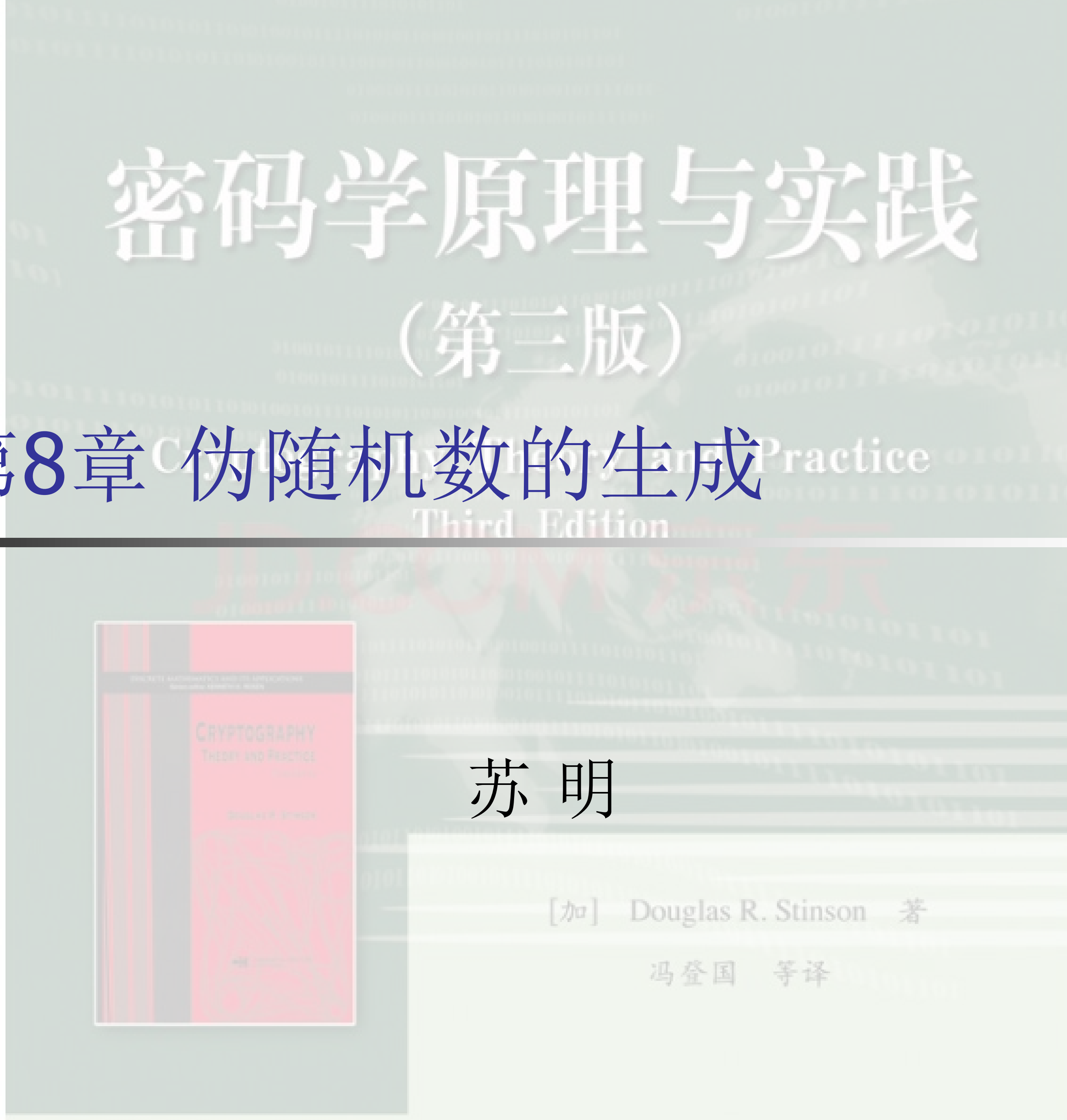
第8章 伪随机数的生成



苏 明

概览

- 8．1 引言与示例
- 8．2 概率分布的不可区分性
- 8．3 Blum-Blum-Shub生成器
- 8．4 概率加密

8．1 引言与示例

- 密码学很多场景需要随机数：公平、安全
- 投掷硬币、物理过程产生随机数费时且昂贵
- 实用： 使用一个**伪随机比特生成器(PRBG)**来产生随机数

8．1 引言与示例

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

8．1 引言与示例

- 完善保密性：**一次一密**

- 挑战？

- 密钥量与明文一样长
- 因此实用中采用了PRBG
- 看起来随机、*不可预测*

8．1 引言与示例

- 随机数生成器还应用于：模拟，Monte Carlo算法、采样、测试，…

如何衡量一个PRG的随机特征？

- 频率、游程、…等指标
- 经过统计测试，比如χ^2测试

8．1 引言与示例

- 一个k阶LFSR(比如m序列)是否安全？

- Berlekamp-Massey Algorithm

8．1 引言与示例

线性同余发生器：See Knuth, Vol. 2

算法 8.1　线性同余生成器

设 $M \geq 2$ 是一个整数，$1 \leqslant a,b \leqslant M-1$。定义 $k = 1 + \lfloor \mathrm{lb}\, M \rfloor$，并令 $k+1 \leqslant \ell \leqslant M-1$。

种子是一个整数 s_0，这里 $0 \leqslant s_0 \leqslant M-1$。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 $1 \leqslant i \leqslant \ell$，定义

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

然后，定义

$$f(s_0) = (z_1, z_2, \cdots, z_\ell)$$

其中 $z_i = s_i \bmod 2$，$1 \leqslant i \leqslant \ell$。

因此，我们称 f 为一个 (k,ℓ) 线性同余生成器。

8．1 引言与示例

- 例子：
- M=31, a=3, b=5

- 构造的(5,10)PRBG具有什么性质？(Z_{31}上周期)

8．1 引言与示例

种 子	序 列	种 子	序 列
0	1010001101	16	0110100110
1	0100110101	17	1001011010
2	1101010001	18	0101101010
3	0001101001	19	0101000110
4	1100101101	20	1000110100
5	0100011010	21	0100011001
6	1000110010	22	1101001101
