ICS 35.040
L 80

中华人民共和国国家标准

GB/T 32213—2015

信息安全技术　公钥基础设施 远程口令鉴别与密钥建立规范

Information security technology—Public key infrastructure—Specification for remote password authentication and key establishment

2015-12-10 发布　　2016-08-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

目　　次

前言 …… Ⅲ
引言 …… Ⅳ
1　范围 …… 1
2　规范性引用文件 …… 1
3　术语和定义 …… 1
4　缩略语 …… 2
5　符号 …… 2
6　文档约定 …… 3
6.1　括号 …… 3
6.2　步骤与顺序 …… 4
6.3　方法参数 …… 4
6.4　参与方 …… 4
7　数学定义 …… 4
7.1　群运算 …… 4
7.2　离散对数体制 …… 4
7.3　椭圆曲线体制 …… 5
8　模型 …… 5
8.1　概述 …… 5
8.2　原语 …… 6
8.3　协议 …… 6
8.4　密码函数 …… 7
9　原语 …… 7
9.1　概述 …… 7
9.2　数据类型转换原语 …… 7
9.3　连带口令公钥生成原语 …… 8
9.4　公钥生成原语 …… 10
9.5　口令验证数据生成原语 …… 11
9.6　随机元素导出原语 …… 12
9.7　秘密值导出原语 …… 14
9.8　密钥检索原语 …… 18
10　口令鉴别密钥建立协议 …… 19
10.1　BPKA-1 …… 19
10.2　BPKA-2 …… 20
10.3　BPKA-3 …… 22
10.4　APKA-1 …… 24

10.5 [DL]APKA-{2,3} …… 26
10.6 [EC]APKA-4 …… 27
10.7 APKA-5 …… 29
10.8 PKR-1 …… 31
11 密码函数 …… 32
11.1 散列函数 …… 32
11.2 掩码生成函数 …… 32
11.3 密钥证实函数 …… 33
11.4 乘法元生成函数 …… 33
11.5 密钥导出原语 …… 34
参考文献 …… 36

前　言

本标准按照 GB/T 1.1—2009 给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本标准由全国信息安全标准化技术委员会(SAC/TC 260)提出并归口。

本标准起草单位:中国科学院软件研究所、中国科学院研究生院、中国电子技术标准化研究院。

本标准主要起草人:张立武、冯登国、张振峰、高志刚、荆继武、张严、王鹏翮、李强、段美姣、高能、陈星。

引　言

目前，基于口令的实体鉴别技术是应用最广泛的鉴别技术，并且可以预见在未来的相当长时间内还将作为一种重要的鉴别技术存在。这一方面是因为口令容易记忆、不需要额外的载体，使用方便；另一方面基于口令的鉴别协议通常简单高效，适用于用户量巨大的信息系统。然而，由于口令一般由可打印的 ASCII 字符组成，选择空间较小，因此安全的基于口令的鉴别协议的设计和实现较为困难。更为不利的因素是用户通常会选择能方便记忆且易于使用的具有特定意义的单词或者词组作为口令，更容易遭受字典式攻击的影响。因此，在构建基于口令的鉴别系统时选择安全的口令鉴别协议变得尤为重要。

非对称密码学的发展为基于口令的身份鉴别和密钥建立协议的构造提供了一种新的方向。通过结合非对称密码学和口令可以构造更安全的口令鉴别密钥建立协议，并能提供抵抗离线蛮力攻击、抵抗字典式攻击、前向安全性等重要安全性质。本标准选取了数个经过广泛理论分析和应用验证的协议，定义了这些协议的数学基础、协议流程。本标准为基于口令鉴别系统的设计和开发提供了参考。

信息安全技术　公钥基础设施
远程口令鉴别与密钥建立规范

1　范围

本标准定义了基于非对称密码技术实现远程口令鉴别与密钥建立的数学定义和协议构造。

本标准适用于采用基于口令鉴别与密钥建立技术的鉴别系统的设计和开发。

2　规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本(包括所有的修改单)适用于本文件。

GB/T 18238.3—2002　信息技术　安全技术　散列函数　第3部分：专用散列函数

GB/T 25069—2010　信息安全技术　术语

3　术语和定义

GB/T 25069—2010界定的以及下列术语和定义适用于本文件。

3.1

口令穷举/蛮力攻击　password exhaustive attack/brute-force attack

通过尝试口令所有可能的值，以获取实际口令，并实施违反信息安全策略的行为。

3.2

口令破解　password crack

成功的穷举/蛮力攻击口令及口令相关秘密数据或者成功攻击一个基于口令的密码系统。

3.3

多重散列　iterated hash

重复多次使用散列函数对输入进行散列计算的方法。

3.4

低等级口令　low grade password

易于受口令穷举/蛮力攻击的口令。

3.5

连带口令公钥　password-entangled public key

从口令和私钥计算得出的公钥。

3.6

口令限制私钥　password-limited private key

从口令计算得到的私钥，该私钥的随机性完全来自于口令，并且其随机性受口令随机性的限制。

3.7

口令限制公钥　password-limited public key

从口令限制私钥生成的用于验证口令正确性的数据。

4 缩略语

下列缩略语适用于本文件。

APKA 增强型口令鉴别密钥协商(augmented password-authenticated key agreement)

BPKA 平衡型口令鉴别密钥协商(balanced password-authenticated key agreement)

CLIENT 客户端(client)

DL 离散对数(discrete logarithm)

EC 椭圆曲线(ellipse curve)

GE2FEP 群元素到域元素值转换原语(group element to field element conversion primitive)

GE2OSP 群元素到八位位组串转换原语(group element to octet string conversion primitive)

FE2OSP 域元素到八位位组串转换原语(field element to octet string conversion primitive)

I2FEP 整数到域元素转换原语(integer to field element conversion primitive)

I2OSP 整数到八位位组串转换原语(integer to octet string conversion primitive)

KCF 密钥证实函数(key confirmation function)

KDF 密钥导出函数(key derivation function)

KRBP 密钥检索盲化原语(key retrieval blinding primitive)

KRPP 密钥检索置换原语(key retrieval permutation primitive)

KRUP 密钥检索解盲化原语(key retrieval unblinding primitive)

MGF 掩码生成函数(mask generation function)

MVCF 多重值产生函数(multiplier value creation function)

OS2IP 八位位组串到整数转换原语(octet string to integer conversion primitive)

PEPKGP 连带口令公钥生成原语(password-entangled PKGP)

PKA 口令鉴别密钥协商(password-authenticated key agreement)

PKGP 公钥生成原语(public key generation primitive)

PKR 口令鉴别密钥检索(password-authenticated key retrieval)

PVDGP 口令验证数据生成原语(password verification data generation primitive)

REDP 随机元素导出原语(random element derivation primitive)

SERVER 服务器(server)

SRP 安全远程口令(secure remote password)

SVDP 秘密值导出原语(secret value derivation primitive)

5 符号

下列符号适用于本文件。

a,b 有限域 $GF(q)$的元素,是定义椭圆曲线 E 的两个系数。

E 由 a,b 定义的,定义在有限域 $GF(q)$上的椭圆曲线。

e,e_1,e_2 群或有限域上的元素。

error 出错。在本标准中如果某个原语或者函数输出 error,表示该原语或者函数执行过程中出错,不能得到预期结果。

exp 指数运算,$\exp(g,a)=g^{\wedge}a$。

$GF(q)$ 一个阶为 q 的有限域。

G 椭圆曲线 E 上阶为 r 的点,该点可以生成一个阶为 r 的子群。

g　有限域$GF(q)$上的阶为r的乘法子群的一个生成元，用来定义离散对数域参数。

hash　散列函数。

hex　十六进制编码函数，把输入的八位位组串编码成十六进制串。

K　协议生成的由两个参与方共享的密钥。

k　在离散对数体制中等于$(q-1)/r$，用来定义离散对数域参数；在椭圆曲线体制中等于$\sharp E/r$，用来定义椭圆曲线域参数，其中$\sharp E$为E上点的数量。

log　对数运算符。

M　消息，一般为八位位组串。

mod　求余操作。

N　自然数，非负整数。

P，Q　一般指代椭圆曲线上的点。

q　正整数，一般用来表示有限域的阶。

r　在离散对数体制中为$q-1$的素因子，用来定义离散对数域参数；在椭圆曲线体制中为$\sharp E$的素因子，用来定义椭圆曲线域参数。

s，u，s'，u'　某一实体的私钥组，一般是正整数。

w，v，w'，v'　某一实体的公钥组，一般为$GF(q)$上的元素。

valid，invalid　有效，无效。在本标准中表示输出结果是否是有效的，如果协议的输出是invalid，则表示协议执行出错，不能继续执行后续操作。

x_e　椭圆曲线上的点e对应的横坐标。

y_e　椭圆曲线上的点e对应的纵坐标。

z，z_1，z_2　共享秘密值，为$GF(q)$上的元素，由秘密值生成原语产生。

(c,d)　数字签名对，一对整数，通过一个签名原语产生。

(s,w)，(u,v)　椭圆曲线密钥对，其中s和u是私钥，w和v是公钥。

$+$　加法操作符。例如$P+Q$表示椭圆曲线上的点P和点Q相加。

$\times$　标量乘法操作符，表示一个域元素P和一个整数n的标量乘法。例如$n\times P$，也可以省略$\times$，记作nP。

/　除法操作符。

$*$　群或者域中的乘法操作符。例如$e_1 * e_2$表示在群元素e_1和e_2间应用乘法运算。

^　指数运算符，表示对$*$操作符的迭代。例如$e\hat{}n$表示对群元素e进行n次迭代操作$e*e*\cdots*e$，其中n必须为正整数，也可省略^，记做e^n。

$=$　赋值运算符。$a=b$表示把b赋值给a。

||　八位位组串的连接操作符。

Σ　求和操作符。

$>>$　比特向右移位操作符。

$[x,y]$　大于或等于x且小于或等于y的整数的集合。

$\lceil x \rceil$　大于或者等于x的最小整数。

$\lfloor x \rfloor$　小于或者等于x的最大整数。

6　文档约定

6.1　括号

包括以下两种：

a) {} 描述了两个或者多个相关的但运行在不同的上下文环境中的方法，如下所示：
   1) {DL，EC}用来描述使用离散对数和椭圆曲线的方法；
   2) {CLIENT，SERVER}用来描述在协议中两个参与方客户端/服务器使用的成对方法；
   3) {DL，EC} PKA-1-{CLIENT，SERVER} 相当于{DLPKA-1-CLIENT，DLPKA-1-SERVER，ECPKA-1-CLIENT，ECPKA-1-SERVER}。
b) [] 使用方括号限定的方法表示仅使用于该情况。例如[DL]表示仅适用于离散对数体制的方法。

6.2 步骤与顺序

本标准中所描述的包含有序的步骤序列的方法，一般可以在保证结果相同的情况下以任意的顺序执行。

但包含子步骤的步骤，所有的子步骤必须在父步骤完成之前完成。

“输出信息并停止”步骤用来限制敏感信息的泄露。这些步骤必须按照本标准指出的顺序执行并在输出信息后结束方法的执行。

6.3 方法参数

本标准中的协议、原语通过选择输入域参数，函数和相关的值进行初始化。这些参数通常认为是协议、原语的选项。因此，两个参与方要使用一个方法成功交互，必须使用相同的选项或者参数。

6.4 参与方

本标准中定义的远程口令鉴别密钥建立协议采用客户端和服务器指代参与协议运行的两个实体，其中客户端指代协议的发起者，服务器指代协议的响应者。客户端和服务器仅是从逻辑上区分协议参与方。一个实体既可以作为客户端参与协议，也可以作为服务器参与协议。

