Weak keys

A key is essentially just a random number – the longer and more random it is, the more difficult it is to crack. The strength of the key should be appropriate for the value of the data it is protecting and the period of time for which it needs to be protected. The key should be [long enough for its intended purpose](https://en.wikipedia.org/wiki/Key_size) and generated using a high-quality (ideally certified) random number generator (RNG), [ideally collecting entropy from a suitable hardware noise source](https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90B.pdf).There are [many instances](https://en.wikipedia.org/wiki/Random_number_generator_attack#Prominent_examples) where poor RNG implementation has resulted in key vulnerabilities.

### 3. Non-rotation of keys

If a key is over-used (e.g. used to encrypt too much data), then it makes the key [more vulnerable to cracking](https://sweet32.info/SWEET32_CCS16.pdf), especially when using older symmetric algorithms; it also means that a high volume of data could be exposed in the event of key compromise. To avoid this, keys should be rotated (i.e. updated / renewed) at appropriate intervals.