第8章　伪随机数的生成

Cryptography Theory and Practice(Stinson)

教材：密码学原理域实践（冯登国译）

内容

引言与实例

在密码学领域中，有时需要产生随机数、随机比特串。

例如，需要从一个指定的密钥空间中随机地生成密钥，而且许多加密和签名方案都需要在它们的执行过程中应用随机数。

通常的方法是投硬币或其他物理过程来产生随机数，但是费时又昂贵。

在实际中通常使用一个伪随机比特生成器来产生随机数。

引言与实例

一个比特生成器可以将一个较短的随机比特串（种子）拓展成一个较长的比特串。这样，一个比特生成器降低了密码学应用中需要的随机比特的数量。

引言与实例

一个比特生成器可以将一个较短的随机比特串（种子）拓展成一个较长的比特串。这样，一个比特生成器降低了密码学应用中需要的随机比特的数量。

定义8.1 设k,l为两个满足$l \geq k+1$的正整数。一个(k,l)比特生成器是一个可在多项式时间内计算的函数$f:(\mathbb{Z}_2)^k \to (\mathbb{Z}_2)^l$。我们称输入$s_0 \in (\mathbb{Z}_2)^k$为种子，而将输出$f(s_0) \in (\mathbb{Z}_2)^l$为生成的比特串。通常要求$l$是$k$的一个多项式函数。

引言与实例

在定义8.1中的函数f是确定性的，这就要求比特串$f(s_0)$值依赖于种子。我们的目的是在种子比特随机选择的前提条件下，要使得产生的比特串$f(s_0)$应该“看起来”像真正的随机比特串。

如果这条性质满足的话，那么这个比特生成器就是安全的，并称为一个伪随机比特生成器。

引言与实例

研究伪随机比特生成器的动机：一次一密体制。

引言与实例

研究伪随机比特生成器的动机：一次一密体制。

随机数生成器广泛应用于包括密码学在内的各个计算机科学领域。比如：模拟、Monte Carlo算法、采样、测试。

引言与实例

研究伪随机比特生成器的动机：一次一密体制。

随机数生成器广泛应用于包括密码学在内的各个计算机科学领域。比如：模拟、Monte Carlo算法、采样、测试。

随机比特生成器所产生的随机数具有一个相对均匀的分布就够了。

引言与实例

研究伪随机比特生成器的动机：一次一密体制。

随机数生成器广泛应用于包括密码学在内的各个计算机科学领域。比如：模拟、Monte Carlo算法、采样、测试。

随机比特生成器所产生的随机数具有一个相对均匀的分布就够了。

衡量一个伪随机数序列各种随机特征的指标包括频率、游程、序列中数之间的间隔等。

引言与实例

在密码学意义下的伪随机比特生成器的安全性。

引言与实例

在密码学意义下的伪随机比特生成器的安全性。

一种重要的伪随机比特生成器，线性反馈移位寄存器(LFSR)。给定一个k比特的种子，一个k阶LFSR在重复前能够用来产生多达$2^k - k - 1$个伪随机比特。然而，由一个LFSR作为比特生成器是非常不安全的：从1.2.5节我们知道，使用任何$2k$个连续比特足以确定出种子，从而整条序列就可以被敌手重构。

引言与实例

线性同余生成器

设$M \geq 2$是一个整数，$1 \leq a, b \leq M-1$。定义$k = 1 + \lfloor lbM \rfloor$，并令$k+l \leq l \leq M-1$。

种子是一个整数s_0，这里$0 \leq s_0 \leq M-1$。注意到一个种子的二元表示就是一个长度不超过k的比特串；然而，并非所有的k长比特串都是被允许使用的种子。现在，对$1 \leq i \leq l$，定义

$$s_i = (as_{i-1} + b) \mod M$$

然后定义

$$f(s_0) = (z_1, z_2, \ldots, z_l)$$

其中$z_i = s_i \mod 2$，$1 \leq i \leq l$。因此，我们称f为一个(k,l)线性同余生成器。

引言与实例

例题8.1

假设我们在线性同余器中令$M=31$，$a=3$，$b=5$，这样就构造了$(5,10)$比特生成器。如果考虑映射$s \to 3s+5 \mod 31$，那么有$13 \to 13$，而其余30个剩余数被置换，并构成一个长度为30的圈，