7 数学定义

7.1 群运算

在本标准中，采用小写字母表示群/有限域中的元素。操作符“*”和“^”被用来描述离散对数密码/椭圆曲线密码体制中的乘法和指数运算。

注：本章中的群运算、离散对数体制和椭圆曲线体制定义是参照 IEEE 1363-2000 中第 5 章和 IEEE 1363a-2004 中第 5 章定义，详见参考文献[9,10]。

7.2 离散对数体制

7.2.1 离散对数域参数

每个离散对数原语和协议都需要离散对数域参数，离散对数域参数也是每一个离散对数密钥隐含的组成部分。离散对数域应指定以下参数：

a) 一个有限域 $GF(q)$，其中 q 可以取值为一个正的奇素数 p、2^m（m 为正整数）、p^m（p 为奇素数，m 为正整数且 $m \geqslant 2$）；
b) 一个整除 $q-1$ 的正素数 r；
c) 有限域 GF(q)的一个 r 阶子群的生成元 g。

若 $q=2^m$，则应同时指定以下参数：

a) 有限域 $GF(q)$上元素的记法，用于 9.2 数据类型转换原语中；

b） 余因子 $k=(q-1)/r$。

7.2.2 离散对数密钥对

对于给定的离散对数域参数，一对离散对数密钥对包含一个离散对数私钥 s 和一个离散对数公钥 w，其中 s 为[1，$r-1$]之间的整数，w 为有限域 GF（q）上的元素且 $w=g^{s}$。为保证离散对数私钥的安全性，其生成应当是不可预测的，其存储应当无法被攻击者获取。

离散对数密钥对与生成它的域参数密切相关，而且只能用于该域参数的上下文中。一个密钥对不应当用于除生成其的域参数之外的域参数环境中。一组域参数可能被多个密钥对共享。

离散对数密钥对不一定由使用它的参与方生成，这取决于信任模型。

密钥管理中参与方建立的离散对数密钥可能符合密钥的一般形式，但却并不满足密钥的定义，这取决于所采用的密钥管理技术。因此，本标准中“离散对数公钥”及“离散对数私钥”表示符合一般形式的密钥，而“有效离散对数公钥”及“有效离散对数私钥”则表示满足定义的密钥。密钥确认是确定一个密钥是否为“有效”的过程。

7.3 椭圆曲线体制

7.3.1 椭圆曲线域参数

每个椭圆曲线原语和协议都需要椭圆曲线域参数，椭圆曲线域参数是每一个椭圆曲线密钥隐含的组成部分。椭圆曲线域应指定以下参数：

a） 一个有限域 GF（q），其中 q 是一个正奇素数 p，或者 2^{m}（m 为正整数）；

b） 两个有限域 GF（q）中的元素 a 和 b，它们是定义椭圆曲线 E 的系数；

c） 一个能整除椭圆曲线 E 上点数量的正素数 r；

d） 一个阶为 r 的点 G，该点也是它所生成的子群的生成元。

若 $q=2^{m}$，则应同时指定以下参数：

有限域 GF（q）上元素的记法，用于 9.2 数据类型转换原语中。

同时，上述参数隐含定义了余因子 $k=\#E/r$，其中 $\#E$ 表示椭圆曲线上的点的个数。

7.3.2 椭圆曲线密钥对

对于给定的椭圆曲线域参数，一对椭圆曲线密钥对包含一个椭圆曲线私钥 s 和一个椭圆曲线公钥 W，其中 s 为[1，$r-1$]之间的整数，W 为椭圆曲线 E 上的点，且 $W=sG$。为保证椭圆曲线私钥的安全性，其生成应当是不可预测的，其存储应当不能被攻击者获取。

椭圆曲线密钥对与生成它的域参数密切相关，而且只能用于该域参数的上下文中。一个密钥对不应当用于其被生成的域参数之外的域参数环境中。一组域参数可能被多个密钥对共享。

椭圆曲线密钥对不一定由使用它的参与方生成，这取决于信任模型。

参与方建立椭圆曲线密钥对是密钥管理的一部分内容。可能会存在参与方建立的椭圆曲线密钥对符合密钥的一般形式，但却并不满足所希望的密钥定义，这取决于所采用的密钥管理技术。因此，本标准中“椭圆曲线公钥”及“椭圆曲线私钥”表示符合一般形式的密钥，而“有效椭圆曲线公钥”及“有效椭圆曲线私钥”则表示满足定义的密钥。密钥确认是确定一个密钥是否为“有效”的过程。

8 模型

8.1 概述

本标准的目标在于为采用基于口令身份鉴别技术的鉴别系统提供安全的基于非对称密码技术的口

令鉴别与密钥建立协议。因此，在一个通用的、允许不同安全需求和不同应用需求的鉴别系统选择合适的协议的框架下描述这些安全协议是有意义的。本标准中的安全协议可以抽象地理解为由原语、协议和密码函数构成。

——原语是具有原子性的方法，提供确定的功能，是构建协议的基础。部分原语具有基于数论上的困难问题的安全性。

——协议是由一系列原语、密码函数构成的集合，提供基于复杂性理论的安全性。

——密码函数是提供特定密码学功能的函数，包括密钥产生函数、散列函数、密钥证实函数等。

口令鉴别与密钥建立协议和一般鉴别与密钥建立协议的主要不同在于证明参与方身份的秘密是一个或者多个预先分发的低等级口令。当使用口令进行相互鉴别时，口令通常被当作共享的秘密，而不是静态的公钥和私钥。

协议包括部分密钥管理操作，例如生成一个私钥或者获取其他参与方的公钥。参与方应该保证密钥和域参数的有效性，也应该保证密钥和域参数生成过程的正确性。因此参与方应采用适当的密钥管理方案，但密钥管理方案超出了本标准的范围。

协议的规范描述由以下几部分组成：

a) 协议选项：协议中使用到的原语和密码函数等；

b) 操作：协议按照一定顺序执行的步骤。

从实现的观点看，这里定义的原语可以看作是底层的实现，密码函数可以看作中层实现，协议可以看作高层或者称为应用层的实现。

8.2 原语

本标准中原语并不提供安全上的保证，只有当使用这些原语和其他操作构成协议时才能提供基于计算复杂性理论的安全性。

原语假设它们的输入符合某些假设，这些假设会在描述原语时列出。只要输入不影响原语实现的进一步操作，原语的实现不强制约束输入必须符合假设。在原语的实现中，出现错误时可以自主决定是否输出错误信息。例如，一个公钥生成原语实现当输入不符合假设时，可以输出一个看起来像是公钥的输出，也可以拒绝输出。这需要原语的使用者来验证输入是否符合前提假设。例如，用户可以选择使用相关的密钥和域参数验证技术。

原语规范是功能性规范，而不是接口规范。输入、输出的格式和实现超出了本标准的范围。

8.3 协议

8.3.1 口令鉴别密钥协商

在口令鉴别密钥协商协议中，参与方使用口令(口令相关数据)和私钥与另一参与方的公钥一起产生一个或者多个会话密钥。如果参与方使用相关的口令、密钥和一致的域参数，并且协议正确地完成，则所有参与方能得到相同的会话密钥。

同时，口令鉴别与密钥协商协议定义了密钥证实操作，协议参与方互相确认对方已经正确获得会话密钥。

口令鉴别密钥协商协议提供以下安全属性：

——通过协议可以成功的产生一个共享会话密钥，并且不知道口令或者基于口令秘密的实体无法成功执行协议；

——获得口令相关的公钥和协议中交换的任何信息不能使第三方可以验证对于口令或者口令相关值的猜测或者确定协议产生的共享会话密钥；

——不知道口令或者口令相关值的参与方在一次协议执行中最多只有一次机会猜测口令或者口令相关的值。

本标准中的口令鉴别密钥协商协议具有以下形式的操作序列：

a) 建立有效的域参数；

b) 根据域参数建立一个或者多个口令及口令相关值；

c) 根据域参数为参与方选择一个有效的私钥；

d) 使用 PEPKGP 或者 PKGP 操作根据步骤 c)产生的私钥建立一个或者多个公钥；

e) 分发公钥给协议的其他参与方；

f) 获取其他参与方的公钥；

g) 选择合适的方法验证域参数和其他参与方的公钥，如果无效则输出“invalid”并停止操作；

h) 根据自己的私钥和其他参与方的公钥产生一个共享秘密值；

i) 使用密钥产生函数根据步骤 h)产生的秘密值和域参数产生会话密钥。

8.3.2 口令鉴别密钥检索

在口令鉴别密钥检索协议中，拥有口令的客户端与至少一个拥有与口令相关的秘密值的服务器交互，最终建立一个或多个长期密钥。

为了成功的执行口令鉴别密钥检索协议，客户端仅需要拥有口令值 π，服务器也仅需要拥有与 π 相关联的秘密值，而不需要知道客户端的口令值 π 或者客户端最终获取的长期密钥。客户端通过与服务器交互并执行密钥检索操作来检索与口令相关联的密钥。

口令鉴别密钥检索与口令鉴别密钥协商主要有两个不同点：

a) 口令鉴别密钥检索建立长期密钥，而口令鉴别密钥协商建立的是临时会话密钥；

b) 口令鉴别密钥检索为客户端建立的长期密钥对于服务器可能是无法获取的，而口令鉴别密钥协商所建立的临时密钥对于客户端和服务器是共享的。

口令鉴别密钥检索和口令鉴别密钥协商同样可以在多服务器的系统中使用，客户端使用口令获取存储在各个服务器的密钥片段并最终检索出密钥。

8.4 密码函数

本标准定义了适用于口令鉴别与密钥建立协议的散列函数、掩码生成函数、密钥证实函数、乘法元生成函数和密钥导出函数。若在协议中采用本标准之外的函数，用户应对实现的安全性进行分析和评估。

9 原语

9.1 概述

本标准中定义了数据类型转换原语、连带口令公钥生成原语、公钥生成原语、口令验证数据生成原语、随机元素导出原语、秘密值导出原语和密钥检索原语。原语采用通用的方式进行描述，除非特别指明，忽略原语所依赖的密码体制。

9.2 数据类型转换原语

9.2.1 OS2IP

对于给定的长度为 k 的八位位组串 M，通过以下步骤或者相当的过程转换成整数 i：

a) 把 M 按字节分解为 $M_{k-1}||M_{k-2}||\cdots||M_0$，其中 M_{k-1} 是最左侧的字节，M_0 是最右侧字节；

b) 对于所有的 $j \in [0,k-1]$，计算 $i=\sum 2^{8j}M_j$；

c) 输出 i。

9.2.2 I2OSP

对于给定的非负整数 i 和目标八位位组串长度 l，应通过以下步骤或者相当的过程转换成八位位组串 M：

a) 如果 $i>256^{l}$，则输出“整数过大”；

b) 对于所有的 $j\in[0,l-1]$，计算 $m_j=i \bmod (256^{j})>>(8j)$；

c) 设 M_j 是 m_j 对应的字节表示，则 $M=M_{l-1}||M_{l-2}||\cdots||M_0$；

d) 输出 M。

9.2.3 I2FEP

对于给定的整数 $i\in[0,q-1]$，应通过以下步骤或者相当的过程转换成有限域 $GF(q)$ 中的元素 j：

a) 应用 I2OSP 转换 i 到$\lceil\log_{256}q\rceil$长度的八位位组串；

b) 应用 OS2FEP 转换结果八位位组串到域元素 j；

c) 输出 j。

9.2.4 GE2FEP

对于给定的群元素 e：

——在离散对数体制中，群元素就是域元素，直接输出域元素 e；

——在椭圆曲线体制中：

a) 如果 e 是无穷远点，则输出域元素“0”并停止；

b) 否则设置域元素 $x_e=e$ 的 x 坐标；

c) 输出域元素 x_e。

9.2.5 FE2OSP

对于给定的有限域 $GF(q)$ 中的元素 e：

——如果 q 为奇素数，则 e 是区间$[1,q-1]$中的整数，则应用 I2OSP 转换 e 到八位位组串；

——如果 q 为 2^{m} 的形式，则 e 即是长度为$\lceil t/8\rceil$的八位位组串，其中 $t=\lceil\log_2 q\rceil$。