7	0101000110	23	0001100101
8	1001101010	24	1101010001
9	1010011010	25	0010110101
10	0110010110	26	1010001100
11	1010100011	27	0110101000
12	0011001011	28	1011010100
13	1111111111	29	0011010100
14	0011010011	30	0110101000
15	1010100011		

8．1 引言与示例

RSA 生成器

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 $\mathbb{Z}_n^*$ 中选择一个 k 比特元素 s_0 作为种子。对 $i \geqslant 1$，定义

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

然后定义

$$f(s_0) = (z_1, z_2, \cdots, z_\ell)$$

这里，对 $1 \leqslant i \leqslant \ell$，有

$$z_i = s_i \bmod 2$$

因此称 f 为一个 (k, ℓ)-RSA 生成器。

8．2 概率分布的不可区分性

- 伪随机数发生器：设计要求

✓ 快速

✓ 安全

安全：不能在k(或者l)的多项式时间内，把由*PRBG*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

8．2 概率分布的不可区分性

例子：

- 真随机序列 VS 2/3概率产生1的PRBG

- 如何区分开来？

8．2 概率分布的不可区分性

- 概率分布的可区分性

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

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

我们称 dst 为一个 p_0 和 p_1 的 ϵ 区分器，如果

$$|E_{\text{dst}}(p_0)-E_{\text{dst}}(p_1)|\geqslant\epsilon$$

称 p_0 和 p_1 是 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 $\text{dst}(z^\ell)$ 可以在 ℓ 的多项式时间内计算出来。

8．2 概率分布的不可区分性

- 推广到随机算法情形

对比特串 $z^\ell=(z_1,\cdots,z_\ell)$，一个区分器以概率 p(依赖于 z^ℓ)猜测成 0，因此以概率 $1-p$ 猜测成 1。在随机区分器的情况下，不难看出

$$E_{\mathrm{dst}}(p_j)=\sum_{z^\ell\in(\mathbf{Z}_2)^\ell}\left(p_j(z^\ell)\times\Pr[\mathrm{dst}(z^\ell)=1]\right)$$

