SGX security background

Monday, April 10, 2017 4:10 PM

3个基本概念

Confidentiality	消息不会被其他人读取	通过加密来保证	AES-GCM, AES-CTR	RSA with PKCS #1 v2.0
Integrity	消息的完整性不会被破坏，或者被破坏后能意识到被攻击了	MAC/signature	HMAC-SHA-2 AES-GCM	DSS-RSA, DSS-ECC
freshness	在integrity的基础上，接收者总是能收到最新的消息，或者能意识到攻击。	Nonces+integrity		

每个加密原语都需要通过一个随机数来生成独一无二的key，而随机数的生成是通过（CSPRNG）来做到的，他的值不应该被预测到。

Confidentiality
对称加密：双方使用同样的key加密和解密，因此key的分发必须能够保证confidentiality 和integrity
非对称加密：使用public key进行加密，private key进行解密。Public key的分发不用保证confidentiality，只需要有integrity。

对称加密中比较有名的有AES，将一个128bit的block转换为另一个128bit的block，最近要求使用256 bit长的key。
非对称加密的RSA.通常非对称加密算法比对称加密算法需要的计算量要大得多，因此通常发送者先用非对称加密的公钥加密一个一次使用的key发送给接收者。如下图所示：

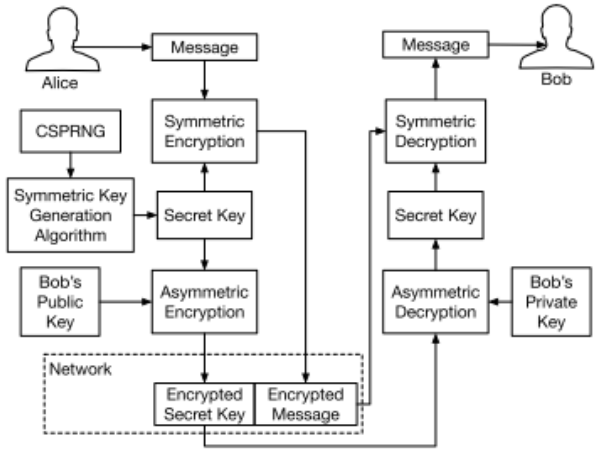
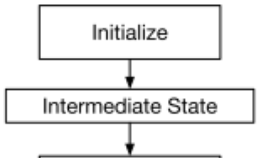


Figure 40: Asymmetric key encryption is generally used to bootstrap a symmetric key encryption scheme.

Integrity
通过 secure hashing functions提供。SHA-2 256 bits
Sha-2 将message 分为block，通过不停的extend block到中间状态，最后得出最终状态。其中，message block 和中间状态的长度是不变的。如下图：



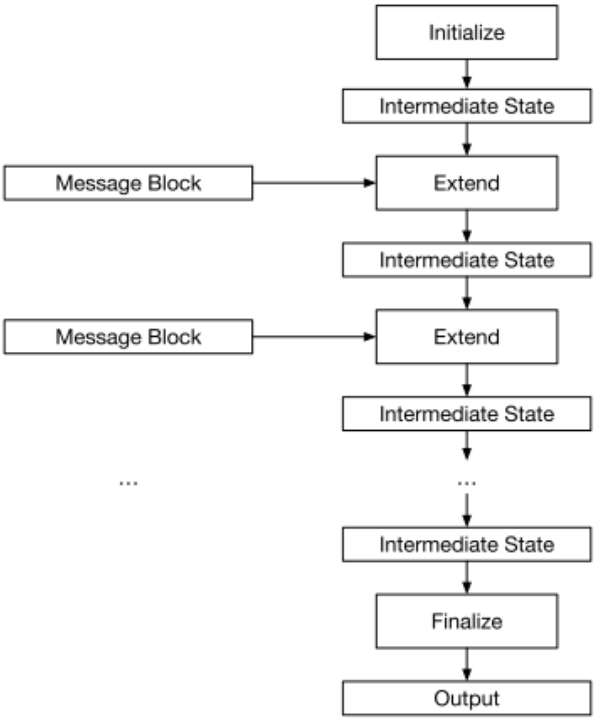
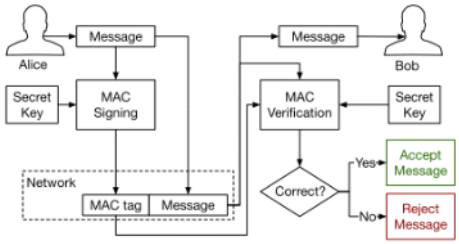


Figure 41: A block hash function operates on fixed-size message blocks and uses a fixed-size internal state.



