第八章　密钥分发和密钥协商

张磊

华东师范大学 • 软件学院

§8.1　密钥分发

密钥分发

解决秘密密钥建立问题的常见方法包括：

1. 密钥预分发
2. 会话密钥分发
3. 密钥协商

密钥预分发

1. 密钥预分发方案中，TA以一种安全的方式为网络中的每个用户“提前”分发密钥信息，注意密钥分发时需要一个安全信道。此后，所有的网络用户可以使用这些秘密密钥来加密其在网络中传输的消息。
2. 该方案本质是网络中的每一对用户都能够根据他们掌握的密钥信息来确定一个密钥，且该密钥只有他们二人知道。

会话密钥分发

1. 在会话密钥分发中，当网络用户请求会话密钥时，一个在线的TA选择会话密钥并通过一个交互协议分发给他们。这样的协议称为会话密钥分发方案并记为SKDS。

密钥协商

1. 密钥协商是指网络用户通过一个交互协议来建立会话密钥的情形。这样的协议称为密钥协商方案，并且记为KAS。
2. 密钥协商可以基于对称密码体制，也可以基于公钥密码体制，通常不需要一个在线的TA。

长期密钥和会话密钥的区分

1. 长期密钥(Long Live Key)指预先计算并安全存储的密钥。或者说，长期密钥可以根据需要由安全存储的秘密信息非交互地计算出来。
2. 长期密钥可以是秘密密钥，为一对用户共同拥有，或者为一个用户与TA共同拥有。也可以是一个私钥，它与存储在用户证书中的公钥相对应。
3. 会话密钥指在特定的会话中，一对用户经常使用的秘密的短期会话密钥，当会话结束时就会把该会话密钥丢弃。会话密钥通常是秘密密钥(也称对称密钥)，用于对称密码体制或者MAC。
4. 长期密钥通常用于协议中传输加密的会话密钥。

Diffie-Hellman 密钥预分发

1. 公开的域参数包括：群 $(G, \bullet)$，一个阶为n的元素 $\alpha \in G$
2. V利用U的证书中的公钥 b_U 和他自己的私钥 a_V 计算

$$K_{U,V} = \alpha^{a_U a_V} = b_U^{a_V}$$

3. U利用V的证书中的公钥 b_V 和他自己的私钥 a_U 计算

$$K_{U,V} = \alpha^{a_U a_V} = b_V^{a_U}$$

注意

这个密钥预分发方案并未进行任何交互，并且假定用户的私钥是安全的，所以不需要考虑主动敌手的可能性。

显然，如果CDH问题是困难的，那么该方案就是计算安全的。

平凡构造:

1. TA选择随机密钥$K_{U,V} = K_{V,U}$
2. ‘离线’安全信道传送给U, V
3. 每个用户必须存储$n-1$个密钥，且TA需要安全地传送 C_n^2 个密钥时，代价将会过高。

Blom 密钥预分发方案可以减少需要传输和存储的信息数量，并且仍然使得每个用户 U 和 V 能够（独立地）计算一个私密密钥$K_{U,V}$。

Blom 密钥预分发方案

1. 素数 p公开，用户U_i公布一个元素$r_i \in \mathbb{Z}_p$，元素 r_i必须时不同的。
2. TA选择三个随机数 $a, b, c \in \mathbb{Z}_p$ ，并构造多项式
$$f(x, y) = a + b(x + y) + cxy \bmod p$$
3. TA为用户U_i计算
$$g_i(x) = f(x, r_i)$$
并通过安全信道向U_i发送
4. U_i和U_j进行安全通信，分别计算密钥
$$k_{ij} = g_i(r_j), k_{ji} = g_j(r_i)$$