9.2.6 OS2FEP

对于给定的长度为 t 的八位位组串 m：

——如果 q 为奇素数，应用 OS2IP 转换 m 为整数 i。若 $i\in[1,q-1]$，则输出 i；否则输出“error”；

——如果 q 为 2^{m} 的形式，则把 m 表示成长度为 $8t$ 的比特串 i。若 $8t\leqslant m$，则输出 i；否则输出“error”。

9.2.7 GE2OSP

对于给定的有限域 $GF(q)$ 中的元素 e：

——在离散对数体制中，群元素即是对应域中的域元素，因此输出 FE2OSP(e)；

——在椭圆曲线体制中：

a) 计算 e_x=GE2FEP(e)；

b) 输出 FE2OSP(e_x)。

9.3 连带口令公钥生成原语

9.3.1 PEPKGP-1

{DL,EC}PEPKGP-1 原语使用{DL,EC}域参数、参与方私钥和口令生成一个连带口令公钥。

输入：

a) 参与方的私钥(整数)s；

b) 基于口令生成的群元素 π_m；

c) 与密钥 s 和群元素 π_m 相关的域参数(包括 g，r)。

假设：私钥 s 和域参数是有效的，群元素 π_m 可以生成 r 阶群。

输出：连带口令公钥 w。

操作：

a) 计算一个群元素 $w=(g\hat{}s)*\pi_m$；

b) 输出 w 作为连带口令公钥。

9.3.2 PEPKGP-2

{DL，EC}PEPKGP-2 原语使用{DL，EC}域参数、参与方私钥和口令生成一个连带口令公钥。

输入：

a) 参与方的私钥(整数)s；

b) 基于口令生成的群生成元 g_π；

c) 与密钥 s 和群元素 g_π 相关的{DL，EC}域参数。

假设：私钥 s 和域参数是有效的，群元素 g_π 可以生成 r 阶群。

输出：连带口令公钥 w。

操作：

a) 计算一个群元素 $w=g_\pi\hat{}s$；

b) 输出 w 作为连带口令公钥。

9.3.3 [DL]PEPKGP-{3，4}-SERVER

[DL]PEPKGP-{3，4}-SERVER 原语使用 DL 域参数、服务器私钥和口令验证数据生成一个连带口令公钥。PEPKGP-4-SERVER 需要一个乘法元生成函数，可选项为 MVCF-1。客户端应该使用和服务器相同的 MVCF 和相关的参数。

输入：

a) 服务器的私钥(整数)s；

b) 口令验证数据 v_π，其中 v_π 是口令限制公钥；

c) 与密钥 s 和验证数据 v_π 相关的 DL 域参数(包括 g_{q-1} 和 q)。

假设：

私钥 s、验证数据 v_π 和域参数是有效的，g_{q-1} 的阶为 $q-1$。

输出：

连带口令公钥 w_s，其中 w_s 是域 $GF(q)$ 的元素。

操作：

a) 计算域元素 $x=\exp(g_{q-1},s)$；

b) 如下方法计算域元素 m_v：

1) PEPKGP-3：$m_v=1$；

2) PEPKGP-4：$m_v=$MVCF()；

c) 计算 $w_s=(v_\pi\times m_v+x)$；

d) 输出 w_s。

9.3.4 [EC]PEPKGP-5-SERVER

[EC]PEPKGP-5-SERVER 原语使用 EC 域参数、服务器私钥和口令验证数据生成连带口令公钥，

该原语在[EC]APKA-4-SERVER 中使用。

该原语需要一个随机元素生成函数作为参数，该参数的可选函数为[EC]REDP-1 和[EC]REDP-2，并应该保证该函数与客户端使用的随机元素生成函数相同。

输入：

a) 服务器的私钥(整数)s；

b) 口令验证数据 v_π，其中 v_π 是口令限制公钥；

c) 与密钥 s 和公钥 v_π 相关的 EC 域参数(包括 g)。

假设：

私钥 s、公钥 v_π 和域参数是有效的。

输出：

连带口令公钥 w_s。

操作：

a) 计算八位位组串 o_1=GE2OSP(v_π)；

b) 计算群元素 e_1=REDP(o_1)；

c) 计算群元素 $w_s=(g^{\wedge}s)*e_1$；

d) 输出 w_s。

9.4 公钥生成原语

9.4.1 PKGP-1

{DL,EC}PKGP-1 使用私钥和域参数来生成公钥。

输入：

a) 私钥(整数)s；

b) 私钥 s 相关的域参数(包括 g 和 r)。

假设：

私钥和相应的域参数有效。

输出：

与私钥对应的公钥 w。

操作：公钥 w 的计算应执行以下两步：

a) 计算公钥组元素 $w=g^{\wedge}s$；

b) 输出 w 作为公钥。

9.4.2 [DL]PKGP-2-CLIENT

[DL]PKGP-2-CLIENT 是离散对数公钥生成原语，该原语根据 DL 域参数和客户端的私钥生成客户端的公钥，该原语应用在[DL]APKA-{2,3}-CLIENT。

输入：

a) 客户端的私钥(整数)s；

b) 与密钥 s 相关的 DL 域参数(包括 g_{q-1} 和 q)。

假设：

私钥 s、验证数据 v_π 和域参数是有效的，g_{q-1} 的阶为 $q-1$。

输出：

公钥 w_c，其中 w_c 是 $GF(q)$ 中的非零元素。

操作：

a) 计算域元素 w_c=exp(g_{q-1},s)；

b) 输出 w_c。

9.5 口令验证数据生成原语

9.5.1 PVDGP-1

{DL,EC}PVDGP-1 原语根据域参数和参与方的口令生成口令验证数据，该原语在{DL,EC}APKA-1-CLIENT 和 {DL,EC}APKA-5-CLIENT 中用来生成口令限制私钥，在{DL,EC}APKA-1-SERVER 和 {DL,EC}APKA-5-SERVER 中用来生成口令验证数据。

该原语需要一个随机元素生成函数和一个散列函数作为参数，其中随机元素生成函数的可选项为{DL,EC}REDP-1 和{DL,EC}REDP-2，散列函数的选择项为 11.1 中定义的散列函数和 MGF1。

输入：

a) 八位位组串口令 π；

b) {DL,EC}域参数。

输出：

口令限制私钥 u_π 和口令验证数据，口令验证数据包括阶为 r 的生成元 g_π 和口令限制公钥 v_π。

操作：

a) 计算八位位组串 $o_\pi = \mathrm{hash}(\pi)$；

b) 计算私钥(整数) $u_\pi = \mathrm{OS2IP}(o_\pi) \mod r$；

c) 计算元素 $g_\pi = \mathrm{REDP}(o_\pi)$；

d) 计算群元素 $v_\pi = g_\pi \hat{} u_\pi$；

e) 输出 u_π 和口令验证数据 (g_π, v_π)。

9.5.2 [DL]PVDGP-2

[DL]PVDGP-2 原语根据 DL 域参数和参与方的口令生成口令验证数据和其他相关值，该原语在[DL]APKA-{2,3}-CLIENT 中用来生产口令限制私钥，在[DL]APKA-{2,3}-SERVER 中用来生成口令验证数据。

[DL]PVDGP-3 需要一个散列函数作为参数，散列函数的选择项为 11.1 中定义的散列函数和 MGF1。

输入：

a) 八位位组串口令 π；

b) 与口令相关的 DL 域参数(包括 g_{q-1} 和 q)。

输出：

口令限制私钥 u_π 和口令验证数据，口令验证数据包括口令限制公钥 v_π。

假设：

口令 π 和 DL 域参数是有效的，g_{q-1} 的阶为 $q-1$ 。

操作：

a) 计算八位位组串 $o_\pi = \mathrm{hash}(\pi)$；

b) 计算私钥 $u_\pi = \mathrm{OS2IP}(o_\pi) \mod (q-1)$；

c) 计算口令限制公钥 $v_\pi = \exp(g_{q-1}, u_\pi)$；

d) 输出 u_π 和口令验证数据 v_π。

9.5.3 [EC]PVDGP-3

[EC]PVDGP-3 原语根据域参数和参与方的口令生成口令验证数据，该原语在[EC]APKA-4-CLIENT 中用来生产口令限制私钥，在[EC]APKA-4-SERVER 中用来生成口令验证数据。

该原语需要一个散列函数作为参数，散列函数的选择项为 11.1 中定义的散列函数和 MGF1。

输入：

a) 八位位组串口令 π；

b) 与口令相关的椭圆曲线域参数(包括 g 和 r)。

输出：

口令限制私钥 u_π 和口令验证数据，口令验证数据 v_π。

操作：

a) 计算八位位组串 $o_\pi = \mathrm{hash}(\pi)$；

b) 计算私钥 $u_\pi = \mathrm{OS2IP}(o_\pi) \bmod r$；

c) 计算群元素 $v_\pi = g \hat{} u_\pi$；

d) 输出 u_π 和口令验证数据 v_π。

9.6 随机元素导出原语

9.6.1 [DL]REDP-1

[DL]REDP-1 是离散对数随机元素生成原语，该原语采用一个散列函数根据一个口令输入选取一个伪随机群元素。该原语可以用在[DL]BPKA-3，[DL]APKA-1，[DL]APKA-5 中产生生成元元素，也可以用在[DL]BPKA-1，[DL]BPKA-2 生成一个掩码元素。

该原语需要一个散列函数作为参数，其中散列函数的选择项为 11.1 中定义的散列函数和 MGF1。

输入：

a) 基于口令生成的八位位组串 o_π；

b) DL 域参数(包括 q 和 k)。

输出：选择的 r 阶的群元素 e。

操作：

a) 计算八位位组串 $o_1 = \mathrm{hash}(o_\pi)$；

b) 计算域元素 $x = \mathrm{I2FEP}(\mathrm{OS2IP}(o_1) \bmod q)$；如果 $x = 0$，则输出“invalid”并终止；

c) 计算域元素 $e = \exp(x, \mathrm{k})$；如果 $e = 1$，则输出“invalid”并终止；

d) 输出 e。

9.6.2 [EC]REDP-1

[EC]REDP-1 原语基于[DL]REDP1，该原语采用一个散列函数根据一个口令输入选取一个伪随机群元素，该元素属于椭圆曲线口令鉴别密钥协商方案所选择的群。该原语可以用在[EC]BPKA-3，[EC]APKA-1，[EC]APKA-5 中产生生成元元素，也可以用在[EC]BPKA-11，[EC]BPKA-2 生成一个掩码元素。

该原语需要一个输出长度为 $oLen$ 的散列函数作为参数，其中散列的选择项为 11.1 中定义的散列函数和 MGF1。

输入：

a) 基于口令生成的八位位组串 o_π；

b) 与 o_π 有关的椭圆曲线域参数(包括 q，p，m，a，b 和 k)。

输出：选择的有效的 r 阶的群元素 e，或者“invalid”。

操作：

a) 计算 $o_1 = \mathrm{hash}(o_\pi)$；

b) 计算 $i_1 = \mathrm{OS2IP}(o_1)$；

c) 计算 $oLen$ 长度的八位位组串 o_2=I2OSP(i_1)；

d) 计算 o_3=hash(o_2)；

e) 计算一个域元素 x=I2FEP(OS2IP(o_3)mod q)；如果 x=0，输出“invalid”并终止；

f) 计算 $\mu=i_1 \bmod 2$；

g) 如果 q 是偶数，则跳转到第 i)步；

h) 如果 q 是奇数则：

1) 设置 α：如果($p>3$)，则设置 $\alpha=\exp(x,3)+a\times x+b$；如果($p=3$)，则设置 $\alpha=\exp(x,3)+a\times\exp(x,2)+b$；

