第 4 次作业

学号：2017221302006　　姓名：周玉川

噪声协议是一种基于 Diffie-Hellman 密钥协议的密码协议框架.噪声可以描述由单个消息和交互协议组成的协议。噪声协议的核心是每一方在握手过程中维护的一组变量,以及通过顺序处理来自消息模式的令牌来发送和接收握手消息的规则。每一方保持以下变量：

s, e 本地静态和短暂的密钥对(可能是空的)。

rs, re*远程静态和短暂的公钥(可能是空的)。

h*a 握手散列值，该值对已发送和接收的所有握手数据进行散列。

ck*a 链式钥匙它会散列所有以前的 DH 输出。握手完成后，将使用链接密钥派生传输消息的加密密钥。

k, n*加密密钥 k(可能是空的)和一个基于计数器的现在。n...每当一个新的 DH 输出导致一个新的 ck 要计算，一个新的 k 也是计算出来的。钥匙 k 而现在 n 用于加密静态公钥和握手有效载荷。并使用当前 h 价值为关联数据这是由 AIAD 认证覆盖的。静态公钥和有效载荷的加密在握手阶段提供了一些机密性和密钥确认。

握手消息由一些 DH 公钥组成,后面跟着一个有效载荷,有效载荷可能包含应用程序选择的证书或其他数据。要发送握手消息，发送方指定有效负载并顺序处理消息模式中的每个令牌。

噪声协议被实例化为一组具体的 DH 函数, 密码函数，和散列函数。

噪声取决于以下因素 DH 函数(及相关常数)：

1. GENERATE_KEYPAIR()生成一个新的 Diffie-Hellman 密钥对。DH 密钥对由 public_key 和 private_key 元素。一个 public_key 将 DH 公钥的编码表示为长度的字节序列。DHLEN...这个 public_key 编码细节特定于每一组 DH 函数。

2. DH(key_pair, public_key)中的私钥之间执行 Diffie-Hellman 计算。key_pair 而 public_key 并返回长度字节的输出序列。DHLEN...为了安全起见，基于此函数的 Gap-DH 问题必须由任何实际的密码分析对手解决.

public_key 要么编码一个大型素数级组中的生成器的值(该值可能具有多个等效编码)，要么是一个无效的值。实现必须通过返回一些输出来处理无效的公钥，这些输出纯粹是公钥的一个函数，不依赖于私钥，或者向调用方发出错误信号。DH 函数可以为处理无效值定义更具体的规则。

DHLEN=指定公钥和 DH 输出的字节大小的常量。出于安全原因，DHLEN 一定是 32 岁或以上。

噪声取决于以下因素密码函数:

1.ENCRYPT(k, n, ad, plaintext)*加密 plaintext 使用密码密钥 k 为 32 个字节和一个 8 字节的无符号整数 n 键必须是唯一的。k...返回密文。加密必须使用关联数据的“aead”加密模式进行。ad(使用来自[1])，并返回与明文大小相同的密文加上用于身份验证数据的 16 个字节。如果密钥是秘密的，则整个密文必须与随机密文无法区分.

2.DECRYPT(k, n, ad, ciphertext)*解密 ciphertext 使用密码密钥 k 为 32 字节，为 8 字节无符号整数。n，以及相关数据 ad...返回明文，除非身份验证失败，否则将向调用方发出错误信号。

3.REKEY(k)的伪随机函数返回一个新的 32 字节密码密钥。k...如果该函数不是为某些密码函数专门定义的，则默认为从 ENCRYPT(k, maxnonce, zerolen, zeros)，在哪里 maxnonce 等于 264-1, zerolen 是一个零长度字节序列，并且 zeros 由 32 个字节组成的序列，其中填充了零。

噪声取决于以下因素散列函数(及相关常数)：

1. HASH(data)::使用抗冲突密码散列函数散列一些任意长度的数据，并返回 HASHLEN 字节。2.HASHLEN=指定哈希输出的大小(以字节为单位)的常量。

BLOCKLEN=一个常数，指定哈希函数内部使用的以字节为单位的大小来划分其输入以进行迭代处理。

3.噪声定义了基于上述功能的附加功能。HASH()职能：

4.HMAC-HASH(key, data)*适用 HMAC 从…[3]使用 HASH()功能。此函数仅作为 HKDF()，在下面。

HKDF(chaining_key, input_key_material, num_outputs)* 采取 chaining_key 字节长度序列 HASHLEN，和一个 input_key_material 长度为零字节、32 个字节或 DHLEN 字节。返回两个或三个字节序列，每个字节序列的长度为 HASHLEN，取决于 num_outputs 是两三个：

```
temp_key = HMAC-HASH(chaining_key, input_key_material).
output1 = HMAC-HASH(temp_key, byte(0x01)).
output2 = HMAC-HASH(temp_key, output1 || byte(0x02)).
If num_outputs == 2 while return (output1, output2).
output3 = HMAC-HASH(temp_key, output2 || byte(0x03)).
Return (output1, output2, output3).
```

噪声的核心是这个由变量、标记和处理规则组成的简单系统，它允许简洁地表达一系列协议。它提供预共享对称密钥或 PSK 模式以支持双方都有一个 32 字节共享密钥的协议.

噪声协议有一个开端的输入，它允许将任意数据散列到 h 变量。如果双方不提供相同的开端的数据，握手将因解密错误而失败。当双方在握手之前进行谈判，并希望确保他们对谈判有相同的看法时，这个开端的输入是有用的。例如，假设 Bob 向 Alice 发送了他愿意支持的噪声协议列表。然后，Alice 将选择并执行单个协议。为了确保“中间人”不编辑 Bob 的列表以删除选项，Alice 和 Bob 可以将列表作为噪声协议的开端数据。