Needham-Schroeder会话密钥方案

1. Alice选择一个随机数r_A，向TA发送ID(Alice),ID(Bob)和r_A
2. TA选择一个随机的会话密钥K。然后计算票据

$$t_{Bob} = e_{K_{Bob}}(K \parallel ID(Alice))$$

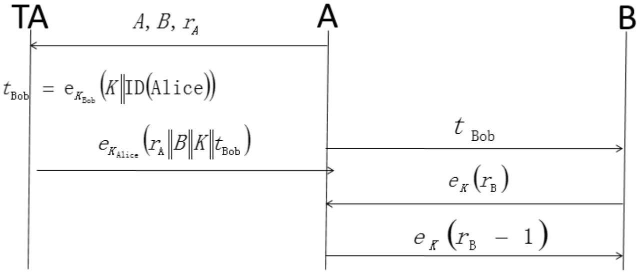
以及

$$y_1 = e_{K_{Alice}}(r_A \parallel ID(Bob) \parallel K \parallel t_{Bob})$$

并发送y_1给Alice(这里假设TA 与每个用户都共享一个秘密密钥)

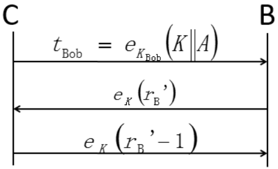
Needham-Schroeder会话密钥方案

1. Alice使用密钥K_{Alice}解密y_1得到K和t_{Bob}，然后发送t_{Bob}给Bob
2. Bob使用密钥K_{Bob}解密t_{Bob}得到K，然后Bob选择随机数r_{Bob}并发送$y_2 = e_K(r_B)$给Alice
3. Alice使用密钥K解密y_2得到r_B，然后发送$y_3 = e_K(r_B - 1)$给Bob



对NS方案的Denning-Sacco攻击(已知会话密钥攻击)

假设C以某种方式得到了会话S的会话密钥K，那么攻击会话S'流程如下：



会话S'之后，Bob认为他产生了一个“新”的与Alice共享的会话密钥K。C知道这个密钥，但是Alice 未必知道，因为在与Bob进行的前一次会话S结束之前Alice可能已经扔掉了密钥K。

所以，在这个攻击中Bob从两个方面被欺骗了：

1. Bob所预期的对等方并不知道在会话S'中分配的密钥K
2. 会话S'的密钥K被其他人知道（即C）。

Kerberos密钥分发方案

Kerberos是MIT于20世纪80年代后期和90年代早期开发出来的一个系列会话密钥分发方案。这里给出该方案的第5版简化形式。

1. Alice选择一个随机数r_A,并发送ID(Alice),ID(Bob)和r_A给TA。
2. TA选择一个随机会话密钥K，以及一个有效期L。计算t_{Bob}和y_1并发送给Alice

$$t_{Bob} = e_{K_{Bob}}(K \parallel ID(Alice) \parallel L)$$

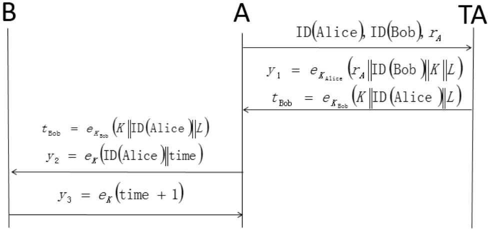
$$y_1 = e_{K_{Alice}}(r_A \parallel ID(Bob) \parallel K \parallel L)$$

1. Alice使用她的密钥K_{Alice}解密y_1，得到K。然后Alice根据当前时间time计算y_2，与t_{Bob}一起发送给Bob

$$y_2 = e_K(ID(Alice) \parallel time)$$

2. Bob使用K_{Bob}解密t_{Bob}，得到K。还要用K解密y_2得到time。然后Bob计算y_3发送给Alice

$$y_3 = e_K(time + 1)$$



NS方案与V5 Kerberos方案

1. NS方案中，TA通过Alice给Bob的票据t_{Bob}需要利用Alice的密钥重新进行加密，这没有任何益处，因为Alice发送该票据给Bob 后，其他人都可以看到这条票据；Kerberos方案中并未对该票据进行双重加密。
2. NS方案存在Denning-Sacco攻击，但Kerberos 方案通过检查时间和有效期来限定了实施Denning-Sacco攻击的时间周期。
3. Kerberos方案中，第三步和第四步利用K加密时间的目的是发送方向接收方证明自己知道密钥。