2) 如果 $\alpha=0$，输出“invalid”并终止；

3) 如果 α 存在平方根，则计算 α 的一个平方根 β；

4) 如果 α 不存在平方根，则跳转到第 j)步；

5) 设置 $y=\exp(p-1,\mu)\times\beta$；

6) 设置点 $T_1=(x,y)$ 并跳转到第 k)步；

i) 如果 q 是偶数则：

1) 设置 $b=x+a+b\times\exp(x,-2)$；

2) 找到一个域元素 z，使得 $\exp(z,2)+z=b$；

3) 如果不存在这样的 z，则表明没有这样的结果，跳转到 j)继续流程；

4) 设置 $y=(z+\mu)\times x$，其中 m 是{0,1}表示了域元素{0,1}；

5) 设置点 $T_1=(x,y)$ 并跳转到 k)步；

j) 如果椭圆曲线上在 x 上不存在点，则：

1) 设置 $i_1=i_1+1$；

2) 跳转到 c)步；

k) 计算群元素 $e=k\times T_1$；如果 e 是曲线上的无穷远点，则输出“invalid”并停止；

l) 输出 e。

9.6.3 REDP-2

REDP-2 原语采用一个散列函数根据口令选取一个伪随机群元素。该原语可以用在{DL,EC}BPKA-3，{DL,EC}APKA-1，{DL,EC}APKA-5 中产生生成元元素，也可以用在{DL,EC}BPKA-1，{DL,EC}BPKA-2 生成一个掩码元素。

该原语需要一个散列函数和两个随机群元素 g_a 和 g_b 作为参数，其中散列的选择项为 11.1 中定义的散列函数和 MGF1，g_a 和 g_b 是由 g 生成的群中的 r 阶元素。

输入：

a) 基于口令生成的八位位组串 o_π；

b) g_a，g_b，o_π 相关的域参数(包括 g 和 r)。

假设：假设域参数是有效的，并且 g_a 和 g_b 在群上符合均匀分布，因此没有实体能获得 g_a 和 g_b 之间的指数关系，也没有实体可以获得 g_b 和 g 之间的指数关系。

输出：选择的 r 阶的群元素 e。

操作：

a) 计算八位位组串 o_1=hash(o_π)；

b) 计算证书 i_2=OS2IP(o_1) mod r；如果 i_2=0，则输出“invalid”并终止；

c) 计算域元素 $e=g_a*(g_b\hat{}i_2)$；如果 $e=1$，则输出“invalid”并终止；

d) 输出 e。

9.7 秘密值导出原语

9.7.1 SVDP-1-CLIENT

{DL,EC}SVDP-1-CLIENT 原语适用于客户端。该原语使用客户端的私钥和服务器的公钥生成共享的秘密值。

输入：

a) 客户端私钥 s；

b) 服务器公钥 w_s；

c) 与密钥 s 和 w_s 相关的{DL,EC}域参数(包括 q，r 和 g)。

假设：私钥 s 和域参数是有效的。

输出：一个属于有限域 $GF(q)$ 中的元素 z。

操作：

a) 计算 $z_g = w_s \hat{} s$；

b) 计算 $z = \text{GE2FEP}(z_g)$；

c) 输出 z。

9.7.2 SVDP-2

{DL,EC}SVDP-2 原语使用参与方的私钥、口令相关值和另一参与方的连带口令公钥生成共享的秘密值。

输入：

a) 参与方的私钥 s；

b) 基于口令的掩码群元素 π_m；

c) 另一参与方的连带口令公钥 w'；

d) 与密钥 s，w' 和 π_m 相关的{DL,EC}域参数(包括 q，r)。

假设：私钥 s 和域参数是有效的，w' 属于一个父群，π_m 是目标群的一个 r 阶生成元。

输出：一个属于有限域 $GF(q)$ 中的元素 z。

操作：

a) 计算 $z_g = (w' * \pi_m \hat{} (-1)) \hat{} s$；

b) 计算 $z = \text{GE2FEP}(z_g)$；

c) 输出 z。

9.7.3 SVDP-3

{DL,EC}SVDP-3 原语使用参与方的私钥、口令相关值和另一参与方的连带口令公钥生成共享的秘密值。

输入：

a) 参与方的私钥 s；

b) 另一参与方的连带口令公钥 w'；

c) 与密钥 s，w' 和 π_m 相关的{DL,EC}域参数(包括 q)；

d) 一个布尔值 b_1 标识是否使用乘法因子。

假设：私钥 s 和域参数是有效的，w' 属于一个父群，s 和 w' 使用相同的域参数生成，所有的参与方协商确定是否使用乘法因子。

输出：一个属于有限域 $GF(q)$ 中的元素 z。

操作：

a) 如果 b_1 表示不使用乘法因子：计算群元素 $z_g=w'^{\wedge}s$；

b) 如果 b_1 表示使用乘法因子：计算 $z_g=w'^{\wedge}(ks)$；

c) 计算 $z=\mathrm{GE2FEP}(z_g)$；

d) 输出 z。

9.7.4 [DL]SVDP-{4,5}-CLIENT

[DL]SVDP-{4,5}-CLIENT 原语适用于客户端。该原语根据客户端的私钥、服务器的连带口令公钥和一个口令相关值生成一个共享秘密数据，应保证输入的口令由相同的域参数生成。

SVDP-{4,5}需要一个散列函数，该散列函数的可选择项为 11.1 中定义的函数和 MGF1。同时应该保证服务器在原语[DL]SVDP-{4,5}-SERVER 选择相同的散列函数。

SVDP-5 需要一个乘法元生成函数 MVCF，该 MVCF 应该选择 MVCF-1，同时应该保证服务器在原语[DL]SVDP-5-SERVER 选择相同的 MVCF 函数。

输入：

a) 客户端的私钥 s；

b) 口令限制私钥 u_π；

c) 口令验证数据(口令限制公钥)v_π；

d) 仅适用于 SVDP-5：客户端的 DL 私钥 w_c；

e) 服务器的连带口令私钥 w_s；

f) 与 s，u_π，v_π，w_c，w_s 相关的域参数。

假设：客户端的私钥 s，u_π，v_π 和域参数都是有效的，w_c，w_s 是父群的元素。

输出：一个属于有限域 $GF(q)$ 中的元素 z。

操作：

a) 计算八位位组串 $o_2=\mathrm{FE2OSP}(w_s)$；

b) 计算整数 i_2：

1) 在 SVDP-4 中：计算 $o_3=\mathrm{hash}(o_2)$；计算整数 $i_1=\mathrm{OS2IP}(o_3)$；计算 $i_2=\lfloor i_1/2^{128}\rfloor$；

2) 在 SVDP-5 中：计算八位位组串 $o_1=\mathrm{FE2OSP}(o_2)$；计算八位位组串 $o_3=\mathrm{hash}(o_1||o_2)$；计算整数 $i_2=\mathrm{OS2IP}(o_3)$；

c) 计算域元素 m_v：

1) 在 SVDP-4 中：$m_v=1$；

2) 在 SVDP-5 中：$m_v=\mathrm{MVCF}()$；

d) 计算 $z=\exp((w_s-v_\pi\times m_v),(s+i_2\times u_\pi))$；

e) 输出 z。

9.7.5 [DL]SVDP-{4,5}-SERVER

[DL]SVDP-{4,5}-SERVER 原语适用于服务器。该原语根据服务器的私钥、客户端的公钥和一个口令相关值生成一个共享秘密数据，应该保证输入的口令由相同的域参数生成。

[DL]SVDP-{4,5}-SERVER 原语需要一个散列函数，该散列函数的可选择项为 11.1 中定义的函数和 MGF1。同时应该保证服务器在原语[DL]SVDP-{4,5}-CLIENT 选择相同的散列函数；

输入：

a) 服务器的私钥 s；

b) 口令验证数据(口令限制公钥)v_π;

c) 客户端的私钥 w_c;

d) 服务器的连带口令私钥 w_s;

e) 与 s,v_π,w_c,w_s相关的域参数。

假设:客户端的私钥 s,v_π和域参数都是有效的,w_c,w_s是父群的元素。

输出:一个属于有限域 $GF(q)$中的元素 z。

操作:

a) 计算八位位组串 o_2=FE2OSP(w_s);

b) 计算整数 i_2:

1) 在 SVDP-4 中:计算 o_3=hash(o_2);计算整数 i_1=OS2IP(o_3);计算 $i_2=\lfloor i_1/2^{128} \rfloor$;

2) 在 SVDP-5 中:计算八位位组串 o_1=FE2OSP(o_2);计算八位位组串 o_3=hash($o_1||o_2$);计算整数 i_2=OS2IP(o_3);

c) 计算 z=exp(((w_c×exp(v_π, i_2)), s);

d) 输出 z。

9.7.6 [EC]SVDP-6-CLIENT

[EC]SVDP-6-CLIENT 原语适用于客户端。该原语根据客户端的私钥、服务器的连带口令公钥和一个口令相关值生成一个共享秘密数据,应该保证输入的口令由相同的域参数生成。

该原语需要以下两个参数:

a) 一个散列函数,该散列函数的可选择项为 11.1 中定义的函数和 MGF1。同时应该保证服务器在原语[EC]SVDP-6-SERVER 选择相同的散列函数;

b) 一个随机元素生成原语 REDP,该 REDP 应该选择[EC]REDP-1 和[EC]REDP-2,同时应该保证服务器在原语[EC]SVDP-6-SERVER 选择相同的随机元素生成函数。

输入:

a) 客户端的私钥 s;

b) 口令限制私钥 u_π;

c) 口令限制公钥 v_π=[EC]PVDGP-3(π);

d) 客户端的公钥 w_c;

e) 服务器的连带口令公钥 w_s;

f) 与 s,u_π,v_π,w_c,w_s相关的域参数。

假设:客户端的私钥 s,u_π,v_π和域参数都是有效的,w_c,w_s是父群的元素。

输出:一个属于有限域 $GF(q)$中的元素 z。

操作:

a) 计算 o_1=GE2OSP(w_c);

b) 计算 o_2=GE2OSP(w_s);

c) 计算 o_3=hash($o_1||o_2$);

d) 计算 i_2=OS2IP(o_3);

e) 计算 o_4=GE2OSP(v_π);

f) 计算群元素 e_1=REDP(o_4);

g) 计算 e_2=w_s * (e_1^(−1));

h) 计算 z_g=e_2^(s+(i_2u_π));

i) 计算 z=GE2FEP(z_g);

j) 输出 z。

9.7.7 [EC]SVDP-6-SERVER

[EC]SVDP-6-SERVER 原语适用于服务器。该原语根据服务器的私钥、服务器的连带口令公钥、口令验证数据和客户端的连带口令公钥生成一个共享秘密值。调用该原语的方案应该保证输入的口令由相同的域参数生成。

该原语需要一个散列函数作为参数，该散列函数的可选择项为 11.1 中定义的函数和 MGF1。同时应该保证服务器在原语 ECSVDP-6-CLIENT 选择相同的散列函数；