即0，5，20，3，14，16，22，9，1，8，29，30，2，11，7，26，21，6，

23，12，10，4，17，25，18，28，27，24，15，19。当种子选取为13之外的任何数时，那么我们实际上是在这个圈上选定一个初始点，从该点开始的0个数经过模2运算之后就形成了一个伪随机序列。

引言与实例

分组密码的输出反馈模式被可以看做是一个比特生成器。

引言与实例

分组密码的输出反馈模式被可以看做是一个比特生成器。
将几个LFSR结合在一起，使其输出看起来像非线性的。

引言与实例

在计算假设下的被证明安全的比特生成器，介绍基于整数分解的（习题中有基于离散对数的）。

引言与实例

在计算假设下的被证明安全的比特生成器，介绍基于整数分解的（习题中有基于离散对数的）。

算法8.2 RSA生成器

设p,q为两个$k/2$比特长的素数，定义$n=pq$。选择b，使其满足关系式$gcd(b,\phi(n))=1$。n和b是公开的，p和q是保密的。

在Z_n^*中选择一个k比特元素s_0作为种子。对$i\geq 1$，定义

$$s_{i+1}=s_i^b \mod n$$

然后定义

$$f(s_0)=(z_1,z_2,\ldots,z_l)$$

这里，对$1\leq i\leq l$，有$z_i=s_i \mod 2$

因此称f为一个$(k,l)-$RSA生成器。

引言与实例

例题8.2 设$n = 91261 = 263 \times 347$，$b = 1547$，$s_0 = 75634$。则从种子$s_0$产生的序列为10000111011110011000

概率分布的不可区分性

伪随机数生成器的目标：1：速度快，2：应当安全。

概率分布的不可区分性

伪随机数生成器的目标：1：速度快，2：应当安全。
这两个目标是相互矛盾的。

概率分布的不可区分性

伪随机数生成器的目标：1：速度快，2：应当安全。
这两个目标是相互矛盾的。
什么样的生成器才是“安全”的？

概率分布的不可区分性

伪随机数生成器的目标：1：速度快，2：应当安全。

这两个目标是相互矛盾的。

什么样的生成器才是“安全”的？直观地说，由一个比特器产生的长为l的比特串应该看起来“随机”，即应该不能在k或l的多项式时间内把由PRGB产生的长为l的比特串与真正随机的长为l的比特串区分开来。

概率分布的不可区分性

例如，如果一个比特生成器以2/3的概率产生1，那么就很容易把该比特生成器产生的比特串和一个真正随机的比特串区分开来。

概率分布的不可区分性

例如，如果一个比特生成器以2/3的概率产生1，那么就很容易把该比特生成器产生的比特串和一个真正随机的比特串区分开来。

概率分布的可区分性质。我们假设$z^i = (z_1, z_2, ..., z_i)$。

概率分布的不可区分性

定义8.2 设p_0和p_1是长度为l的所有比特串之集$(\mathbb{Z}_2)^l$上的两个概率分布。对$j=0,1$和$z^l \in (\mathbb{Z}_2)^l$，$p_j(z^l)$表示比特串z^l在分布p_j下出现的概率。
设$dst : (\mathbb{Z}_2)^l \to \{0,1\}$是一个函数，$\varepsilon > 0$。对$j=0,1$，定义

$$E_{dst}(p_j) = \sum_{\{z^l \in (\mathbb{Z}_2)^l : dst(z^l)=1\}} p_j(z^l)$$

我们称dst为一个p_0和p_1的ε区分器，如果

$$E_{dst(p_0)} - E_{dst(p_1)} \geq \varepsilon$$

称p_0和p_1的ε可区分的，如果存在这样一个p_0和p_1的ε区分器。称dst为多项式时间区分器，如果$dst(z^l)$可以在l的多项式时间内计算出来。

概率分布的不可区分性

上述区分器的直观意义如下。函数或算法dst力图决定一个给定的l长比特串z^l更可能是按p_0和p_1中的哪一个分布产生的。$dst(z^l)$的结果表示区分器猜测p_0和p_1哪个更可能产生z^l。$E_{dst}(p_j)$的值表示dst在两个概率分布p_0和p_1上输出的平均值（期望值）。如果$E_{dst}(p_0)$和$E_{dst}(p_1)$这两个期望值至少距离ε较远，那么就说dst是一个ε区分器。