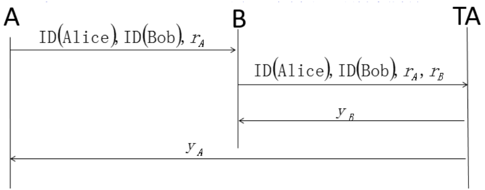
Bellare-Rogaway方案

1995年，Bellare和Rogaway提出了一个会话密钥分发方案

1. Alice选择一个随机数r_A，与ID(Alice),ID(Bob)一起发送给Bob。
2. Bob选择一个随机数r_B，与ID(Alice),ID(Bob),r_A一起发送给TA。
3. TA选择一个随机的会话密钥K,计算y_B, y_A。发送y_B给Bob，y_A给Alice。

$$y_B = (e_{K_{Bob}}(K), MAC_{Bob}(ID(Alice) \parallel ID(Bob) \parallel r_B \parallel e_{K_{Bob}}(K)))$$

$$y_A = (e_{K_{Alice}}(K), MAC_{Alice}(ID(Bob) \parallel ID(Alice) \parallel r_A \parallel e_{K_{Alice}}(K)))$$



Bellare-Rogaway方案

Bellare-Rogaway方案注意事项：

1. TA中生成的挑战是采用完美的随机数生成器产生的。
2. 该方案没有实现密钥确认。也就是说，Alice/Bob在收到会话密钥后无法确认Bob/Alice 也获得了会话密钥。
3. 该方案可以保证任何其他人都不能计算出新的会话密钥。否则，就存在一个多项式时间算法要么可以攻陷加密算法，要么可以攻陷MAC算法。

Bellare-Rogaway方案证明思路

BR是一个密钥分发方案，因此攻陷该方案就是敌手执行协议并使得双方被分发得到两个不同的会话密钥。

1. 首先，在协议的多次(总次数是密钥长度的一个多项式)执行过程中，Alice/Bob选择的随机挑战出现重复的概率可以忽略不计，因为挑战长度与密钥长度是同一个数量级的。
2. 其次，在Alice/Bob选择的随机挑战没有重复的情况下，为了攻陷协议，敌手在不知道TA与Alice/Bob共享密钥的情况下，要么找到一个$K'(K' \neq k)$，使得 $e_{K_{Alice}}(K) = e_{K_{Alice}}(K')$或者$e_{K_{Bob}}(K) = e_{K_{Bob}}(K')$；要么对某个$e_{K_{Alice}}(K)$或者$e_{K_{Bob}}(K)$伪造出对应的MAC值。也就是说，敌手要么攻陷了加密算法，要么攻陷了MAC算法。
3. 最后，由于加密算法和MAC算法是安全的，所以BR方案是安全的。

§ 8.2　密钥协商

密钥协商

密钥预分发方案和会话密钥分发方案都需要一个可信权威机构TA来选取密钥，并将他们分发给网络用户。

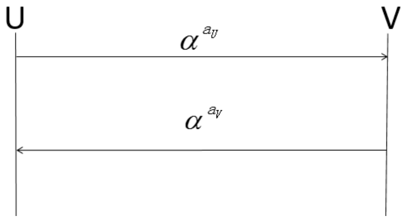
下面我们讨论密钥协商方案，这类方案不需要TA的参与，而是通过一个交互协议来共同确定一个新的会话密钥。

我们主要是在公钥环境下来讨论密钥协商方案。

Diffie-Hellman 密钥协商方案

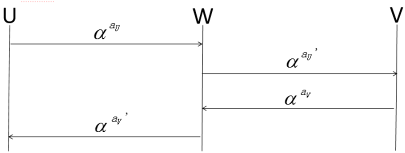
公开的域参数包括：群$(G,\bullet)$，一个阶为n的元素$\alpha \in G$

1. U选取一个随机数a_U,$0 \le a_U \le n-1$然后计算$b_U = \alpha_U^a$,并发送V
2. V选取一个随机数a_V,$0 \le a_V \le n-1$然后计算$b_V = \alpha_V^a$,并发送U
3. U计算$K = (b_V)^{a_U}$,V计算$K = (b_U)^{a_V}$