在对称加密算法中，Message Authenitication code （ MAC ）用来保证integrity。
通常MAC没有特殊的算法，直接使用对应的对称加密算法来解密MAC
HMAC（Hash MAC)可以使用任何secure hash算法来生成MAC。

非对称密钥的原语可以提供个integrity保证是signatures 。sender用自己的私钥进行签名算法。

Freshness

CA 机制
CA（Certificate Authorities）。
Certificate通常是一个用CA的私钥进行的加密签名。

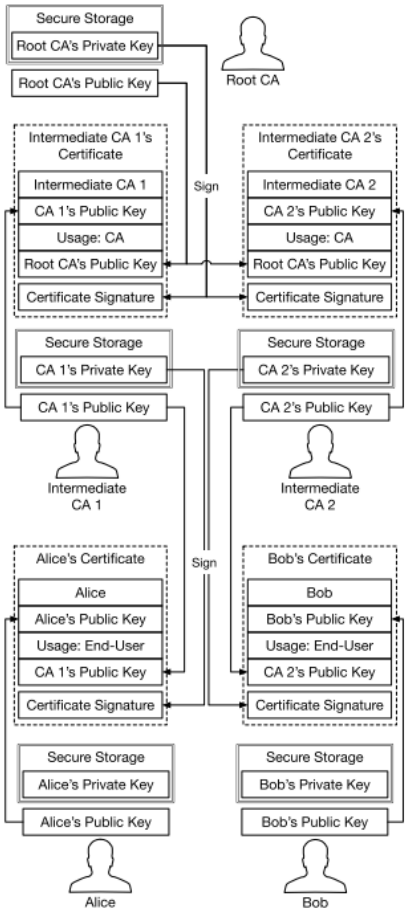


Figure 49: A hierarchical CA structure minimizes the usage of the root CA's private key, reducing the opportunities for it to get compromised. The root CA only signs the certificates of intermediate CAs, which sign the end users' certificates.

为了降低Root CA被攻击的可能性，CA的机制通常是层次性的，如上图所示。

Key agreement protocols

密钥交换协议通常用来在不安全的通信条件下给双方建立一个共享密钥。如下图所示

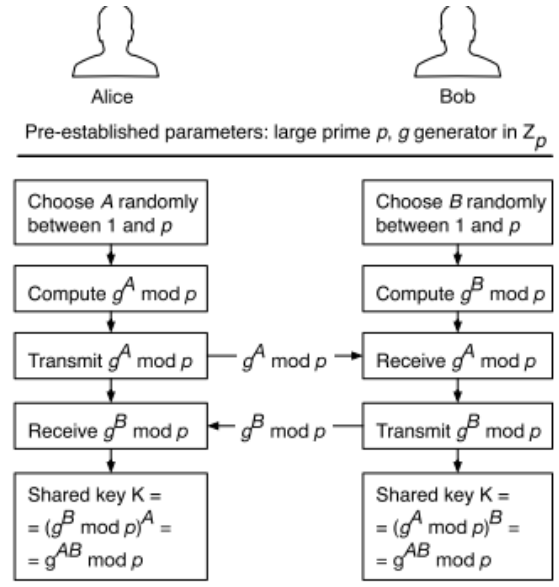


Figure 51: In the Diffie-Hellman Key Exchange (DKE) protocol, Alice and Bob agree on a shared secret key $K = g^{AB} \mod p$. An adversary who observes $g^A \mod p$ and $g^B \mod p$ cannot compute K.

Alice 和Bob同时 创建A和B，然后计算g^A mod p 和g^B mod p， 并交换这两个mod。
利用数学上的特性，即g^{AB} mod p = $(g^A \bmod p)^B = (g^B \bmod p)^A$ 同时，攻击者就算知道了其中任何一个余数，也很难猜到其他的信息或者计算出g^{AB} mod p, 以此完成密钥交换。

同时，如下图所示，DKE算法无法识别中间人攻击，因此必须结合其他的方法来识别：

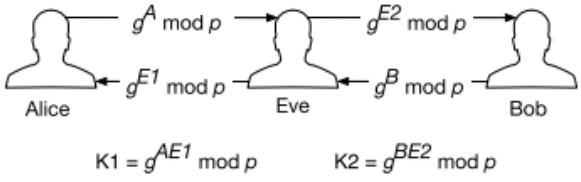


Figure 52: Any key agreement protocol is vulnerable to a man-in-the-middle (MITM) attack. The active attacker performs key agreements and establishes shared secrets with both parties. The attacker can then forward messages between the victims, in order to observe their communication. The attacker can also send its own messages to either, impersonating the other victim.

如当存在CA是，双方中最后一个发出消息的（Bob）用自己的公钥将key agreement签名同时向Alice 发送自己的certificate。Alice可以验证这个消息是否真的由Bob所发。