概率分布的不可区分性

区分器和PRGB的相关性如下。假设一个由比特生成器产生的l比特序列。有2^l个可能的l比特序列，如果这l比特是独立且均匀随机选择的话，那么这2^l个序列中的每一个将以等概率$1/2^l$的机会发生。这是一个真正的随机分布。我们用p_u来表示这个均匀概率分布。

概率分布的不可区分性

区分器和PRGB的相关性如下。假设一个由比特生成器产生的l比特序列。有2^l个可能的l比特序列，如果这l比特是独立且均匀随机选择的话，那么这2^l个序列中的每一个将以等概率$1/2^l$的机会发生。这是一个真正的随机分布。我们用p_u来表示这个均匀概率分布。

现在考虑由比特生成器f产生的序列，用p_f来表示这个均匀概率分布。如果不同的种子产生不同的序列，则在2^l个可能的序列中，2^k个序列每个以概率$1/2^k$发生，而其他的不会发生。因此概率分布p_f是非常不均匀地。

概率分布的不可区分性

即使两个概率分布p_u和p_f是相当不同的，但仍仅有可能仅对小的ε值，它们是多项式时间ε可区分的。这正是我们构造PRGB时追求的目标，然而，这个目标可能很难达到。比如，通过例子8.3发现，即使以相等的概率来产生0和1，这也不足以保证不可区分性。

下一比特预测器

比特预测器是研究比特生成器中的一个重要概念。工作方式如下：设f是一个(k,l)比特生成器。假定对$1 \leq i \leq l-1$，函数$nbp : (\mathbb{Z}_2)^{i-1} \to \mathbb{Z}_2$以$z^{i-1} = (z_1, ..., z_{i-1})$为输入，$z^{i-1}$表示$f$产生的前$i-1$个比特。现在，函数$nbp$力图预测$f$产生的下一个比特$z_i$。

我们说nbp是一个ε的第i比特预测器，如果给定前$i-1$个比特，nbp能够至少以概率$1/2+\varepsilon(\varepsilon > 0)$来预测所产生伪随机序列的第$i$个比特。

下一比特预测器

假设$0 \leq i \leq l$，我们将第i个比特看做一个随机变量，并记为z_i。

定理8.1 设f是一个(k,l)比特生成器，那么函数nbp是一个关于f的ε的第i比特预测器当且仅当

$$\sum_{z^{i-1} \in (\mathbb{Z}_2)^{i-1}} (p_f(z^{i-1}) \times Pr[z_i = nbp(z^{i-1}) | z^{i-1}]) \geq 1/2 + \varepsilon$$

下一比特预测器

在这个定义中，使用表达式$1/2+\varepsilon$的原因是任何一个预测算法都只能以概率1/2来预测完全随机序列中的任何一个比特。如果一个序列不是完全随机的，那么就有可能以高一些的概率来预测一个给定比特。

下一比特预测器

在这个定义中，使用表达式$1/2+\varepsilon$的原因是任何一个预测算法都只能以概率1/2来预测完全随机序列中的任何一个比特。如果一个序列不是完全随机的，那么就有可能以高一些的概率来预测一个给定比特。

例8.1 对任意的$i: 1 \leq i \leq 9$，定义一个第i比特预测器如下

$$nbp(z^{i-1}) = 1 - z_{i-1}$$

即函数nbp预测在1后面更有可能是0，反之亦然。不难由表8.1计算出，对任意的$i \geq 1$，nbp是一个9/62下一比特预测器。

下一比特预测器

我们可以使用下一比特预测器来构造一个区分算法。假设对一个给定的整数$i \leq l$，nbp是一个ε的第i比特预测器。

算法8.3 Distinguish(z^i)

external nbp

$z \leftarrow nbp(z^{i-1})$

if $z = z_i$

then return(1)

else return(0)

下一比特预测器

定理8.2 假设nbp对(k,l)比特生成器f来说是一个多项式时间ε的第i比特预测器。设p_f是由f导出的$(\mathbb{Z}_2)^i$上的概率分布，p_u是$(\mathbb{Z}_2)^i$上的均匀分布。那么算法8.3是一个p_f和p_u的多项式时间ε区分器。

下一比特预测器

Yao提出的伪随机比特生成器理论中的一个重要结果是：下一比特生成器是一个通用测试，亦即，一个比特生成器是“安全的”，当且仅当除对非常小的ε 值外，不存在该生成器的任何多项式时间ε的第i比特预测器。

