PKC'04 Bresson (BR'2000和CDH, OMDHEKE, CCS03改进)
New Security Results on Encrypted key Exchange

优点: ① 基于CSS'03做了改进,仅用ROM和CDH证明AuthA协议的安全性,使用了full domain hash。加密原语采用了掩码生成函数(mask generation function) ② 采用非均匀分布的口令分布Dpw表示最可能的q个口令的集合. ③ 采用puzzle的方式,让客户端计算一个算力的证明给服务器,来防止非法客户端采用DDoS攻击耗尽服务器的计算资源.

问题: ① 采用puzzle的方式,是否会在实践中影响客户端实体的使用体验,因为客户端要计算很多,这一点认为文中应简要讨论. ② 协议采用的是双方共享口令(或变体),若被敌手获取可以直接冒充用户,无需做进一步猜测. ③ 写法可以改进一些: 例如:第2章开P未介绍含义;证明过程个人认为可再详细一些.

OMDHKE协议: One-Mask Diffie-Hellman key Exchange.
口令分布, D_{pw}. $D_{pw}(q) = \max_{P\subseteq Password}\{\Pr[pw\in P \mid \#P\le q]\}$

pw为口令, $PW = G(pw)$, 为full domain function值. $(0,1)^* \to G$群中

Client A
$x, X\leftarrow g^x$ A_1 —— $A, X^*\leftarrow X\times PW$ ——→ Server S
$X\leftarrow X^*/PW$. $y, Y\leftarrow g^y$ (S_1), $k_S\leftarrow X^y$ (S_2).
$k_A\leftarrow Y^x$ ←—— $S, Y, Auth_S$ (S_3) ——
$Auth_S \overset{?}{=} H_1(C\|\cdots\|k_A)$ (A_2)
accept.
$sk_A\leftarrow H_0(C\|\cdots\|k_A)$, terminate
$Auth_S\leftarrow H_1(C\|S\|X^*\|Y\|PW\|k_S)$
accept
$sk_S\leftarrow H_0(C\|\cdots\|k_S)$
terminate.

安全属性: 语义安全性,认证. 基于CDH完成证明

通过 $G_0\sim G_5$, 证明OMDEKE可抵抗字典攻击,保证语义安全性且保证认证正确用.(单向认证),通过Shoup定理限制事件概率.

定理为: $Adv^{ake}_{omdhke}(A)\le \frac{2q_s}{2^{l_0}} + 12\times D_{pw}(q_s) + 12q_h^2\times Succ^{cdh}_{g,G}(t+2T_e) + \frac{2Q^2}{q}$.

$Succ^{s\text{-}auth}_{omdhke}(A)\le \frac{q_s}{2^{l_0}} + 3\times D_{pw}(q_s) + 3q_h^2\times Succ^{cdh}_{g,G}(t+3T_e) + \frac{Q^2}{2q}$.

其中, $Q = q_p+q_s+q_g$.

每个Game中的事件event: ① S_n(语义安全性) $b=b'$时发生,敌手猜到sk
② A_n(认证), A_i已accept,但没有S_j对应.(同一会话)

Game G_0: random oracle model下的真实协议,其中有:
$Adv^{ake}_{omdhke}(A) = 2\Pr[S_0]-1$, $Succ^{s\text{-}auth}_{omdhke}(A) = \Pr[A_0]$.
下文主要关注 A_n 和约束事件: $SA_n = S_n\wedge\neg A_n$.

G_1: 仿真了所有的hash oracle, Send, Execute, Reveal和Test询问的实例,对于hash, G_1, H_0和H_1,以及G_3中的H_0', H_1',均保有了记录ΛH. $\Lambda H'$及ΛG.(不在在记录就选一个随机值).
Send的事件 $A_1\sim A_3$ $S_1\sim S_3$ 均正常执行.
易知 G_1和G_0不可区分.

G_2: 为后续分析,去除了两个不太可能发生的碰撞
1> $((CA, X^*), (S, Y))$在不同会话中碰撞,由于该记录至少有一个诚实的实体,故X^*或Y中至少一个为均匀分布.
2> G函数的输出的碰撞.
两者的概率可通过生日悖论限制: $\Pr[Coll_2]\le \frac{(q_p+q_s)^2}{2q} + \frac{q_g^2}{2q}$.