我们得到和证明的所有结果对随机区分器也同样成立。

8．2 概率分布的不可区分性

- p_u: 均匀概率分布
- p_f: 由f产生的序列的概率分布

- 假设PRBG f产生的序列是平衡的， 即
- $l/2$个比特为0， $l/2$个比特为1
- 请构造区分器(dst), 区分f和 真随机序列？

8．2 概率分布的不可区分性

$$\mathrm{dst}(z_1,\cdots,z_\ell)=\begin{cases}1 & \text{如果 } (z_1,\cdots,z_\ell) \text{恰有} \ell/2 \text{ 个比特为 } 0 \\ 0 & \text{其他}\end{cases}$$

$$E_{\mathrm{dst}}(p_u)=\frac{\binom{\ell}{\ell/2}}{2^\ell}$$

$$E_{\mathrm{dst}}(p_f)=1$$

$$\lim_{\ell\to\infty}\frac{\binom{\ell}{\ell/2}}{2^\ell}=0$$

因此，对任意固定的 $\epsilon<1$，如果 ℓ 是充分大的，那么 p_u 和 p_f 是 ϵ 可区分的。

8.2.1 下一比特预测器

- nbp(Next Bit Predictor)

如果给定前i-1个比特，nbp能够至少以概率$\frac{1}{2}+\epsilon(\epsilon>0)$来预测所产生伪随机序列的第i个比特。

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

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

8.2.1 下一比特预测器

- nbp→Distinguish

算法 8.3　Distinguish (z^i)

external　　nbp

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

if　$z = z_i$

　then return (1)

　else return (0)

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。

8.2.1 下一比特预测器

- Distinguish → nbp

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 $(\mathbb{Z}_2)^\ell$ 上的概率分布，p_u 是 $(\mathbb{Z}_2)^\ell$ 上的均匀概率分布，那么对某一 i，$1 \leqslant i \leqslant \ell - 1$，存在关于 f 的一个(多项式时间) ϵ/ℓ 的第 i 比特预测器。

8.3 Blum-Blum-Shub生成器

算法 8.5 Blum-Blum-Shub 生成器

设 p，q 是两个满足 $p \equiv q \equiv 3 \bmod 4$ 的 $(k/2)$ 比特素数，定义 $n = pq$。$\mathrm{QR}(n)$ 表示模 n 的二次剩余的集合。

一个种子 s_0 是 $\mathrm{QR}(n)$ 中的任何一个元素。对 $0 \leqslant i \leqslant \ell - 1$，定义

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

然后定义

$$f(s_0) = (z_1, z_2, \cdots, z_\ell)$$

其中

$$z_i = s_i \bmod 2$$

$1 \leqslant i \leqslant \ell$。那么 f 是一个 (k, ℓ) PRBG，称为 Blum-Blum-Shub 生成器，简写为 BBS 生成器。

一种选择合适种子的方法是先选择一个 $s_{-1} \in \mathbb{Z}_n^*$，然后计算 $s_0 = s_{-1}^2 \bmod n$。这保证了 $s_0 \in \mathrm{QR}(n)$。

8.3 Blum-Blum-Shub生成器

x 是模 n 的二次剩余当且仅当

$$\left(\frac{x}{p}\right)=\left(\frac{x}{q}\right)=1$$

定义

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

这样

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

$$|\mathrm{QR}(n)|=|\widetilde{\mathrm{QR}}(n)|=?$$

8.3.1 Blum-Blum-Shub生成器安全性

- 复合二次剩余(Composite Quadratic Residues)

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：$x \in \mathrm{QR}(n)$ 吗？

- 本质上需要区别模n的二次剩余、伪二次剩余
- 通常猜测：若分解n不可行，那么该问题难解

8.3.1 Blum-Blum-Shub生成器安全性

- pbp是一个ϵ**前一比特预测器**：若它正确猜测z_0的概率至少为1/2+ ϵ（对所有可能种子s_0）
- 利用δ前一比特预测器**构造**概率算法，把模n二次剩余、伪二次剩余 以1/2+ δ 区分开来

算法 8.6 QR-TEST(x,n)

external pbp

$s_1 \leftarrow x^2 \bmod n$

comment: s_1是一个模n的二次剩余

$z_1 \leftarrow s_1 \bmod 2$

由种子s_1使用 BBS 生成器计算出$z_2,\cdots,z_\ell$

$z \leftarrow \text{pbp}(z_1,\cdots,z_\ell)$

if $(x \bmod 2) = z$

    then return (yes)

    else return (no)