下一比特预测器

定理8.2证明了一个方向上的结论。为了证明其逆，必须证明区分器的存在怎样意味着某个第i比特预测器的存在。

下一比特预测器

定理8.2证明了一个方向上的结论。为了证明其逆，必须证明区分器的存在怎样意味着某个第i比特预测器的存在。

定理8.3 假设dst是一个p_f和p_u的多项式时间ε区分器，这里p_f是由(k,l)比特生成器f导出的$(\mathbb{Z}_2)^l$上的概率分布，p_u是$(\mathbb{Z}_2)^l$上的均匀概率分布，那么对某一i，$1 \leq i \leq l-1$，存在关于f的一个多项式时间ε/l的第i比特预测器。

下一比特预测器

我们构造一个第i比特的预测器。

算法8.4 NBP(z^{i-1})

external dst

随机选择$z_i,\ldots,z_l) \in (\mathbb{Z}_2)^{l-i+1}$

$z \leftarrow dst(z_1,\ldots,z_l)$

return$(z+z_i \mod 2)$

BLUM-BLUM-SHUB生成器

这一节，我们将描述并分析一个由Blum，Blum和Shub提出的最流行的PRGB。对任意的奇数n，记模n的二次剩余为QR(n)，即$QR(n) = \{x^2 \bmod n : x \in \mathbb{Z}_n^*\}$。

BLUM-BLUM-SHUB生成器

算法8.5 Blum-Blum-Shub生成器

设p,q为两个满足$p \equiv q \equiv 4(\mod 4$的$k/2$比特素数，定义$n = pq$。$QR(n)$表示模$n$的二次剩余的集合。

一个种子s_0是$QR(n)$中的任何一个元素。对$0 \leq i \leq l-1$，定义

$$s_{i+1} = s_i^2 \mod n$$

然后定义

$$f(s_0) = (z_1, z_2, \ldots, z_l)$$

这里，对$1 \leq i \leq l$，有$z_i = s_i \mod 2$

因此称f为一个$(k,l)-$PRGB，称为Blum-Blum-Shub生成器，简称BBS生成器。

BLUM-BLUM-SHUB生成器

例8.4 假设$n = 192649 = 383 \times 503$，$s_0 = 101355^2 \mod n = 20749$。根据BBS生成器，我们得到前20个比特为

$$11001110000100111010$$

BLUM-BLUM-SHUB生成器

假设p和q是两个不同素数，令$n = pq$。由Jacobi符号的定义，很容易得到

$$\left(\frac{x}{n}\right) = \begin{cases} 0,\ if\ gcd(x,n) > 1 \\ 1,\ if\ \left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = 1\ or\ \left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = -1 \\ -1,\ if\ \left(\frac{x}{p}\right) = -\left(\frac{x}{q}\right) \end{cases}$$

BLUM-BLUM-SHUB生成器

定义

$$\widetilde{QR}(n) = \{x \in \mathbb{Z}_n^* \setminus QR(n) : (\frac{x}{n}) = 1\}$$

这样

$$\widetilde{QR}(n) = \{x \in \mathbb{Z}_n^* : (\frac{x}{p}) = (\frac{x}{q}) = -1\}$$

元素$x \in \widetilde{QR}(n)$称为模n的一个伪平方。

Blum-Blum-Shub生成器

Blum-Blum-Shub生成器的安全性基于复合二次剩余问题的难处理性。

问题8.1 复合二次剩余

实例：正整数n是两个未知不同奇素数p和q之积，整数$x \in \mathbb{Z}_n^*$满足$\left(\frac{x}{n}\right) = 1$。

问题：$x \in QR(n)$吗？

BLUM-BLUM-SHUB生成器

Blum-Blum-Shub生成器的安全性基于复合二次剩余问题的难处理性。

问题8.1 复合二次剩余

实例：正整数n是两个未知不同奇素数p和q之积，整数$x \in \mathbb{Z}_n^*$满足$\left(\frac{x}{n}\right) = 1$。

问题：$x \in QR(n)$吗？

复合二次剩余问题需要我们能够区别模n的二次剩余和模n的伪二次剩余。这不比分解n更困难。

BLUM-BLUM-SHUB生成器

对于BBS生成器，下列特征是重要的。

BLUM-BLUM-SHUB生成器

对于BBS生成器，下列特征是重要的。

因为$n = pq$，$p \equiv q \equiv 3 \mod 4$，因此，对任意的二次剩余$x$，有唯一一个$x$的平方根，它也是二次剩余，这个平方根称为$x$的主平方根。