G_3: 通过仿真的private oracle H_0', H_1'来计算sk及Auth.
更改A_2/S_3: $Auth = H_1'(CA\|S\|X^*\|Y)$, $sk_A/s = H_0'(CA\|S\|X^*\|Y)$
由于无需k_A和k_S,更改A_2和S_2,为do nothing.
无需PW,可以简化 X^*的生成 $X^* = g^x$

▲除非以下事件发生, G_2和G_3是不可区分的.
① $AskH_{13}$: 即存询问 $H_1(C\|k_A)$或$H_1(C\|\cdots\|k_S)$. ← Auth
或$AskH_{13}$未发生,发生了$AskH_0|_3$. 即有实体accept了,存询问了 $H_0(C\|\cdots\|k_A)$或$H_0(C\|\cdots\|k_S)$. ← sk.

由于auth通过private oracle计算,故敌手无法猜测得到,除非只能每次试另外有会话也为$(CA, X^*), (S, Y)$,这已在G_2排除.
对于sk,敌手也无优势猜测,可得知, $\Pr[A_2]\le\frac{q_s}{2^{l_0}}$, $\Pr[SA_2] = \frac{1}{2}$.

$AskH_1$可分为3个子事件: ①被动窃听, X^*和Y均为仿真
② X^*仿真,而Y是敌手发出的 ③ Y仿真, X^*是敌手发出的

G_4: 引入随机DH实例(P, Q)其中P, Q均$\in G^*$, 为G生成元.
更改oracle G,引入Q,让 $Y = Q^{-k}\cdot\Lambda_G$中存(Q, Y, k).
对于S_1改为: $Y = P^y$,排除了$PW=1$和$Y=1$的小情况,概率不变.
▲ $|\Pr[AskH_4] - \Pr[AskH_3]|\le\frac{q_s+q_p}{q} + \frac{q_g}{q}$

G_5 $AskH$事件的可能性的评估.
首先排除碰撞$Coll_5$. 即对于G^3,存与A^2均有元组在Λ_H中, $(X^*, Y, PW, CDH_{g,G}(X^*/PW, Y))$. 通过Lemma2得概率.
若同时存在PW_0, PW_1,有(X^*, Y, PW_i, Z_i). 在Λ_H中,其中:
$Z_i = CDH_{g,G}(X^*/PW_i, Y) = CDH_{g,G}(X^y, Y)\times CDH_{g,G}(P^{w_i}, Q^{w_i})^{-k}$, 采用解CDH
▲ $\Pr[Coll_5]\le q_h^2\times Succ^{cdh}_{g,G}(t, T_e)$, $|\Pr[AskH_5] - \Pr[AskH_4]|\le\Pr[Coll_5]$
通过在上述AskH的3种小情况下分析:
① 若X^*, Y均为仿真, $X^* = g^x$, $Y = P^y$, $PW = Q^{-k}$,有: $Z = CDH_{g,G}(X^*, Y)\times CDH(Q, Y)^k = P^{xy}\times CDH_{g,G}(P, Q)^{yk}$,则求值必须解决CDH问题. $\Pr[AskH_1 - Passive_5]\le q_h\times Succ^{cdh}_{g,G}(t+2T_e)$.
② 存何A仿假,但敌手只能一次尝试一个pw.来发送auth,当不考虑G碰撞时,有: $\Pr[AskH_1 - With\,A_5]\le D_{pw}(q_s)$.
③ 同上分析。若以上3个事件未发生,则双方实体accept.
有: $\Pr[AskH\,Owl_5]\le D_{pw}(q_s) + q_h\times Succ^{cdh}_{q,G}(t+2T_e)$.
有: $\Pr[AskH_5]\le 3D_{pw}(q_s) + 2q_h\times Succ^{cdh}_{q,G}(t+2T_e)$.

防止DoS攻击的思想: puzzle.
Client A —A→ Server S ↗随机值与时期T的MAC.
←S, Ns, cookie— $MAC_{sks}(A,S,Ns)$→cookie
穷举y,使$(Ns\|y)=0^k$.
A, X*, Ns, y, cookie. 验证参数并锁定cookie
防止DoS攻击.