输入：

a) 服务器的私钥 s；

b) 口令验证数据 v_π；

c) 客户端的公钥 w_c；

d) 服务器的连带口令公钥 w_s；

e) 与 s，v_π，w_c，w_s 相关的域参数。

假设：客户端的私钥 s，u_π，v_π，w_c，w_s 和域参数都是有效的。

输出：一个属于有限域 $GF(q)$ 中的元素 z。

操作：

a) 计算 $o_1 = \mathrm{GE2OSP}(w_c)$；

b) 计算 $o_2 = \mathrm{GE2OSP}(w_s)$；

c) 计算 $o_3 = \mathrm{hash}(o_1 || o_2)$；

d) 计算 $i_2 = \mathrm{OS2IP}(o_3)$；

e) 计算 $z_g = (w_c * (v_\pi \hat{} i_2)) \hat{} s$；

f) 计算 $z = \mathrm{GE2FEP}(z_g)$；

g) 输出 z。

9.7.8 SVDP-7-CLIENT，SVDP-7-SERVER

{DL，EC} SVDP-7-{CLIENT，SERVER}是{离散对数，椭圆曲线}秘密值产生原语，该原语适用于客户端和服务器。原语 SVDP-7-CLIENT 使用客户端的口令相关值、私钥、和服务器的连带口令公钥生成共享的秘密值。SVDP-7-SERVER 根据口令验证数据，服务器的连带口令私钥和连带口令公钥及客户端的连带口令公钥生成一个共享秘密。

该原语需要一个散列函数作为参数，该散列函数的可选择项为 11.1 中定义的函数和 MGF1。同时应该保证 SVDP-7-CLIENT，SVDP-7-SERVER 选择相同的散列函数。

输入：

a) 参与方的私钥 s；

b) 适用于客户端：连带口令私钥 u_π；

c) 适用于服务器：口令验证数据 v_π；

d) 适用于服务器：客户端的连带口令公钥 w_c；

e) 服务器的连带口令公钥 w_s；

f) 与密钥 s，u_π，v_π，w_c，w_s 相关的{DL，EC}域参数(包括 q)；

假设：密钥 s，u_π，v_π，w_c，w_s 和相关的{DL，EC}域参数是有效的。

输出：一个属于有限域 $GF(q)$ 中的元素 z。

操作：

a) 计算 o_1=GE2OSP(w_s)；

b) 计算 o_2=hash(o_1)；

c) 计算 i_1=OS2IP(o_2)；

d) 计算群元素 z_g：

1) 适用于客户端：$z_g=w_s\hat{}\ (s+u_\pi\hat{}\ i_1)$；

2) 适用于服务器：$z_g=(w_c * (v_\pi\hat{}\ i_1))\hat{}\ s$；

e) 计算 z =GE2FEP(z_g)；

f) 输出 z。

9.8 密钥检索原语

9.8.1 KRBP-1

{DL,EC}KRBP-1 原语用于生成盲化的口令。{DL,EC}KRBP-1 使用客户端的私钥、基于口令生成的群生成元，生成盲化的口令。

输入：

a) 客户端的私钥 s；

b) 从口令生成的群生成元 g_π；

c) 与密钥 s 和 w_s 相关的{DL,EC}域参数(包括 r 和 g)。

假设：私钥 s 和域参数是有效的，g_π 是阶为 r 的群的生成元。

输出：一个公开的且被盲化的口令值 w_c。

操作：

a) 计算 $w_c=g_\pi\hat{}\ s$；

b) 输出盲化的口令值 w_c。

9.8.2 KRPP-1

{DL,EC}KRPP-1 原语用于密钥抽取的置换。{DL,EC}KRPP-1 使用服务器私钥、盲化的口令生成置换后的盲化口令。

输入：

a) 服务器的私钥 u；

b) 客户端发送的盲化口令 w_c；

c) 与密钥 u 和 w_c 相关的{DL,EC}域参数(包括 r)。

假设：私钥 u，w_c 和域参数是有效的。

输出：置换后的盲化口令 w_s。

操作：

a) 计算 $w_s=w_c\hat{}\ u$；

b) 输出置换的盲化口令 w_s。

9.8.3 KRUP-1

{DL,EC}KRUP-1 原语用于从置换后的盲化口令中恢复置换后的口令。

输入：

a) 客户端的私钥 s；

b) 服务器发送的置换的盲化口令 w_s；

c) 与密钥 s 和 w_s 相关的{DL,EC}域参数(包括 r 和 g)。

假设:私钥 s,w_s 和域参数是有效的。

输出:父群上的一个元素作为置换后的口令值。

操作:

a) 计算 $z_g = w_s \hat{} (s^{-1} \bmod r)$;

b) 输出置换的口令值 z_g。

10 口令鉴别密钥建立协议

10.1 BPKA-1

10.1.1 概述

{DL,EC}BPKA-1-{CLIENT,SERVER}是{离散对数,椭圆曲线}平衡型口令鉴别密钥协商协议,包含参与方{CLIENT,SERVER}。

BPKA-1 方案中客户端必须在使用 Z 或者 $K_1, K_2 \cdots K_t$ 之前确认服务器正确获得了 Z。

注:BPKA-1 方案基于参考文献[1]。

10.1.2 协议选项

客户端和服务器应建立或协商确定以下选项:

a) 公钥产生原语、秘密值产生原语和相关的参数:可选项为 PEPKGP-1,PKGP-1,SVDP-1-CLIENT,SVDP-2;

b) 共享的八位位组串口令 π;

c) 和 π 相关的有效的{DL,EC}域参数,包括 q,r,密钥 s , w_c , w_s;

d) 随机元素生成函数:可选项为{DL,EC}REDP-1,{DL,EC}REDP-2;

e) 密钥产生函数 KDF:可选项为 KDF1,KDF2;

f) 一个或者多个密钥产生参数八位位组串组$\{P_1, P_2 \cdots\}$;

g) 密钥证实函数:可选项为 KCF1。

10.1.3 密钥协商操作

10.1.3.1 客户端密钥协商

客户端执行以下操作:

a) 由 π 计算群元素 π_m = REDP(π);

b) 随机选择一个整数 $s \in [1, r-1]$,作为私钥;

c) 计算连带口令公钥 w_c = {DL,EC}PEPKGP-1(s,π_m);

d) 发送 w_c 给服务器;

e) 从服务器接收公钥 w_s;

f) 如果 w_s 不是群的有效元素,则输出“invalid”并停止;

g) 计算域元素 z = {DL,EC}SVDP-1-CLIENT(s, w_s);

h) 计算八位位组串 Z = FE2OSP(z);

i) 对每一个密钥产生参数 P_i,根据 Z,产生一个共享密钥 K_i = KDF(Z,P_i);