Diffie-Hellman密钥协商

对Diffie-Hellman密钥协商方案的中间人攻击



认证密钥协商方案的性质：

1. 协议执行完成后，双方协商出的密钥相同
2. 敌手实施任何攻击流程后，诚实参与者会“接受”的概率可忽略

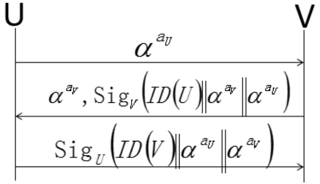
端-端密钥协商方案

端-端密钥协商方案(STS)是将Diffie-Hellman密钥协商方案和一个安全的交互识别方案结合在一起，构成一认证密钥协商方案。

1. 每个用户U有一个签名方案，其算法记为sig_U，验证算法为ver_U。
2. TA也有一个签名方案，其公开验证算法为ver_{TA}。
3. 每个用户有一个证书

$$\mathrm{Cert}(U)=(\mathrm{ID}(U),\mathrm{ver}_U,\mathrm{sig}_{TA}(\mathrm{ID}(U),\mathrm{ver}_U)),$$

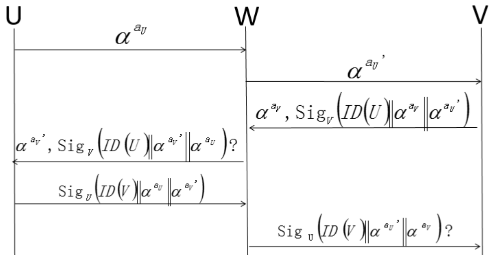
这里$\mathrm{ID}(U)$ 是U 的识别信息。



端-端密钥协商方案

简化的端-端密钥协商方案可抵抗中间人攻击

1. 假设W截取了α^{a_u}并替换为$\alpha^{a'_u}$，发送给V
2. W收到V发来的α^{a_v}和$sig_V(ID(U) \parallel \alpha^{a_v} \parallel \alpha^{a'_U})$。若W要将$\alpha^{a_v}$替换为$\alpha^{a'_v}$，由于W不知道V的签名算法$\mathrm{sig}_V$，所以无法计算新的$\mathrm{sig}_V$
3. 同理，W不知道U的签名算法，无法计算新的sig_U



三层保证机制

密钥协商方案的三层保证机制：

1. 隐式密钥认证:如果U可以被确保除了V之外，没有人能计算出K（特别是敌手），则该方案提供了隐式密钥认证。
2. 隐式密钥确认:如果U可以被确保V可以计算出K （假设V按照规定执行了方案），并且除了V 之外，没有人能计算出K，则该方案提供了隐式密钥确认。
3. 显式密钥确认:如果U可以被确保V已经计算出了K，除了V之外没有人能计算出K，则该方案提供了显式密钥确认。

Bellare-Rogaway会话密钥方案提供了隐式密钥认证。在该方案中双方都不能确认对方是否已经收到(或者能够计算出)会话密钥。

端-端密钥协商方案对交换的指数进行了数字签名，在DDH问题是难解的假设条件下，该方案为双方提供了隐式密钥确认特性。

Kerberos和Needham-Schroeder协议都提供了显式密钥确认。

已知会话密钥攻击

当实际的具有多个用户的网络环境下，可能会同时有多个端-端方案的会话发生。我们需要考虑不同的会话之间可能造成的影响。

因此，我们研究已知会话密钥攻击下端-端方案的安全性。即假设敌手可以被告知某些会话的会话密钥，如$S_1, S_2, S_3 \ldots S_t$，敌手希望借助这些会话密钥来得到其他会话的会话密钥。