BLUM-BLUM-SHUB生成器

对于BBS生成器，下列特征是重要的。

因为$n = pq$，$p \equiv q \equiv 3 \mod 4$，因此，对任意的二次剩余$x$，有唯一一个$x$的平方根，它也是二次剩余，这个平方根称为$x$的主平方根。

因此，用于定义BBS生成器的映射$x \to x^2 \mod n$是$QR(n)$中的一个置换，即模n的二次剩余集的置换。

BLUM-BLUM-SHUB生成器

例8.5 假设$n = 253 = 11 \times 23$，则

$$|QR(n)| = \frac{10 \times 22}{4} = 55$$

计算表明$\mathbb{Z}_{55}$中的元素：该置换有一个长度为1的圈，一个长度为4的圈，一个长度为10的圈和两个长度为20的圈。

BBS生成器的安全性

假设BBS生成器所产生的伪随机比特与l个随机比特是ε可区分的。

BBS生成器的安全性

假设BBS生成器所产生的伪随机比特与l个随机比特是ε可区分的。

前一比特预测器。

BBS生成器的安全性

假设BBS生成器所产生的伪随机比特与l个随机比特是ε可区分的。

前一比特预测器。

定理8.4 假设存在一个p_f和p_u的多项式时间ε区分器，这里p_f是(k,l)-BBS生成器f导出的$(\mathbb{Z}_2)^l$上的概率分布，p_u 是$(\mathbb{Z}_2)^l$上的均匀概率分布。那么存在一个关于f的（多项式时间）$(\frac{\varepsilon}{l})$前一比特预测器。

BBS生成器的安全性

下面我们使用δ前一比特预测器pbp来构造一个概率算法，把模n二次剩余与模n伪二次剩余以$1/2+\delta$的概率区分开来。

BBS生成器的安全性

算法8.6 QR-TEST(x,n)

external pbp

$s_1 \leftarrow x^2 \mod n$

comment:s_1是一个模n的二次剩余

$z_1 \leftarrow s_1 \mod 2$

由种子s_1使用BBS生成器计算出$z_2, \ldots, z_l$

$z \leftarrow pbp(z_1, \ldots, z_l)$

如果 $(x \mod 2) = z$

　那么return(yes)

　否则return(no)

BBS生成器的安全性

定理8.5 假设pbp是关于(k,l)-BBS生成器f的一个δ前一比特预测器，那么算法QR-TEST至少以$1/2+\delta$的概率在多项式时间内正确地确定二次剩余性，这里的概率计算是在所有可能的输入$x \in QR(n) \bigcup \widetilde{QR}(n)$上求平均，而这些输入是均匀随机选择的。

BBS生成器的安全性

定理8.5表明，我们如何以至少$1/2+\delta$的概率将二次剩余和伪平方区别开来，这个结果可以被改进为至少以$1/2+\delta$正确确定出二次剩余性的Monte Carlo算法。

BBS生成器的安全性

定理8.5表明，我们如何以至少$1/2+\delta$的概率将二次剩余和伪平方区别开来，这个结果可以被改进为至少以$1/2+\delta$正确确定出二次剩余性的Monte Carlo算法。

算法8.7 MC-QR-TEST(x)

external QR-TEST

随机选择$r \in \mathbb{Z}_n^*$

$x' \leftarrow r^2 x \mod n$

随机选择$s \in \{0,1\}$

$x' \leftarrow sx' \mod n$

$t \leftarrow QR-TEST(x')$

如果((t=yes)和(s=1))或((t=no)和(s=-1))

　那么返回“yes”，否则返回”no”

BBS生成器的安全性

定理8.6 假设QR-TEST以至少$1/2+\delta$的平均概率在多项式时间内正确确定二次剩余性，那么算法8.7中描述的算法MC-QR-TEST是一个关于复合二次剩余的多项式时间Monte Carlo算法，其错误概率至多为$1/2-\delta$。

BBS生成器的安全性

介绍一个算法属于无偏差Monte Carlo算法，它表明，对任意的$\gamma > 0$，任何一个错误概率至多为$1/2 - \delta$的无偏差Monte Carlo算法都能用于构造一个错误概率至多为γ的无偏差Monte Carlo算法。其思想是对某一整数m，运行给定的Monte Carlo算法2m+1次，并按“择多选择”做出一个回答。