j) 输出产生的密钥 $K_1, K_2 \cdots K_t$。

10.1.3.2 服务器密钥协商

服务器执行以下操作：

a) 由 π 计算群元素 π_m=REDP(π)；

b) 随机选择一个整数 $s\in[1,r-1]$，作为私钥；

c) 计算连带口令公钥 w_s={DL,EC}PKGP-1(s)；

d) 发送 w_s 给客户端；

e) 从客户端接收连带口令公钥 w_c；

f) 如果 w_c 不是群的有效元素，则输出“invalid”并停止；

g) 计算域元素 z ={DL,EC}SVDP-2(s,π_m, w_s)；

h) 计算八位位组串 Z=FE2OSP(z)；

i) 对每一个密钥产生参数 P_i，根据 Z，产生一个共享密钥 K_i=KDF(Z,P_i)；

j) 输出产生的密钥 K_1,K_2…K_t。

10.1.4 密钥证实操作

10.1.4.1 服务器密钥证实

客户端执行以下操作：

a) 必选项

1) 计算 o_π=GE2OSP(π_m)；

2) 计算 o_s=KCF(hex(03),w_c,w_s,Z, o_π)；

3) 发送 o_s 给客户端。

b) 可选项

1) 从客户端接收八位位组串 o_c；

2) 计算 o_π=GE2OSP(π_m)；

3) 计算 o_4=KCF(hex(04), w_c, w_s, Z, o_π)；

4) 如果 $o_4\neq o_c$，输出“invalid”并停止。

10.1.4.2 客户端密钥证实

服务器执行以下操作：

a) 必选项

1) 从服务器接收八位位组串 o_s；

2) 计算 o_π=GE2OSP(π_m)；

3) 计算 o_3=KCF(hex(03),w_c,w_s, Z, o_π)；

4) 如果 $o_3\neq o_s$，输出“invalid”并停止。

b) 可选项

1) 计算 o_π=GE2OSP(π_m)；

2) 计算 o_c=KCF(hex(04),w_c,w_s, Z, o_π)；

3) 发送 o_c 给服务器。

10.2 BPKA-2

10.2.1 概述

{DL,EC}BPKA-2-{CLIENT,SERVER}是{离散对数，椭圆曲线}口令鉴别密钥协商方案，其中方

案包含参与方{CLIENT,SERVER}。

BPKA-2 方案中客户端可以选择是否在使用 Z 或者 $K_1,K_2\cdots K_l$ 之前确认服务器正确获得了 Z。对于服务器而言该步骤同样是可选的。

注：BPKA-2 方案基于参考文献[1]。

10.2.2 协议选项

客户端和服务器应该建立或者协商确定以下选项：

a) 公钥产生原语和秘密值产生原语：应选择 PEPKGP-1，SVDP-2；

b) 共享的基于口令的八位位组串 π；

c) 和 π 相关的有效的{DL,EC}域参数，包括 q，r，以及密钥 s，w_e，w_s；

d) 随机元素生成函数 REDP：可选项为{DL,EC}REDP-1，{DL,EC}REDP-2；

e) 密钥产生函数 KDF：可选项为 KDF1，KDF2；

f) 一个或者多个密钥产生参数八位位组串{P_1，$P_2\cdots$}；

g) 密钥证实函数：可选项为 KCF1。

10.2.3 密钥协商操作

10.2.3.1 客户端密钥协商

客户端执行以下操作：

a) 计算第一个群元素 π_T=REDP(hex(01)||π)；

b) 计算第二个群元素 π_R=REDP(hex(02)||π)；

c) 随机选择一个整数 $s\in[1,r-1]$，作为私钥；

d) 计算连带口令公钥 w_e={DL,EC}PEPKGP-1(s，π_T)；

e) 发送 w_e 给服务器；

f) 从服务器接收公钥 w_s；

g) 如果 w_s 不是群的有效元素，则输出“invalid”并停止；

h) 计算域元素 z ={DL,EC}SVDP-2(s，π_R，w_s)；

i) 计算八位位组串 Z=FE2OSP(z)；

j) 对每一个密钥产生参数 P_i，根据 Z，产生一个共享密钥 K_i=KDF(Z，P_i)；

k) 输出产生的密钥 $K_1,K_2\cdots K_l$。

10.2.3.2 服务器密钥协商

服务器执行以下操作：

a) 计算第一个群元素 π_T=REDP(hex(02)||π)；

b) 计算第二个群元素 π_R=REDP(hex(01)||π)；

c) 随机选择一个整数 $s\in[1,r-1]$，作为私钥；

d) 计算连带口令公钥 w_s={DL,EC}PEPKGP-1(s，π_T)；

e) 发送 w_s 给客户端；

f) 从客户端接收公钥 w_e；

g) 如果 w_e 不是群的有效元素，则输出“invalid”并停止；

h) 计算域元素 z ={DL,EC}SVDP-2(s，π_R，w_e)；

i) 计算八位位组串 Z=FE2OSP(z)；

j) 对每一个密钥产生参数 P_i，根据 Z，产生一个共享密钥 K_i=KDF(Z，P_i)；

k） 输出产生的密钥 K_1，K_2…K_t。

10.2.4 密钥证实操作

10.2.4.1 服务器密钥证实

客户端执行以下操作：

a） 可选项

1） 计算 o_T=GE2OSP(π_T)；

2） 计算 o_s=KCF(hex(03)，w_c，w_s，Z，o_T)；

3） 发送 o_s给客户端。

b） 可选项

1） 从客户端接收八位位组串 o_c；

2） 计算 o_R=GE2OSP(π_R)；

3） 计算 o_4=KCF(hex(04)，w_c，w_s，Z，o_R)；

4） 如果 $o_4 \neq o_c$，输出“invalid”并停止。

10.2.4.2 客户端密钥证实

服务器执行以下操作：

a） 可选项

1） 从服务器接收八位位组串 o_s；

2） 计算 o_R=GE2OSP(π_R)；

3） 计算 o_3=KCF(hex(03)，w_c，w_s，Z，o_R)；

4） 如果 $o_3 \neq o_s$，输出“invalid”并停止。

b） 可选项

1） 计算 o_T=GE2OSP(π_T)；

2） 计算 o_c=KCF(hex(04)，w_c，w_s，Z，o_π)；

3） 发送 o_c给服务器。

10.3 BPKA-3

10.3.1 概述

{DL，EC}BPKA-3-{CLIENT，SERVER}是{离散对数，椭圆曲线}口令鉴别密钥协商方案，包含参与方{CLIENT，SERVER}。

BPKA-3 方案中客户端可以选择是否在使用 Z 或者 K_1，K_2…K_t之前确认服务器正确获得了 Z。对于服务器而言该步骤同样是可选的。

注：BPKA-3 方案基于参考文献[2，3]。

10.3.2 协议选项

客户端和服务器应该建立或者协商确定以下选项：

a） 公钥产生原语和秘密值产生原语和相关的参数，应为 PEPKGP-2，SVDP-3；

b） 一个布尔值 b_1表示是否使用乘法因子；

c） 共享的基于口令的八位位组串 π；

d） 和 π 相关的有效的{DL，EC}域参数，包括 q，r；以及密钥 s，w，w'；

e） 随机元素生成函数 REDP：可选项为{DL，EC}REDP-1，{DL，EC}REDP-2；

f) 密钥产生函数 KDF：可选项为 KDF1，KDF2；

g) 一个或者多个密钥产生参数八位位组串{P_1，P_2…}；

h) 密钥证实函数：可选项为 KCF1。

10.3.3 密钥协商操作

两个参与方应该执行如下(或者等同的)操作产生共享的密钥 K_1，K_2…K_t：

a) 计算群元素 g_π=REDP(π)；

b) 随机选择一个整数 $s\in[1,r-1]$，作为私钥；

c) 计算连带口令公钥 w={DL，EC}PEPKGP-2(s，g_π)；

d) 发送 w 给另一参与方(另一参与方收到后作为 w')；

e) 从另一参与方接收公钥 w'：

1) 如果 w'不是群的有效元素，则输出“invalid”并停止；

2) 如果 w'的阶过小不能接受，则输出“invalid”并停止；

3) (可选)如果 w'不是有效的公钥，则输出“invalid”并停止；

f) 计算域元素 z ={DL，EC}SVDP-3(s，w'，b_1)；

g) 计算八位位组串 Z=FE2OSP(z)；

h) 对每一个密钥产生参数 P_i，根据 Z，产生一个共享密钥 K_i=KDF(Z，P_i)；

i) 输出产生的密钥 K_1，K_2…K_t。

10.3.4 密钥证实操作

10.3.4.1 服务器密钥证实

客户端执行以下操作：

a) 可选项

1) 计算 o_π=GE2OSP(g_π)；

2) 计算 o_s=KCF(hex(03)，w'，w，Z，o_π)；

3) 发送 o_s给客户端。

b) 可选项

1) 从客户端接收八位位组串 o_c；

2) 计算 o_π=GE2OSP(g_π)；

3) 计算 o_4=KCF(hex(04)，w'，w，Z，o_π)；

4) 如果 $o_4\neq o_c$，输出“invalid”并停止。

10.3.4.2 客户端密钥证实

服务器执行以下操作：

a) 可选项

1) 从服务器接收八位位组串 o_s；

2) 计算 o_π=GE2OSP(g_π)；

3) 计算 o_3=KCF(hex(03)，w，w'，Z，o_π)；

4) 如果 $o_3\neq o_s$，输出“invalid”并停止。

b) 可选项

1) 计算 o_π=GE2OSP(g_π)；

2) 计算 o_c=KCF(hex(04)，w，w'，Z，o_π)；

3） 发送 o_c 给服务器。

10.4 APKA-1

10.4.1 概述

{DL,EC}APKA-1-{CLIENT,SERVER}是{离散对数,椭圆曲线}口令鉴别密钥协商方案,该方案是增强型口令鉴别密钥协商方案,包含参与方{CLIENT,SERVER}。在该方案中,客户端拥有口令,而服务器拥有口令的验证数据。

APKA-1 方案中客户端可以选择是否在使用 Z 或者 $K_1,K_2\cdots K_t$ 之前确认服务器正确获得了 Z。对于服务器而言该步骤同样是可选的。

注：APKA-1 方案基于参考文献[3,4]。

10.4.2 协议选项

客户端和服务器应该建立或者协商确定以下选项：

a） 口令相关数据：

1） 适用于客户端：口令 π；

2） 适用于服务器：使用{DL,EC}PVDGP-1 根据 π 生成口令验证元素 g_π 和口令限制公钥 v_π；

b） 公钥产生原语和秘密值产生原语和相关的参数：可选原语为 PVDGP-1,PEPKGP-2,SVDP-3；

c） 和 v_π,g_π,π 相关的有效的{DL,EC}域参数,包括 q,g,r,以及密钥 s,w_c,w_s；

d） 一个布尔值 b_1 表明是否希望进行因子乘法运算；

e） 密钥产生函数 KDF：可选项为 KDF1,KDF2；

f） 一个或者多个密钥产生参数八位位组串{$P_1,P_2\cdots$}；

g） 密钥证实函数：可选项为 KCF1。

10.4.3 密钥协商操作

10.4.3.1 客户端密钥协商

客户端执行以下操作：

a） 使用{DL,EC}PVDGP-1(π) 计算生成元 g_π 和受限于口令的私钥 u_π；

b） 随机选择一个整数 $s\in[1,r-1]$,作为私钥；

c） 计算连带口令公钥 $w_c=\{DL,EC\}PEPKGP\text{-}2(s,g_\pi)$；

d） 发送 w_c 给服务器；

e） 从服务器接收连带口令公钥 w_s：

1） 如果 w_s 不是群的有效元素,则输出“invalid”并停止；

2） 如果 w_s 的阶过小不能接受,则输出“invalid”并停止；

3） (可选)如果 w_s 不是有效的公钥,则输出“invalid”并停止；

f） 计算域元素 $z_1=\{DL,EC\}SVDP\text{-}3(s,w_s,b_1)$；

g） 计算域元素 $z_2=\{DL,EC\}SVDP\text{-}3(u_\pi,w_s,b_1)$；

h） 计算八位位组串 $Z_1=FE2OSP(z_1)$；

i） 计算八位位组串 $Z_2=FE2OSP(z_2)$；

j） $Z=Z_1||Z_2$；

k） 对每一个密钥产生参数 P_i,根据 Z,产生一个共享密钥 $K_i=KDF(Z,P_i)$；

l） 输出产生的密钥 $K_1,K_2\cdots K_t$。

10.4.3.2 服务器密钥协商

服务器执行以下操作：

a) 随机选择一个整数 $s\in[1,r-1]$作为私钥；

b) 计算连带口令公钥 $w_s=\{DL,EC\}PEPKGP\text{-}2(s,g_\pi)$；

c) 发送 w_s给客户端；

d) 从客户端接收连带口令公钥 w_c：

1) 如果 w_c不是群的有效元素，则输出“invalid”并停止；

2) 如果 w_c的阶过小不能接受，则输出“invalid”并停止；

3) (可选)如果 w_c不是有效的公钥，则输出“invalid”并停止；

e) 计算域元素 $z_1=\{DL,EC\}SVDP\text{-}3(s,w_c,b_1)$；

f) 计算域元素 $z_2=\{DL,EC\}SVDP\text{-}3(s,v_\pi,b_1)$；

g) 计算八位位组串 $Z_1=FE2OSP(z_1)$；

h) 计算八位位组串 $Z_2=FE2OSP(z_2)$；

i) $Z=Z_1||Z_2$；

j) 对每一个密钥产生参数 P_i，根据 Z，产生一个共享密钥 $K_i=KDF(Z,P_i)$；

k) 输出产生的密钥 $K_1,K_2\cdots K_t$。

10.4.4 密钥证实操作

10.4.4.1 服务器密钥证实

客户端执行以下操作：

a) 可选项

1) 计算 $o_\pi=GE2OSP(g_\pi)$；

2) 计算 $o_s=KCF(hex(03),w_c,w_s,Z,o_\pi)$；

3) 发送 o_s给客户端。

b) 可选项

1) 从客户端接收八位位组串 o_c；

2) 计算 $o_\pi=GE2OSP(g_\pi)$；

3) 计算 $o_4=KCF(hex(04),w_c,w_s,Z,o_\pi)$；

4) 如果 $o_4\neq o_c$，输出“invalid”并停止。

10.4.4.2 客户端密钥证实

服务器执行以下操作：

a) 可选项

1) 从服务器接收八位位组串 o_s；

2) 计算 $o_\pi=GE2OSP(g_\pi)$；

3) 计算 $o_3=KCF(hex(03),w_c,w_s,Z,o_\pi)$；

4) 如果 $o_3\neq o_s$，输出“invalid”并停止。

b) 可选项

1) 计算 $o_\pi=GE2OSP(g_\pi)$；

2) 计算 $o_c=KCF(hex(04),w_c,w_s,Z,o_\pi)$；

3) 发送 o_c给服务器。

10.5 [DL]APKA-{2,3}

10.5.1 概述

[DL]APKA-{2,3}-{CLIENT,SERVER}是离散对数口令鉴别密钥协商方案,包含参与方{CLIENT,SERVER}。

[DL]APKA-{2,3}使用优化的 Diffie-Hellman 交换在两个参与方间直接建立共享密钥。

[DL]APKA-{2,3}方案中服务器必须在使用 Z 或者 $K_1,K_2\cdots K_t$ 之前确认客户端正确获得了 Z。对于客户端而言该步骤同样是可选的。

注:APKA-2 方案基于参考文献[5,6],APKA-3 基于参考文献[6]。Diffie-Hellman 交换参见参考文献[11]。

10.5.2 协议选项

客户端和服务器应该建立或者协商确定协商口令验证数据生成原语,公钥生成原语,秘密值生成原语和相关参数。可选项包括为[DL]PVDGP-2,[DL]PKGP-2-CLIENT,[DL]PEPKGP-{3, 4}-SERVER,[DL]SVDP-{4,5}-CLIENT,[DL]SVDP-{4,5}-SERVER。APKA-3 客户端和服务器必须使用相同的散列函数参数和相同的 MVCF 参数。

a) 口令相关秘密和验证数据:
   1) 适用于客户端:口令相关秘密八位位组串 π,可以包含"盐"和单方或者双方的身份信息;
   2) 适用于服务器:使用[DL]PVDGP-2 根据口令相关秘密 π 生成受口令限制公钥 v_π;
b) 和 π,v_π 相关的有效的 DL 域参数,包括 g_{q-1},q,r,k,以及密钥 s,w_c,w_s;
c) 密钥产生函数 KDF:可选项为 KDF1,KDF2;
d) 一个或者多个密钥产生参数八位位组串{$P_1,P_2\cdots$};
e) 密钥证实函数:可选项为 KCF1。

