The Noise Protocol Framework

noise是基于Diffie-Hellman密钥协议的加密协议的框架。noise可能描述了包含单个消息的协议以及交互协议。noise协议始于两方交换**握手消息**。在此**握手阶段**，双方会交换DH公钥并执行一系列DH操作，将DH结果散列为共享密钥。在握手阶段之后，各方都可以使用此共享密钥来发送加密的**传输消息。**

握手消息由一些DH公共密钥和有效载荷组成。有效负载可能包含证书或应用程序选择的其他数据。为了发送握手消息，发送方指定有效负载并按消息模式顺序处理每个令牌。

所有noise消息的长度均小于或等于65535字节。限制邮件大小有几个优点：测试更简单，因为可以轻松测试最大尺寸；减少内存处理错误或整数溢出的可能性；支持大型数据流的流解密和随机访问解密；使封装noise消息的高层协议能够使用16位的有效标准长度字段。

**noise变量**

noise的核心是在握手过程中各方维护的一组变量，以及通过顺序处理消息模式中的令牌来发送和接收握手消息的规则。

各方维护以下变量：

s，e：本地用户的静态和临时密钥对（可能为空）。

rs，re：远程方的静态和临时公共密钥（可能为空）。

h：一个握手哈希值，该哈希值散列已发送和接收的所有握手数据。

ck：对所有先前的DH输出进行哈希处理的链接密钥。一旦握手完成，链接密钥将用于导出传输消息的加密密钥。

k，n：加密密钥k（可以为空）和基于计数器的随机数n。每当新的DH输出导致要计算新的ck时，也会计算新的k。密钥k和随机数n用于加密静态公共密钥和握手有效负载。使用k加密使用某种AEAD密码模式，并将当前的h值用作AEAD身份验证涵盖的关联数据。静态公共密钥和有效负载的加密在握手阶段提供了一些机密性和密钥确认。

可能的标记为：

“ e”：发送者生成一个新的临时密钥对，并将其存储在e变量中，将临时公共密钥作为明文写入消息缓冲区，并将该公共密钥与旧的h一起哈希以得出新的h。

“ s”：发送方将其从s变量的静态公钥写入消息缓冲区，如果k为非空，则对其进行加密，然后将输出与旧的h一起哈希以得出新的h。

“ ee”，“ se”，“ es”，“ ss”：在发起者的密钥对（由首字母确定是静态还是临时的）和响应者的密钥对（由静态确定或短暂的）之间进行DH第二个字母）。将结果与旧ck一起进行哈希处理以得出新的ck和k，并且n设置为零。在握手消息中处理了最终令牌之后，发送方将有效负载写入消息缓冲区，如果k为非空，则对其进行加密，然后将输出与旧的h一起哈希以得出新的h。举一个简单的例子，握手模式描述了未经身份验证的DH握手：-> e <-e，ee发起方发送第一条消息，它只是一个短暂的公共密钥。响应者发回自己的临时公共密钥。然后执行DH，并将输出散列到共享密钥中。请注意，在明文短暂的公共密钥之后，在第一条消息中发送明文有效载荷，在明文短暂的公共密钥之后，在响应消息中发送加密的有效载荷。应用程序可以发送所需的任何有效负载。

在握手消息中处理了最终令牌之后，发送方将有效负载写入消息缓冲区，如果k为非空，则对其进行加密，然后将输出与旧的h一起哈希处理以得出新的h。

**握手模式**

消息模式是来自集合的一些令牌序列（“ e”，“ s”，“ ee”，“ es”，“ se”，“ ss”，“ psk”）。未来的规范可能会引入其他令牌。消息前模式是以下令牌序列之一：

* + “ e”
  + “ s”
  + “ e，s”
  + empty

握手模式包括：

* 启动器的消息前模式，表示响应者已知的有关启动器公钥的信息。
* 响应者的消息前模式，表示发起者已知的有关响应者公钥的信息。
* 实际握手消息的消息模式序列。

预先消息表示在握手之前以某种方式执行的公共密钥交换

举一个简单的例子，握手模式描述了未经身份验证的DH握手：

-> e

<-e，ee

发起方发送第一条消息，它只是一个短暂的公共密钥。响应者发回自己的临时公共密钥。然后执行DH，并将输出散列到共享密钥中。请注意，在明文短暂的公共密钥之后，在第一条消息中发送明文有效载荷，在明文短暂的公共密钥之后，在响应消息中发送加密的有效载荷。应用程序可以发送所需的任何有效负载。

响应者可以发送其静态公钥（经过加密），并通过略有不同的模式进行身份验证：

-> e

<-e，ee，s，es

在这种情况下，最终的ck和k值是两个DH结果的哈希值。由于es令牌指示发起者的临时密钥与响应者的静态密钥之间的DH，因此，发起者对第二条消息的有效负载进行的成功解密可用于向发起者认证响应者。请注意，第二条消息的有效负载可能包含零长度的明文，但是有效负载密文仍将包含身份验证数据（例如身份验证标签或“合成IV”），因为加密是通过AEAD模式进行的。第二条消息的有效负载也可以用于为响应者的静态公钥传递证书。启动器可以使用握手模式和一条附加消息来发送其静态公共密钥（经过加密）并进行身份验证：

-> e

<-e，ee，s，es

-> s，se