下面我们证明如果DDH问题是难解的，那么端-端密钥协商方案在已知密钥攻击下是安全的。

现在我们来说明Oscar如何构造一个模拟J_{sim}的。

1. Oscar选取b_1
2. Oscar选取一个随机值a_2，并计算$b_2 = \alpha^{a_2}$
3. Oscar计算$b_3 = (b_1)^{a_2}$
4. Oscar定义$T_m = (b_1, b_2, b_3)$

无论是Oscar同伴选取b_2还是Oscar自身选取b_2，b_2都是均匀选取的。所以有：

$$Pr[T = (b_1, b_2, b_3)] = Pr[T_m = (b_1, b_2, b_3)]$$

又由于DDH问题的难解性，给定T, T_m对于敌手的攻击具有不可区分性，否则可以利用敌手的攻击来解决DDH 问题。

因此，无论Oscar利用已知会话密钥攻击做什么，他也可以使用一个完全的被动攻击做同样的事情。

公开的域参数包括：群$(G,\bullet)$，一个阶为n的元素$\alpha \in G$每个用户T有一个秘密指数a_T，其中$0 \le a_T \le n-1$对应的公开值为$b_T = \alpha^{a_T}$, b_T被包含在T的证书中并被TA签名。

1. U选取一个随机数$r_U, 0 \le r_U \le n-1$，计算

$$S_U = \alpha^{r_U},$$

然后U将$Cert(U)$和S_U发送给V

2. V选取一个随机数$r_V, 0 \le r_V \le n-1$，计算

$$S_V = \alpha^{r_V},$$

然后V将$Cert(V)$和S_V发送给U

MTI/A0密钥协商方案

1. V计算出会话密钥

$$K = S_U^{a_V} b_U^{r_V},$$

   其中b_U从$Cert(U)$中获得。

2. U计算出会话密钥

$$K = S_V^{a_U} b_V^{r_U},$$

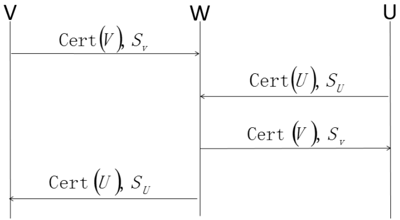
   其中b_V从$Cert(V)$中获得。

3. 会话结束时，U和V计算出相同的会话密钥

$$K = \alpha^{r_U a_V + r_V a_U}$$

对于MTI/A0的已知会话密钥攻击1

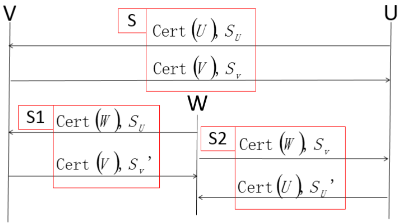
W是两个会话中的主动参与者，同时假装成V与U对话，假装成U与V对话。



之所以可以进行秉性对话攻击是因为密钥是双方提供的输入的一个对称函数值。为了消除这种攻击，可以使用一个Hash函数h作为密钥推导函数。

对于MTI/A0的已知会话密钥攻击2 (Burmester三角攻击)

W观察到V与U之间的会话S，然后参与他们之间的另外两个会话S_1和S_2。



当两个会话结束，分别请求密钥。

对于MTI/A0的已知会话密钥攻击2 (Burmester三角攻击)

会话S, S_1, S_2的密钥K, K_1, K_2分别如下：

$$K = \alpha^{r_U a_V + r_V a_U},$$

$$K_1 = \alpha^{r_U a_V + r'_V a_U},$$

$$K_2 = \alpha^{r'_U a_V + r_V a_U}$$

给定K_1, K_2，W可以计算

$$K = \frac{K_1 K_2}{(S'_V S'_U)^{a_W}}$$

因此这是一个成功的已知会话密钥攻击。

- Girault密钥协商方案结合了RSA和基于离散对数两种方案的特点。
- 假定$n = pq$，其中$p = 2p_1 + 1, q = 2q_1 + 1$，$p, q, p_1, q_1$都是大素数。乘法群 $\mathbb{Z}_n^*$ 与 $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$ 是同构的，因此 $\mathbb{Z}_n^*$ 中的一个元素，在p和q足够大的情况下，由α产生的 $\mathbb{Z}_n^*$ 中的循环子群比较适合用来构造离散对数问题。