10.5.3 密钥协商操作

10.5.3.1 客户端密钥协商

客户端执行以下操作:

a) 随机选择一个整数 $s\in[1,q-2]$作为私钥;
b) 计算 DL 公钥 $w_c=$[DL]PKGP-2-CLIENT(s);
c) 发送 w_c 给服务器;
d) 从服务器接收连带口令公钥 w_s;如果 w_s 不是群的有效元素,则输出"invalid"并停止;
e) 使用原语[DL]PVDGP-2(π)计算受限于口令的私钥 u_π 和口令验证数据 v_π;
f) 计算协商域元素 z:
   1) APKA-2:$z=$[DL]SVDP-4-CLIENT(s, u_π, v_π, w_s);
   2) APKA-3:$z=$[DL]SVDP-5-CLIENT(s, u_π, v_π, w_c, w_s);
g) 计算八位位组串 $Z=$FE2OSP(z);
h) 对每一个密钥产生参数 P_i,根据 Z,产生一个共享密钥 $K_i=$KDF(Z, P_i);
i) 输出产生的密钥 $K_1,K_2\cdots K_t$。

10.5.3.2 服务器密钥协商

服务器执行以下操作:

a) 随机选择一个整数 $s\in[1,q-2]$,作为私钥;
b) 计算连带口令公钥 w_s;

1) APKA-2：w_s=[DL]PEPKGP-3-SERVER(s，v_π)；
2) APKA-3：w_s=[DL]PEPKGP-4-SERVER(s，v_π)；

c) 从客户端接收公钥 w_c；如果 w_c 不是群的有效元素，则输出“invalid”并停止；

d) 发送 w_s 给客户端；

e) 计算协商域元素 z：
1) APKA-2：z=[DL]SVDP-4-SERVER(s，v_π，w_c，w_s)；
2) APKA-3：z=[DL]SVDP-5-SERVER(s，v_π，w_c，w_s)；

f) 计算八位位组串 Z=FE2OSP(z)；

g) 对每一个密钥产生参数 P_i，根据 Z，产生一个共享密钥 K_i=KDF(Z，P_i)；

h) 输出产生的密钥 K_1，K_2…K_t。

10.5.4 密钥证实操作

10.5.4.1 客户端密钥证实

服务器执行以下操作：

a) 必选项
1) 计算 o_π=FE2OSP(v_π)；
2) 计算 o_c=KCF(hex(04)，w_c，w_s，Z，o_π)；
3) 发送 o_c 给服务器。

b) 可选项
1) 从服务器接收八位位组串 o_s；
2) 计算 o_4=KCF(hex(03)，w_c，w_s，Z，o_π)；
3) 如果 $o_4 \neq o_s$，输出“invalid”并停止。

10.5.4.2 服务器密钥证实

客户端执行以下操作：

a) 必选项
1) 从客户端接收八位位组串 o_c；
2) 计算 o_π=FE2OSP(v_π)；
3) 计算 o_3=KCF(hex(04)，w_c，w_s，Z，o_π)；
4) 如果 $o_3 \neq o_c$，输出“invalid”并停止。

b) 可选项
1) 计算 o_s=KCF(hex(03)，w_c，w_s，Z，o_π)；
2) 发送 o_s 给客户端。

10.6 [EC]APKA-4

10.6.1 概述

[EC]APKA-4-{CLIENT，SERVER}是椭圆曲线口令鉴别密钥协商方案，包含参与方{CLIENT，SERVER}。

[EC]APKA-4 使用优化的基于口令的 Diffie-Hellman 交换在两个参与方直接建立共享密钥。客户端拥有基于口令的八进制值 π，服务器拥有使用 PVDGP 函数根据客户端的口令生成的口令验证数据。

注：APKA-4 方案基于参考文献[6]。

APKA-4 方案中服务器必须在使用 Z 或者 $K_1, K_2 \cdots K_t$ 之前确认客户端正确获得了 Z。对于客户端而言该步骤同样是可选的。

10.6.2 协议选项

客户端和服务器应该建立或者协商确定协商口令验证数据生成原语，公钥生成原语，秘密值生成原语和相关参数。可选项为[EC]PVDGP-3，[EC]PKGP-1，[EC]PEPKGP-5-SERVER，[EC]SVDP-6-CLIENT，[EC]SVDP-6-SERVER。客户端和服务器必须使用相同的域参数。

a) 口令相关秘密和验证数据：
   1) 适用于客户端：基于口令的八位位组串 π，可以包含“盐”和单方或者双方的身份信息；
   2) 适用于服务器：使用[EC]PVDGP-3 根据输入 π 生成口令限制公钥 v_π；
b) 和 π, v_π 相关的有效的 EC 域参数；以及密钥 s, w_c, w_s；
c) 密钥产生函数 KDF：可选项为 KDF1，KDF2；
d) 一个或者多个密钥产生参数八位位组串 $\{P_1, P_2 \cdots\}$；
e) 密钥证实函数：可选项为 KCF1。

10.6.3 密钥协商操作

10.6.3.1 客户端密钥协商

客户端执行以下操作：

a) 随机选择一个整数 $s \in [1, r-1]$，作为私钥；
b) 计算 DL 公钥 w_c = [EC]PKGP-1 (s)；
c) 发送 w_c 给服务器；
d) 从服务器接收连带口令公钥 w_s；如果 w_s 不是群的有效元素，则输出“invalid”并停止；
e) 使用原语[EC]PVDGP-3(π)计算受限于口令的私钥 u_π 和口令验证数据 v_π；
f) 计算协商域元素 z = [DL]SVDP-6-CLIENT(s, u_π, v_π, w_c, w_s)；
g) 计算八位位组串 Z = FE2OSP(z)；
h) 对每一个密钥产生参数 P_i，根据 Z，产生一个共享密钥 K_i = KDF(Z, P_i)；
i) 输出产生的密钥 $K_1, K_2 \cdots K_t$。

10.6.3.2 服务器密钥协商

服务器执行以下操作：

a) 随机选择一个整数 $s \in [1, r-1]$，作为私钥；
b) 计算连带口令公钥 w_s = [EC]PEPKGP-5-CLIENT (s, v_π)；
c) 发送 w_s 给客户端；
d) 从客户端接收公钥 w_c；如果 w_c 不是群的有效元素，则输出“invalid”并停止；
e) 计算协商域元素 z = [DL]SVDP-6-SERVER(s, v_π, w_c, w_s)；
f) 计算八位位组串 Z = FE2OSP(z)；
g) 对每一个密钥产生参数 P_i，根据 Z，产生一个共享密钥 K_i = KDF(Z, P_i)；
h) 输出产生的密钥 $K_1, K_2 \cdots K_t$。

10.6.4 密钥证实操作

10.6.4.1 客户端密钥证实

服务器执行以下操作：

a) 必选项

1) 计算 o_π=FE2OSP(v_π)；

2) 计算 o_c=KCF(hex(04)，w_c，w_s，Z，o_π)；

3) 发送 o_c 给服务器。

b) 可选项

1) 从服务器接收八位位组串 o_s；

2) 计算 o_4=KCF(hex(03)，w_c，w_s，Z，o_π)；

3) 如果 $o_4 \neq o_s$，输出“invalid”并停止。

10.6.4.2 服务器密钥证实

客户端执行以下操作：

a) 必选项

1) 计算 o_π=FE2OSP(v_π)；

2) 从客户端接收八位位组串 o_c；

3) 计算 o_3=KCF(hex(04)，w_c，w_s，Z，o_π)；

4) 如果 $o_3 \neq o_c$，输出“invalid”并停止。

b) 可选项

1) 计算 o_s=KCF(hex(03)，w_c，w_s，Z，o_π)；

2) 发送 o_s 给客户端。

10.7 APKA-5

10.7.1 概述

{DL，EC}APKA-5-{CLIENT，SERVER}是{离散对数，椭圆曲线}口令鉴别密钥协商方案，包含参与方{CLIENT，SERVER}。

{DL，EC}APKA-5-{CLIENT，SERVER}使用优化的 Diffie-Hellman 交换在两个参与方直接建立共享密钥。客户端拥有基于口令的八进制值 π，服务器拥有使用 PVDGP 函数根据客户端的口令生成的口令验证数据。

APKA-5 方案中客户端可以选择是否在使用 Z 或者 K_1，$K_2 \cdots K_t$ 之前确认服务器正确获得了 Z。对于服务器而言该步骤同样是可选的。

注：APKA-5 方案基于参考文献[3,4]。

10.7.2 协议选项

客户端和服务器应该建立或者协商确定公钥产生原语和秘密值产生原语和相关的参数。可选项：PVDGP-1，PEPKGP-2，SVDP-7-CLIENT，SVDP-7-SERVER；客户端和服务器应该使用相同的散列函数参数。

a) 口令相关秘密和验证数据：

1) 适用于客户端：基于口令的八位位组串 π，可以包含“盐”和单方或者双方的身份信息；

2) 适用于服务器：使用{DL，EC}PVDGP-1 产生口令验证生成元 g_π 和口令限制公钥 v_π；{DL，EC}PVDGP-1 的输入值和参数应和客户端口令协商中使用的相同；

b) 和 π，g_π，v_π 相关的{DL，EC}域参数；以及密钥 s，w_c，w_s；

c) 密钥产生函数 KDF：可选项为 KDF1，KDF2；

d) 一个或者多个密钥产生参数八位位组串{P_1，$P_2 \cdots$}；

e) 密钥证实函数：可选项为 KCF1。

10.7.3 密钥协商操作

10.7.3.1 客户端密钥协商

客户端执行以下操作：

a) 使用{DL,EC}PVDGP-1(π) 计算生成元 g_π；
b) 使用{DL,EC}PVDGP-1(π) 计算受限于口令的私钥 u_π；
c) 随机选择一个整数 $s\in[1,r-1]$，作为私钥；
d) 计算连带口令公钥 $w_c=\{DL,EC\}PEPKGP\text{-}2(s, g_\pi)$；
e) 发送 w_c 给服务器；
f) 从服务器接收连带口令公钥 w_s：
   1) 如果 w_s 不是群的有效元素，则输出“invalid”并停止；
   2) 如果 w_s 的阶过小不能接受，则输出“invalid”并停止；
   3) (可选)如果 w_s 不是有效的公钥，则输出“invalid”并停止；
g) 计算域元素 $z=\{DL,EC\}SVDP\text{-}7\text{-}CLIENT(s, u_\pi, w_s)$；
h) 计算八位位组串 $Z=FE2OSP(z)$；
i) 对每一个密钥产生参数 P_i，根据 Z，产生一个共享密钥 $K_i=KDF(Z, P_i)$；
j) 输出产生的密钥 $K_1, K_2 \cdots K_t$。

10.7.3.2 服务器密钥协商

服务器执行以下操作：

a) 随机选择一个整数 $s\in[1,r-1]$，作为私钥；
b) 计算连带口令公钥 $w_s=\{DL,EC\}PEPKGP\text{-}2(s, g_\pi)$；
c) 发送 w_s 给服务器；
d) 从客户端接收连带口令公钥 w_c：
   1) 如果 w_c 不是群的有效元素，则输出“invalid”并停止；
   2) 如果 w_c 的阶过小不能接受，则输出“invalid”并停止；
   3) (可选)如果 w_c 不是有效的公钥，则输出“invalid”并停止；
e) 计算域元素 $z=\{DL,EC\}SVDP\text{-}7\text{-}SERVER(s, u_\pi, w_c, w_s)$；
f) 计算八位位组串 $Z=FE2OSP(z)$；
g) 对每一个密钥产生参数 P_i，根据 Z，产生一个共享密钥 $K_i=KDF(Z, P_i)$；
h) 输出产生的密钥 $K_1, K_2 \cdots K_t$。

10.7.4 密钥证实操作

10.7.4.1 服务器密钥证实

客户端执行以下操作：

a) 可选项
   1) 计算 $o_\pi=GE2OSP(g_\pi)$；
   2) 计算 $o_s=KCF(hex(03), w_c, w_s, Z, o_\pi)$；
   3) 发送 o_s 给客户端。
b) 可选项
   1) 从客户端接收八位位组串 o_c；