BBS生成器的安全性

介绍一个算法属于无偏差Monte Carlo算法，它表明，对任意的$\gamma > 0$，任何一个错误概率至多为$1/2 - \delta$的无偏差Monte Carlo算法都能用于构造一个错误概率至多为γ的无偏差Monte Carlo算法。其思想是对某一整数m，运行给定的Monte Carlo算法2m+1次，并按“择多选择”做出一个回答。

定理8.7 假设A是一个错误概率至多为$1/2 - \delta$的无偏差Monte Carlo算法。定义一个算法A^n，这个算法对一个给定的实例I运行A $n = 2m + 1$次，并输出最常出现的回答。那么，算法A^n的错误概率至多为

$$\frac{(1 - 4(\delta)^2)^m}{2}$$

BBS生成器的安全性

小结：

BBS生成器的安全性

小结：

- (k,l)-BBS生成器与l个随机比特是ε可区分的;

BBS生成器的安全性

小结：

- (k,l)-BBS生成器与l个随机比特是ε可区分的;
- 对(k,l)-BBS生成器的(ε/l)前一比特预测器；

BBS生成器的安全性

小结：

- (k,l)-BBS生成器与l个随机比特是ε可区分的;
- 对(k,l)-BBS生成器的(ε/l)前一比特预测器；
- 正确概率至少为$1/2+\varepsilon/l$的关于复合二次剩余的区分算法；

BBS生成器的安全性

小结：

- (k,l)-BBS生成器与l个随机比特是ε可区分的;
- 对(k,l)-BBS生成器的(ε/l)前一比特预测器；
- 正确概率至少为$1/2+\varepsilon/l$的关于复合二次剩余的区分算法；
- 关于复合二次剩余的错误概率至多为$1/2-\varepsilon/l$的无偏差Monte Carlo算法；

BBS生成器的安全性

小结：

- (k,l)-BBS生成器与l个随机比特是ε可区分的;
- 对(k,l)-BBS生成器的(ε/l)前一比特预测器；
- 正确概率至少为$1/2+\varepsilon/l$的关于复合二次剩余的区分算法；
- 关于复合二次剩余的错误概率至多为$1/2-\varepsilon/l$的无偏差Monte Carlo算法；
- 对任一$\gamma>0$，关于复合二次剩余的错误概率至多为γ的无偏差Monte Carlo算法。

BBS生成器的安全性

小结：

- (k,l)-BBS生成器与l个随机比特是ε可区分的;
- 对(k,l)-BBS生成器的(ε/l)前一比特预测器；
- 正确概率至少为$1/2+\varepsilon/l$的关于复合二次剩余的区分算法；
- 关于复合二次剩余的错误概率至多为$1/2-\varepsilon/l$的无偏差Monte Carlo算法；
- 对任一$\gamma>0$，关于复合二次剩余的错误概率至多为γ的无偏差Monte Carlo算法。

所以,BBS生成器是安全的，这就是一个可证明安全性的例子。

BBS生成器的安全性

BBS生成器的改进：

BBS生成器的安全性

BBS生成器的改进：

在BBS中，伪随机比特序列是取每一个s_i的最低比特来构造的。现在假设我们从每一个s_i中抽取r个最低比特，这里r是一个正整数。这将把PRGB的效率提高r倍，但PRGB是否仍保持安全性？

BBS生成器的安全性

BBS生成器的改进：

在BBS中，伪随机比特序列是取每一个s_i的最低比特来构造的。现在假设我们从每一个s_i中抽取r个最低比特，这里r是一个正整数。这将把PRGB的效率提高r倍，但PRGB是否仍保持安全性？

现在已经证明，如果$r \leq lb(lbn)$条件下，这个方法仍然是安全的。

概率加密

定义8.3 一个概率公钥密码体制定义为一个6元组$(\mathscr{P},\mathscr{C},\mathscr{K},\mathscr{E},\mathscr{D},\mathscr{R})$，其中$\mathscr{P}$是明文集，$\mathscr{C}$是密文集，$\mathscr{K}$是密钥空间，$\mathscr{R}$是随机化种子的集合。对每一个密钥$K\in\mathscr{K}$，$e_K\in\mathscr{E}$是一个公开加密规则，$d_K\in D$是一个秘密解密规则。同时，满足下列特性：