- 在Girault密钥协商方案中，仅有TA是知道n是如何分解的。n和α是公共参数，而p, q，p_1, q_1都是秘密的。TA选取了一个公共的RSA加密指数，记为e。相对应的解密指数d是秘密的（通常，$d = e^{-1} \bmod \phi(n)$)
- 每个用户U有一个认证字符串ID(U)。一个用户U从TA中获取一个自认证公钥p_U。观察到U需要TA的帮助来产生公钥p_U。通过使用公开的信息可以从p_U和ID(U)计算出

$$b_U = p_U^e + \mathrm{ID}(U) \bmod n$$

使用自认证密钥的密钥协商

Girault公钥生成

注：n是公开的，d是仅由TA所知道的秘密值。

1. U选取一个秘密指数a_U，计算
$$b_U = \alpha^{a_U} \bmod n.$$
U 将b_U 给TA
2. TA计算
$$p_U = (b_U - ID(U))^d \bmod n,$$
TA将p_U给U

Girault密钥协商方案

1. U选取一个随机数r_U，计算
$$s_U = \alpha^{r_U} \bmod n,$$
然后U将$ID(U), p_U$ 和s_U发送给V。
2. V选取一个随机数r_V，计算
$$s_V = \alpha^{r_V} \bmod n,$$
然后V将$ID(V), p_V$ 和s_V发送给U。
3. U计算
$$K = s_V^{a_U}(p_V^e + ID(V))^{r_U} \bmod n$$
4. V计算
$$K = s_U^{a_V}(p_U^e + ID(V))^{r_V} \bmod n$$

加密密钥交换（EKE）是Diffie-Hellman KAS的一种变形，它使用口令来加密会话中传递的指数。

公开域参数由群（$G, \bullet$)和具有阶n的$\alpha \in G$构成。

- U随机选择a_U，$0 \leq a_U \leq n-1$。接着计算

$$b_U = \alpha^{a_U} \text{和} y_U = e_{pwd_{U,V}}(b_U)$$

  然后把ID(U)和y_U发送给V。

- V随机选择a_V，$0 \leq a_V \leq n-1$。接着计算

$$b_V = \alpha^{a_V} \text{和} y_V = e_{pwd_{U,V}}(b_V)$$

  然后把ID(V)和y_V发送给U。

加密密钥交换

- U计算

$$b_V = d_{pwd_{U,V}}(y_V)\text{和}K = (b_V)^{a_U}$$

V计算

$$b_U = d_{pwd_{U,V}}(y_U)\text{和}K = (b_U)^{a_V}$$

假定敌手无法获得关于$\mathrm{pwd}_{U,V}$的任何消息。口令$\mathrm{pwd}_{U,V}$仅仅是用来加密用户推导会话密钥的信息（如两个指数）。即使会话密钥被敌手掌握，也不会泄露任何关于未加密的指数和口令的信息。

- 会议密钥协商方案（CKAS）是一种网络中两个或更多用户组成的一个子集可以构建一个共享密钥的密钥协商方案。
- Burmester-Desmedt和Steiner-Tsudik-Waidner会议密钥协商方案中，m个用户，$U_0, ...U_{m-1}$计算一个共同的秘密密钥。他们都是建立在一个有限群的子群基础上，在该子群中，判定性Diffie-Hellman问题是困难的。

Burmester-Desmedt会议密钥协商方案

公开域参数由群$(G,\bullet)$和具有阶n的$\alpha \in G$构成。

1. 对于$i, 0 \leq i \leq m-1$，U_i选择随机数$a_i, 0 \leq a_i \leq n-1$。计算$b_i = \alpha^{a_i}$，并把b_i传给U_{i+1}和U_{i-1}。
2. 对于i，$0 \leq i \leq m-1$，U_i计算
$$X_i = (b_{i+1}/b_{i-1}^{a_i})$$
接着U_i把X_i转发给其他$m-1$个用户。
3. 对于i，$0 \leq i \leq m-1$，X_i，U_i计算
$$Z = b_{i-1}^{a_i m} X_i^{m-1} X_{i+1}^{m-2} \ldots X_{i-2}^{1}$$
这样$Z = \alpha^{a_0 a_1 + a_1 a_2 + \ldots + a_{m-1} a_0}$就是由$U_0, \ldots U_{m-1}$共同计算出来的秘密会议密钥。

