6.1.3 Backwards Compatibility Attacks

Use of state-of-the-art and secure encryption algorithms such as RSA-OAEP and AES-GCM can become insecure when the adversary can force the server to process eavesdropped ciphertext with legacy algorithms such as RSA-PKCS#1 v1.5 or AES-CBC [*NDSS-2012-TLS*]:

1. The attacker ~~can~~ may be able to break the security of an AES-GCM ciphertext if he is able to force the server to process the ciphertext with AES-CBC and the same symmetric key.
2. The attacker ~~can~~ may be able to decrypt an RSA-OAEP ciphertext if he is able to force the server to process the ciphertext with RSA-PKCS#1 v1.5 and the same asymmetric key.
3. The attacker ~~can~~ may be able to forge valid server signatures if the server decrypts RSA-PKCS#1 v1.5 ciphertexts and the signatures are computed with the same asymmetric key pair.

Accordingly, in situations where an attacker may be able to mount chosen-ciphertext attacks, we recommend the following to implementers:

1. ~~2. It is a bad cryptographic practice to apply the same cryptographic keys for different cryptographic tasks and algorithms. We recommend enforcing~~ Always use of different keys for public key ~~end~~decryption and signature processing (ciphertext decryption and signature creation) and different keys for different algorithms, even if serving a similar function. This can be done using derived keys basing the derivation on the algorithm identifier, for example
2. ~~1.~~ When appropriate, restrict algorithm usage to algorithms known to be secure in the face of chosen-ciphertext attacks (RSA-OAEP, AES-GCM). In that case, documents containing RSA-PKCS#1 v1.5 and AES-CBC ciphertexts ~~must~~ may be rejected without decryption. ~~Allowing use of either RSA-PKCS#1 v1.5 or AES-CBC is dangerous~~.