概率加密

定义8.3 一个概率公钥密码体制定义为一个6元组$(\mathscr{P},\mathscr{C},\mathscr{K},\mathscr{E},\mathscr{D},\mathscr{R})$，其中$\mathscr{P}$是明文集，$\mathscr{C}$是密文集，$\mathscr{K}$是密钥空间，$\mathscr{R}$是随机化种子的集合。对每一个密钥$K\in\mathscr{K}$，$e_K\in\mathscr{E}$是一个公开加密规则，$d_K\in D$是一个秘密解密规则。同时，满足下列特性：

1. 每一个$e_K:\mathscr{P}\times\mathscr{R}\to\mathscr{C}$和$d_K:\mathscr{C}\to\mathscr{P}$是满足

$$d_K(e_K(b,r))=b$$

的函数，对每一个明文$b\in\mathscr{P}$和每一个$r\in\mathscr{R}$[特别地，它意味着如果$x\neq x'$，那么$e_K(x,r)\neq e_K(x',r)$]

概率加密

定义8.3（续）2. 该体制的安全性定义如下。设ε是一个指定的安全参数。对任意固定的$K \in \mathscr{K}$和任意的$x \in \mathscr{P}$，定义一个C上的概率分布$p_{K,x}$，这里$p_{K,x}(y)$表示给定K是密钥，x是明文时，y是密文的概率。假设$x, x' \in \mathscr{P}$，$x \neq x'$，$K \in \mathscr{K}$，那么概率分布$p_{K,x}$和$p_{K,x'}$不是多项式时间ε可区分的。

概率加密

定义8.3（续）2. 该体制的安全性定义如下。设ε是一个指定的安全参数。对任意固定的$K \in \mathscr{K}$和任意的$x \in \mathscr{P}$，定义一个C上的概率分布$p_{K,x}$，这里$p_{K,x}(y)$表示给定K是密钥，x是明文时，y是密文的概率。假设$x, x' \in \mathscr{P}$，$x \neq x'$，$K \in \mathscr{K}$，那么概率分布$p_{K,x}$和$p_{K,x'}$不是多项式时间ε可区分的。

很明显，一个如上定义的概率公钥密码体制能够提供语义安全性。

概率加密

我们提出一个称为Goldwasser-Micali公钥密码体制作为密码体制8.1。

概率加密

我们提出一个称为Goldwasser-Micali公钥密码体制作为密码体制8.1。

这个密码体制每次加密一个比特。0比特被加密成一个模n的随机二次剩余，1比特被加密成一个模n的随机伪二次剩余。

概率加密

我们提出一个称为Goldwasser-Micali公钥密码体制作为密码体制8.1。

这个密码体制每次加密一个比特。0比特被加密成一个模n的随机二次剩余，1比特被加密成一个模n的随机伪二次剩余。

当Bob接收到元素$y \in QR(n) \bigcup \widetilde{QR}(n)$时，他能使用关于$n$分解的知识来确定是否$y \in QR(n)$还是$y \in \widetilde{QR}(n)$。他通过计算

$$\left(\frac{y}{p}\right) = y^{(p-1)/2} \mod p$$

来完成，那么

$$y \in QR(n) \Leftrightarrow \left(\frac{y}{p}\right) = 1$$

概率加密

密码体制8.1 Goldwasser-Micali公钥密码体制

设$n = pq$，其中p和q是不同的奇素数。设$m \in \widetilde{QR}(n)$，整数n和m是公开的，$n = pq$的分解是保密的。设$\mathscr{P} = \{0,1\}$，$\mathscr{C} = \mathscr{R} = \mathbb{Z}_n^*$，定义$\mathscr{K} = \{(n,p,q,m)\}$，其中$n,p,q$和$m$如上定义。

对$K = (n,p,q,m)$，定义

$$e_K(x,r) = m^x r^2 \mod n$$

和

$$d_k(y) = \begin{cases} 0, y \in QR(n) \\ 1, y \in \widetilde{QR}(n) \end{cases}$$