Burmester-Desmedt会议密钥协商方案

- Burmester-Desmedt会议密钥协商方案以两“轮”的方式进行。在第一轮中，所有的参与者向他们的两个邻居发送消息。第二轮中，每个参与者向其他所有的人广播一个消息。在每一轮中，所有的动作可以并行进行。
- 总体上，这种方案每执行一次，每个参与者需要发送两种消息，接受m-1种消息。这是高效的，但是需要广播网络的支持。
- Steiner，Tsudik和Waidner提出一种更自然的CKAS，它不需要广播网络。

Steiner-Tsudik-Waidner会议密钥协商方案

公开域参数由群$(G, \bullet)$和具有阶n的$\alpha \in G$构成。

阶段1

U_0选择随机数a_0，计算α^{a_0}，并传送$\mathcal{L}_0 = (\alpha^{a_0})$给$U_1$。

对于$i = 1, ..., m-2, U_i$从U_{i-1}接收$\mathcal{L}_{i-1}$。接着，U_i选择一个随机数a_i并计算

$$\alpha^{a_0 a_1 \ldots a_i} = (\alpha^{a_0 a_1 \ldots a_{i-1}})^{a_i}$$

接着他传送$\mathcal{L}_i = \mathcal{L}_{i-1} || \alpha^{a_0 a_1 \ldots a_i}$到$U_{i+1}$。

U_{m-1}从U_{m-2}接收到$\mathcal{L}_{m-2}$，接下来他选择一个随机数a_{m-1}，并计算：

$$\alpha^{a_0 a_1 \ldots a_{m-1}} = (\alpha^{a_0 a_1 \ldots a_{m-2}})^{a_{m-1}}$$

接着他构建列表$\mathcal{L}_{m-1} = \mathcal{L}_{m-2} || \alpha^{a_0 a_1 \ldots a_{m-1}}$

Steiner-Tsudik-Waidner会议密钥协商方案

阶段2

U_{m-1}从$\mathcal{L}_{m-1}$中解出会议密钥$Z = \alpha^{a_0 a_1 \ldots a_{m-1}}$。对于每一个元素$y \in \mathcal{L}_{m-1}$,$U_{m-1}$计算$y^{a_{m-1}}$。接着$U_{m-1}$构建$m-1$个元素的列表：

$$\mathcal{M}_{m-1} = (\alpha^{a_{m-1}}, \alpha^{a_0 a_{m-1}}, \ldots, \alpha^{a_0 a_1 a_{m-1}}, \ldots, \alpha^{a_0 a_1 \ldots a_{m-3} a_{m-1}})$$

并传送$\mathcal{M}_{m-1}$给U_{m-2}。

阶段2（续）

对于$i = m - 2, ..., 1$，U_i从U_{i+1}处接受到列表$\mathcal{M}_{i+1}$。从$\mathcal{M}_{i+1}$的最后一元素中计算会议密钥$Z = (\alpha^{a_0 \cdots a_{i-1} a_{i+1} \cdots a_{m-1}})^{a_i}$。对于任何其他的$y \in \mathcal{L}_{i+1}$，$U_i$计算$y^{a_i}$。接着$U_i$构建一个$i$个值的列表

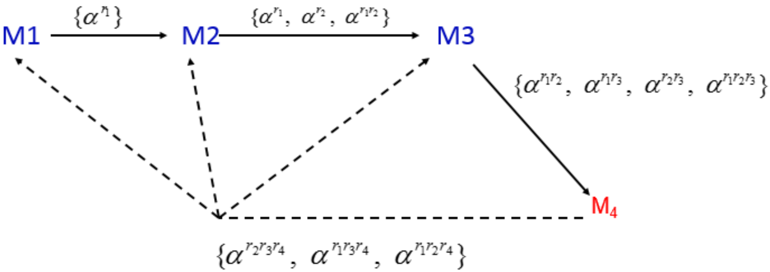
$$\mathcal{M}_i = (\alpha^{a_i \cdots a_{m-1}}, \alpha^{a_0 a_i \cdots a_{m-1}}, \alpha^{a_0 a_1 a_i \cdots a_{m-1}}, ..., \alpha^{a_0 a_1 \cdots a_{i-2} a_i \cdots a_{m-1}})$$