8.3.1 Blum-Blum-Shub生成器安全性

(k,ℓ)-BBS 生成器与 ℓ 个随机比特是 ϵ 可区分的

$\Downarrow$

对 (k,ℓ)-BBS 生成器的 (ϵ/ℓ) 前一比特预测器

$\Downarrow$

正确概率至少为 $1/2+\epsilon/\ell$ 的关于复合二次剩余的区分算法

$\Downarrow$

关于复合二次剩余的错误概率至多为 $1/2-\epsilon/\ell$ 的无偏差 Monte Carlo 算法

$\Downarrow$

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

普遍相信对复合二次剩余问题不存在一个小错误概率的多项式时间Monte Carlo算法

因此相信BBS生成器是安全的

8.4 概率加密

- 密码体制的**语义安全性**

- 密文**不可区分性**

8.4 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($\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D},\mathcal{R}$)，其中 $\mathcal{P}$ 是明文集，$\mathcal{C}$ 是密文集，$\mathcal{K}$ 是密钥空间，$\mathcal{R}$ 是随机化子的集合。对每一个密钥 $K\in\mathcal{K}$，$e_K\in\mathcal{E}$ 是一个公开加密规则，$d_K\in\mathcal{D}$ 是一个秘密解密规则。同时，要满足下列特性：

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

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

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

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 $K\in\mathcal{K}$ 和任意的 $x\in\mathcal{P}$，定义一个 $\mathcal{C}$ 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(这个概率的计算是在所有随机选择的 $r\in\mathcal{R}$ 上进行的)。假设 x，$x'\in\mathcal{P}$，$x\neq x'$，$K\in\mathcal{K}$，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。

8.4 概率加密

- 如果$x \neq x'$, 那么x的所有加密的概率分布与x′的所有加密的概率分布是（多项式时间）不可区分的

8.4 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 $m\in\widetilde{\mathrm{QR}}(n)$[a]，整数 n 和 m 是公开的，$n=pq$ 的分解是保密的。设 $\mathcal{P}=\{0,1\}$，$\mathcal{C}=\mathcal{R}=\mathbb{Z}_n^*$，定义 $\mathcal{K}=\{(n,p,q,m)\}$，其中 n，p，q 和 m 如上定义。

对 $K=(n,p,q,m)$，定义

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

和

$$d_K(y)=\begin{cases}0, & \text{如果} y\in \mathrm{QR}(n)\\ 1, & \text{如果} y\in \widetilde{\mathrm{QR}}(n)\end{cases}$$

此处 $x=0$ 或 1，r 和 $y\in\mathbb{Z}_n^*$。

[a] 如果 $p\equiv 3 \pmod 4$ 且 $q\equiv 3 \pmod 4$，那么，我们可以取 $m=-1$。这将提高加密的效率，这是因为不再需要进行 m^x 的指数运算。

8.4 概率加密

Pros and Cons

- 优点　可证明安全性
- 缺点　密文膨胀大

8.4 概率加密

密码体制 8.2 Blum-Goldwasser 公钥密码体制

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

1. 使用 BBS 生成器从种子 $s_0=r$ 计算出 $z_1,\cdots,z_\ell$。
2. 计算出 $s_{\ell+1}=s_0^{2^{\ell+1}} \bmod n$。
3. 对 $1 \leqslant i \leqslant \ell$ 计算出 $y_i=(x_i+z_i) \bmod 2$。
4. 定义 $e_K(x,r)=(y_1,\cdots,y_\ell,s_{\ell+1})$。

为了解密 y，Bob 完成下列步骤：

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

$$r \equiv b_1 \pmod p \text{ 和 } r \equiv b_2 \pmod q$$

6. 利用 BBS 生成器从种子 $s_0=r$ 计算出 $z_1,\cdots,z_\ell$。
7. 对 $1 \leqslant i \leqslant \ell$ 计算出 $x_i=(y_i+z_i) \bmod 2$。
8. 明文 $x=(x_1,\cdots,x_\ell)$。

8.4 概率加密

$x^{((p+1)/4)^{\ell+1}}$ 将是 x 模 p 的主 $2^{\ell+1}$ 次根

- Blum-Goldwasser公钥密码体制：**公钥流密码**
  - 加密时： s_{l+1}作为密文的一部分进行传输
  - 解密时： 从s_{l+1}计算出s_0, 重构出密钥流

- 数据扩展还算**合理**