2） 计算 o_π=GE2OSP(g_π)；

3） 计算 o_4=KCF(hex(04)，w_c，w_s，Z，o_π)；

4） 如果 $o_4 \neq o_c$，输出“invalid”并停止。

10.7.4.2 客户端密钥证实

服务器执行以下操作：

a） 可选项

1） 从服务器接收八位位组串 o_s；

2） 计算 o_π=GE2OSP(g_π)；

3） 计算 o_3=KCF(hex(03)，w_c，w_s，Z，o_π)；

4） 如果 $o_3 \neq o_s$，输出“invalid”并停止。

b） 可选项

1） 计算 o_π=GE2OSP(g_π)；

2） 计算 o_c=KCF(hex(04)，w_c，w_s，Z，o_π)；

3） 发送 o_c 给服务器。

10.8 PKR-1

10.8.1 概述

{DL，EC}PKRS-1-{CLIENT，SERVER}是{离散对数，椭圆曲线}口令鉴别密钥检索方案，包含参与方{CLIENT，SERVER}。密钥建立操作计算客户端与服务器短交互的值，并根据客户端的口令值 π 与服务器的私钥为客户端检索出长期密钥。

PKRS-1 方案使得伪造的客户端(不具有口令值 π)以及第三方观察者(获取到合法的客户端与服务器交互的所有信息)不能够获取到用户的口令，服务器的私钥，以及建立的长期密钥，并能够抵抗暴力攻击。另外，服务器相关的私钥并不包含足够的能够确定客户端口令以及所建立的长期密钥的信息。

注：PKR-1 方案基于参考文献[7，8]。

10.8.2 协议选项

客户端和服务器应该建立或者协商如下选项：

仅客户端适用选项：

a） 基于口令的八位位组串 π，可以包含“盐”和单方或者双方的身份信息；

b） 一个随机元素生成函数 REDP，应为{DL，EC}REDP-1 或者{DL，EC}REDP-2；

c） 一个密钥抽取函数，应为 KDF1 或者 KDF2；

d） 一个或多个密钥抽取参数字符串{P_1，P_2，…P_k}，用于抽取协商的密钥。

仅服务器适用选项：对应于客户端口令值 π 的服务器长期密钥 u。

客户端，服务器共同适用选项：

a） 盲化函数 KRBP-1，解盲化函数 KRUP-1，置换函数 KRPP-1 的原语以及相关的参数；

b） 合法的{离散对数，椭圆曲线}定义域集合(包括 q 和 r)以及相关的 π，u，s，w_c，w_s。

10.8.3 密钥建立操作

10.8.3.1 客户端密钥建立

客户端执行以下操作：

a） 计算生成元 g_π=REDP(π)；

b) 随机选择一个整数 $s \in [1, r-1]$，作为私钥；

c) 计算连带口令公钥 $w_c = \{DL,EC\}KRBP\text{-}1(s, g_x)$；

d) 发送 w_c 给服务器；

e) 从服务器接收被置换以及盲化的口令值 w_s：

1) 如果 w_s 不是群的有效元素，则输出“invalid”并停止；

2) （可选）如果 w_s 不是有效的公钥，则输出“invalid”并停止；

f) 计算置换的口令值 $z_g = \{DL,EC\}KRUP\text{-}1(s, w_s)$；

g) 计算八位位组串 $Z = GE2OSP(z_g)$；

h) 对每一个密钥产生参数 P_i，根据 Z，产生一个共享密钥 $K_i = KDF(Z, P_i)$；

i) 输出产生的密钥 $K_1, K_2 \cdots K_t$。

10.8.3.2 服务器密钥建立

服务器执行以下操作：

a) 从客户端接收盲化的口令值 w_c：

1) 如果 w_c 不是群的有效元素，则输出“invalid”并停止；

2) （可选）如果 w_c 不是有效的公钥，则输出“invalid”并停止；

b) 计算置换以及盲化的口令值 $w_s = \{DL,EC\}KRPP\text{-}1(u, w_c)$；

c) 发送 w_s 给客户端。

11 密码函数

11.1 散列函数

散列函数为可变长度的字符串输入生成一个确定长度的消息摘要字符串，输出的消息摘要的长度根据所选择的散列函数的不同，通常散列函数不限制输入的长度。散列函数的输出仅与函数的输入相关，即散列函数是确定性的，对于同一个输入产生的结果必然相同。

本标准中应用到的散列函数定义见 GB/T 18238.3—2002。

本标准同时支持符合国家密码管理规定和相关密码算法标准的散列函数。

11.2 掩码生成函数

11.2.1 概述

掩码生成函数与散列函数不同，它为可变长度的八位位组串生成可变长度的消息摘要。掩码生成函数通常需要一个整数进行初始化，其中该整数参数设定了掩码生成函数的输出长度，因此采用掩码生成函数可以为消息生成想要的特定长度的摘要值。与散列函数相同，掩码生成函数也是确定性的，即输出仅依赖于输入消息。

11.2.2 MGF1

该掩码生成函数需要使用选择的散列函数和输出长度参数进行初始化，其中散列函数应为 11.1 中所定义的函数中的一种，输出长度参数定义了 MGF1 的输出八位位组串的长度。

输入：

a) 长度为 *zLen* 的八位位组串 Z，或者长度为 *zBits* 的比特串 ZB（这取决于选择的散列函数，且可能存在长度限制）；

b) 指定的输出八位位组串的长度 *oLen*（若以八位位组串的形式返回输出），或者指定的输出比特

串的长度 $oBits$(若以比特串的形式返回输出);其中 $oLen$ 和 $oBits$ 均为正整数,且 $8 * oLen$ 和 $oBits$ 均小于 $hBits \times 2^{32}$,其中 $hBits$ 为所选择的散列函数的输出长度。

输出:

掩码值,可能为 $oLen$ 长的八位位组串 $mask$,或 $oBits$ 长的比特串 MB,或"error"。

操作:

a) 若输入为长度 $zLen$ 的八位位组串 Z,则转换为长度为 $zBits = 8 * zLen$ 的比特串 ZB;

b) 若指定输出为 $oLen$ 长的八位位组串,则计算 $oBits = 8 * oLen$;

c) 若($zBits + 32$)超出了所选散列函数规定的输入长度限制,或者 $oBits > hBits \times 2^{32}$,则输出"error"并终止;

d) 令 MB 为空串;计算 $cThreshold = \lceil (oBits) / (hBits) \rceil$;

e) 令 $counter = 0$;

1) 将 $counter$ 转换为 32 bits 长的比特串 CB;

2) 计算 $hBits$ 长的散列值 $HB = hash(ZB \,||\, CB)$;

3) 令 $MB = MB \,||\, HB$;

f) 如输出格式为比特串,则输出 MB;否则转换 MB 为八位位组串并输出。

11.3 密钥证实函数

11.3.1 概述

密钥证实函数(Key confirmation function)用来在远程口令鉴别和密钥协商方案中产生一个确认消息以证明参与方正确地获得了共享的密钥和相关的数据。KCF 函数的输入可能包括交互过程中产生的公开消息,也可能包括协议运行双方预共享的长期秘密。

11.3.2 KCF1

该密钥证实函数需要一个散列函数作为参数进行初始化,该散列函数应为 11.1 中定义的函数中的一个。

输入:

a) 密钥产生参数 P;

b) 客户端公钥 w_c;

c) 服务端公钥 w_s;

d) 共享密钥 Z;

e) 共享的口令 o_π;

f) 有效的 DL 或 EC 域参数。

输出:密钥证实八位位组串。

操作:

a) 计算 $o_c = \mathrm{GE2OSP}(w_c)$;

b) 计算 $o_s = \mathrm{GE2OSP}(w_s)$;

c) 计算 $o = \mathrm{hash}(P \,||\, o_c \,||\, o_s \,||\, Z \,||\, o_\pi)$;

d) 输出 o。

11.4 乘法元生成函数

11.4.1 概述

乘法元生成函数生成一个域乘法元素,该元素在[DL]SVDP-5-CLIENT 和[DL]SVDP-5-SERVER

原语之间共享，该原语分别有 APKA-3-CLIENT 和 APKA-3-SERVER 调用。该函数根据客户端和服务器共享信息生成域乘法元，该函数的输出仅与输入有关。MVCF 设计用来防止一个恶意的参与方强制使另一方使用他选择的相关域参数（在这种情况下，恶意参与方可以进行猜测攻击）。

11.4.2 MVCF-1

MVCF-1 函数根据一个散列函数和由客户端和服务器共享的域参数生成域乘法元。

输入：DL 域参数（包括 g_{q-1} 和 q）。

输出：一个有限域乘法元元素。

操作：

a) 使用 I2OSP 生成一个长度为 $\lceil \log_{256}(q+1) \rceil$ 域元素 o_1；

b) 计算八位位组串 $o_2 = \text{FE2OSP}(g_{q-1})$；

c) 计算八位位组串 $o_3 = \text{hash}(o_1 \| o_2)$；

d) 计算乘法元 $m_v = \text{I2FEP}(\text{OS2IP}(o_3) \bmod q)$；

e) 输出 m_v。

11.5 密钥导出原语

11.5.1 概述

密钥导出函数用在口令鉴别与密钥建立协议中生成会话密钥。密钥导出函数基于散列函数实现，散列函数的可选项为 11.1 中定义的函数。由于所选择的散列函数的输入长度限制，KDF 在输入长度超过散列函数输入长度时会输出“error”并停止。

11.5.2 KDF-1

输入：

a) 长度为 $zLen$ 的八位位组串 Z；

b) 长度为 $pLen$ 的密钥导出参数 P，P 为八位位组串。

输出：长度为 $kLen$ 的共享密钥 K。

操作：

a) 如果 $zLen + pLen$ 超过了所选择的散列函数的最大输入长度，则输出“error”并停止；

b) 计算 $K = \text{hash}(Z \| P)$；

c) 输出 K 作为共享密钥。

11.5.3 KDF-2

KDF-2 支持八位位组串和比特串格式的输入。若输入格式为比特串，则输出格式也为比特串，输出的长度使用比特长度表示。

输入：

a) 长度为 zLen 的八位位组串 Z，或者长度为 zBits 的比特串 ZB（这取决于选择的散列函数，且可能存在长度限制）。

b) 指定的输出八位位组串的长度 oLen（若以八位位组串的形式返回输出），或者指定的输出比特串的长度 $oBits$（若以比特串的形式返回输出）；其中 $oLen$ 和 $oBits$ 均为正整数，且 $8 * oLen$ 和 $oBits$ 均小于 $hBits \times 2^{32}$，其中 $hBits$ 为所选择的散列函数的输出长度。

c) 长度为 $pLen$ 的密钥导出参数八位位组串 P，或者长度为 $pBits$ 的比特串 PB。

输出：长度为 $oLen$ 或者 $oBits$ 的共享密钥。

操作：

a) 若输入为长度 $zLen$ 的八位位组串 Z，则转换为长度为 $zBits=8*zLen$ 的比特串 ZB；

b) 若指定输出为 $oLen$ 长的八位位组串，则计算 $oBits=8*oLen$；

c) 若($zBits+pBits+32$)超出了所选散列函数规定的输入长度限制，或者 $oBits>hBits\times(2^{32}-1)$，则输出“error”并终止；

d) 令 MB 为空串；计算 $cThreshold=\lceil oBits/hBits\rceil$；

e) 令 $counter=0$；

1) 将 $counter$ 转换为 32 bits 长的比特串 CB；

2) 计算 $hBits$ 长的散列值 $HB=hash(ZB||CB||PB)$；

3) 令 $MB=MB||HB$；

4) $counter=counter+1$，如果 $counter\leqslant cThreshold$，跳转到步骤 1)；

f) 如果输出是比特串，则输出 MB，否则转换 MB 为八位位组串并输出。