并把$\mathcal{M}_i$传给U_{i-1}。

U_0从U_1接收$\mathcal{M}_1$。他通过$\mathcal{M}_1$中的（唯一）元素计算会议密钥$Z = (\alpha^{a_1 \cdots a_{m-1}})^{a_0}$

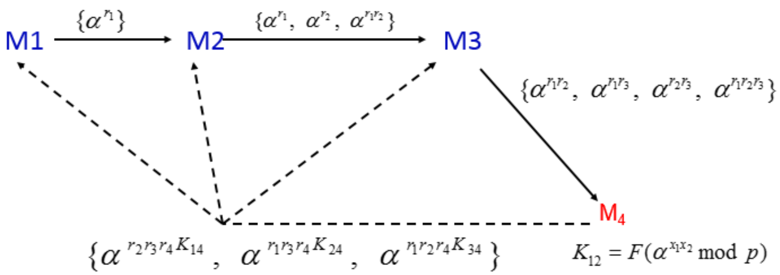
Group Diffie-Hellman(GDH)及其变型

GDH



$$S_4 = \alpha^{r_1r_2r_3r_4} \bmod p$$

A-GDH



$K_{12} = F(\alpha^{x_1 x_2} \bmod p)$

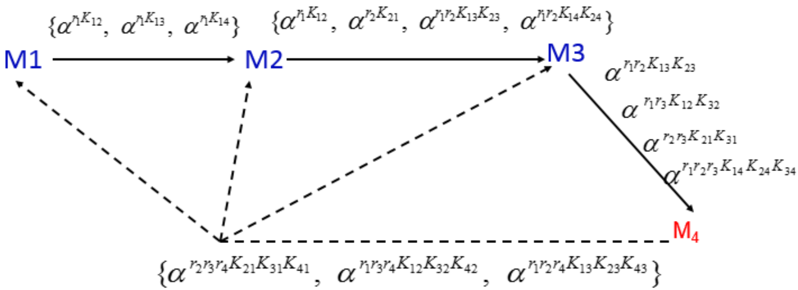
$K_{24} = F(\alpha^{x_2 x_4} \bmod p)$

$K_{34} = F(\alpha^{x_3 x_4} \bmod p)$

$$S_4 = \alpha^{r_1 r_2 r_3 r_4} \bmod p \quad (i.e.\ S_4(M_1) = \alpha^{r_2 r_3 r_4 K_{14}(r_1 K^{-1}_{14})} \bmod p)$$

Group Diffie-Hellman(GDH)及其变型

SA-GDH



$$K_{12} = K_{21} = F(\alpha^{x_1x_2} \bmod p)$$
$$K_{13} = K_{31} = F(\alpha^{x_1x_3} \bmod p)$$
$$K_{14} = K_{41} = F(\alpha^{x_1x_4} \bmod p)$$

$$K_{23} = K_{32} = F(\alpha^{x_2x_3} \bmod p)$$
$$K_{24} = K_{42} = F(\alpha^{x_2x_4} \bmod p)$$
$$K_{34} = K_{43} = F(\alpha^{x_3x_4} \bmod p)$$

$$S_4 = \alpha^{r_1r_2r_3r_4} \bmod p \quad (i.e.\ S_4(M_1) = \alpha^{r_2r_3r_4K_{21}K_{31}K_{41}(r_1K^{-1}{}_{12}K^{-1}{}_{13}K^{-1}{}_{14})} \bmod p)$$

SA-GDH

优势：

1. 可以抵抗已知密钥攻击
2. 具有完美的前向安全性

劣势：

1. 每个成员需要很多对密钥