概率加密

Goldwasser-Micali公钥密码体制具有非常高的数据扩展。

概率加密

Goldwasser-Micali公钥密码体制具有非常高的数据扩展。

Blum与Goldwasser给出了一个更有效的概率公钥密码体制（从数据扩展的角度）。

概率加密

Goldwasser-Micali公钥密码体制具有非常高的数据扩展。

Blum与Goldwasser给出了一个更有效的概率公钥密码体制（从数据扩展的角度）。

Blum-Goldwasser公钥密码体制是一种公钥流密码。

概率加密

Blum-Goldwasser公钥密码体制的基本思想：

概率加密

Blum-Goldwasser公钥密码体制的基本思想：一个随机种子s_0利用BBS生成器产生l个伪随机比特$z_1,\ldots,z_l$，然后将z_i用做密钥流，即把它们与l长的明文比特异或形成密文。同时，第$(l+1)$个元素$s_{l+1}=s_0^{2^{l+1}} \mod n$作为密文的一部分进行传送。

概率加密

Blum-Goldwasser公钥密码体制的基本思想：一个随机种子s_0利用BBS生成器产生l个伪随机比特$z_1,\ldots,z_l$，然后将z_i用做密钥流，即把它们与l长的明文比特异或形成密文。同时，第$(l+1)$个元素$s_{l+1}=s_0^{2^{l+1}} \mod n$作为密文的一部分进行传送。

当Bob接收密文时，他能从s_{l+1}中计算出s_0；然后重构出密钥流，最后把密钥流与l个密文比特进行异或得到明文。

概率加密

Blum-Goldwasser公钥密码体制的基本思想：一个随机种子s_0利用BBS生成器产生l个伪随机比特$z_1,\ldots,z_l$，然后将z_i用做密钥流，即把它们与l长的明文比特异或形成密文。同时，第$(l+1)$个元素$s_{l+1}=s_0^{2^{l+1}} \mod n$作为密文的一部分进行传送。

当Bob接收密文时，他能从s_{l+1}中计算出s_0；然后重构出密钥流，最后把密钥流与l个密文比特进行异或得到明文。

一个问题：Bob怎样从s_{l+1}得到s_0？

概率加密

密码体制8.2 Blum-Goldwasser公钥密码体制

设$n = pq$，其中p,q是素数，且$p \equiv q \equiv 3 \pmod 4$。整数$n$是公开的，$n = pq$的分解是保密的。设$\mathscr{P} = (\mathbb{Z}_2)^l$，$\mathscr{C} = (\mathbb{Z}_2)^l \times \mathbb{Z}_n^*$，$\mathscr{R} = \mathbb{Z}_n^*$。定义$\mathscr{K} = \{(n,p,q)\}$，其中$n,p$和$q$如上定义。对$K = (n,p,q)$，$x \in (\mathbb{Z}_2)^l$，$r \in \mathbb{Z}_n^*$，加密$x$如下：

1. 使用BBS生成器从种子$s_0 = r$计算出$z_1,\ldots,z_l$。
2. 计算出$s_{l+1} = s_0^{2^{l+1}} \mod n$。
3. 对$1 \le i \le l$计算出$y_i = (x_i + z_i) \mod 2$。
4. 定义$e_K(x,r) = (y_1,\ldots,y_l,s_{l+1})$。

概率加密

密码体制8.2（续）

为了解密y，Bob完成下列步骤：

1. 计算出$a_1 = ((p+1)/4)^{l+1} \mod (p-1)$。
2. 计算出$a_2 = ((q+1)/4)^{l+1} \mod (q-1)$。
3. 计算出$b_1 = s_{l+1}^{a_1} \mod p$。
4. 计算出$b_2 = s_{l+1}^{a_2} \mod q$。
5. 使用中国剩余定理找到r满足

$$r \equiv b_1 (\mod p),\ r \equiv b_2 (\mod q)$$

6. 利用BBS生成器从种子$s_0 = r$计算出$z_1, \ldots, z_l$。
7. 对$1 \le i \le l$计算出$x_i = (y_i + z_i) \mod 2$。
8. 明文$x = (x_1, \ldots, x_l)$。

概率加密

例题8.8

概率加密

例题8.8

Blum-Goldwasser公钥密码体制的数据扩展还算合理。

概率加密

例题8.8

Blum-Goldwasser公钥密码体制的数据扩展还算合理。

与Goldwasser-Micali公钥密码体制相比，这个体制在数据扩展方面是一个巨大的进步。然而，似乎并不存在一个具体的分析可对一个给定的模尺寸（比如k），指出l取多大才是安全的。虽然在我们的计算假设下，$l = k^2$是渐进安全的，但对固定的模尺寸，比如$n \approx 2^{1024}$，还不知道这个体制是否安全。