*Machine Learning and Artificial Intelligence*
*A.J. Tallón-Ballesteros and C. Chen (Eds.)*

*doi:10.3233/FAIA200807*

# Order-Preserving and Efficient One-to-Many Search on Encrypted Data

Dongping HU, Aihua YIN [1], Huaying YAN, Tao LONG
*School of Software and Internet of Things Engineering, Jiangxi University of Finance and Economics, Nanchang 330023, Jiangxi, China*

**Abstract.** Order-preserving encryption (OPE) is an useful tool in cloud computing as it allows untrustworthy server to execute range query or exact keyword search directly on the ciphertexts. It only requires sub-linear time in the data size while the queries are occurred. This advantage is very suitable in the cloud where the data volume is huge. However, the order-preserving encryption is deterministic and it leaks the plaintexts' order and its distribution. In this paper, we propose an one-to-many OPE by taking into account the security and the efficiency. For a given plaintext, the encryption algorithm firstly determines the corresponding ciphertext gap by performing binary search on ciphertext space and plaintext space at the same time. An exact sample algorithm for negative hypergeometric distribution is used to fix the size of the gap. Lastly a value in the gap is randomly chosen as the mapping of the given plaintext. It is proven that our scheme is more secure than deterministic OPE with realizing efficient search. In particular, a practical and exact sampling algorithm for the negative hypergeometric distribution (NHGD) is first proposed.

**Keywords.** Order-preserving Searchable Encryption, One-to-many Mapping Encryption, Range Query, Negative Hypergeometric Distribution

## 1. Introduction

In cloud computing, the sensitive data are always encrypted before outsourcing for the security concern. It is an effective approach to protect the confidentiality of the data. However, the encryption operation reduces the utilization of data. For example, comparison operations, such as MIN, MAX and range queries, are common operations in encrypted SQL database which requires the server to perform operations over ciphertexts. For the server, one approach is to decrypt the whole database first and then runs the operations for correct results. Obviously, it requires trusting server and the process is time-consuming. Recently, the fully homomorphic encryption shows it can perform compare computations directly on encrypted data. However, its performance cost is extremely high, on the order of $10^9$ times [1].

For tackling this problem, searchable encryption was proposed and it allows the cloud sever (in our work, the cloud server is believed to be an untrustworthy server) to execute queries on ciphertexts. And then Order-preserving Encryption was proposed in [2]. This primitive ensures the order between ciphertexts is as the same as the order between plaintexts. It permits efficient range queries or comparison operations. In the paper, "efficient" means at least in sub-linear time in the database size, for executing

[1] *Corresponding author. E-mail: Aihuayin@jxufe.edu.cn.

linear work (such as homomorphic encryption) is prohibitively slow in real world applications for huge database. Due to this advantage, OPE is prevalent in cloud computing.

Agrawal *et al.* [2] initialized OPE for supporting range queries over ciphertexts. The encryption of the scheme takes all the plaintexts as input before encrypting. In real applications, it is difficult for users to know all plaintexts in advance. Furthermore, the work of [2] did not provide the security definition.

Boldyreva et al. formally defined the security of OPE in [3]. The ideal security of OPE, named IND-OCPA, was first given. Authors then proved that IND-OCPA was unachievable in practical because of huge ciphertexts space required. Instead a 'weaker' security notion POPF-CCA was proposed, which meant that adversary cannot distinguish between OPE and ideal order-preserving function (OPF). Furthermore, a deterministic OPE scheme satisfied POPF-CCA is proposed. Here we called it BCLO scheme.

To satisfy IND-OCPA, Popa et al.[4] proposed a new OPE model by building a balanced tree to store ciphertext values. The tree would be rebalanced after a new ciphertext inserted. A secure client decrypted the ciphertexts first and then returns the ordering after compares the plaintexts. Although Popa's. scheme satisfied the ideal security; it was not a real OPE in nature as the order of plaintexts was not preserved. As discussed in [5,6], the computation cost of Popa's scheme was linear time and not suitable the applications with large data (e.g. cloud computing). Some other OPE schemes have been proposed in [7-10] and they either provided weaker security guarantee or are not efficient.

Recently, Liu et al. [11] focused on how to conceal the distribution values. They inserted many letters into ciphertext space to expand it and then divided it nonlinearly. Then a plaintext value was mapped to a value in the expanded space. It helped enhance security and order comparison could be executed directly over the ciphertexts.

In this work, we improve the Boldyreva's work and design a one-to-many OPE (Otm OPE) that achieves higher security and efficiency.

We organized our paper as follows. Our Otm OPE is introduced in Section 2. An exact sampling algorithm for generating the negative hypergeometric random variables is proposed in Section 3. Security proof is presented in Section 4. We evaluate the performance in Section 5. A conclusion is given at last.

## 2. Our Otm OPE

The BCLO scheme is deterministic and it leaks not only the order but also the plaintext space's distribution. To overcome its weaknesses, we adapt the BCLO scheme by raising its randomness. We execute binary search directly on plaintext space for achieving efficient search. For a plaintext, we first determine its corresponding interval of ciphertext space. This is achieved by employing an exact sample algorithm for NHGD we propose. Finally, a value is randomly chosen from the interval. In this way, same plaintext can be mapped to different ciphertexts with different random seeds. The Notations List of the paper is given in Table 1. We first describe the Algorithm Binary Search in Algorithm 1. And then we present our Otm OPE in Algorithm 2.

**Table 1.** Notations List

| Notation | Descriptions |
|---|---|
| $SK$ | Symmetric Key |
| $x$ | a Plaintext |
| $y$ | a Ciphertext |
| P$S$ | Plaintext Space $\{1,...,\mathrm{P}\}$ |
| C$S$ | Ciphertext Space $\{1,...,\mathrm{C}\}$ |
| $NHGD$ | Negative Hypergeometric distribution |
| $NBD$ | Negative Binomial Distribution |
| $Gamma$ | Gamma Distribution |
| $F_T(t)$ | The Cumulative Distribution Function of variable $t$, where $t$ follows distribution $T$ |

**Algorithm 1 Binary Search**

Input: $SK$, $PS$, $CS$, $x$.

Output: $PS$, $CS$.

1 Let $A=|PS|$; $B=|CS|$;
2 Let $a$ be the minimum value of $PS$ and $b$ is the minimum value of $CS$;
3 Let $s=a+\lceil A/2\rceil-1$;
4 Let $rd=PRndGen(SK,l,(PS,CS,s))$;
5 Let $t=b+NHGDExtSample(A,B,s,rd)$;
6 If $x<s$ then
7 $PS=\{a,...,s-1\}$;
8 $CS=\{b,...,t-1\}$;
9 If $x\geq s$ then
10 $PS=\{s+1,...,a+A\}$;
11 $CS=\{t+1,...,b+B\}$;
12 Endif
13 Return $\{PS, CS\}$.

In Algorithm 1, $PRndGen()$ is a pseudorandom function (PRF) whose input length and output length are both variable. It take as input $SK$, output length $l$ and $\{0,1\}^*$ and then return $\{0,1\}^l$, where the input $\{0,1\}^*$ depends on $PS$, $CS$ and the plaintext.

The $NHGDExtSample()$ is a sampling algorithm for generating negative hypergeometric variables. It takes as input $|PS|, |CS|$ and a plaintext $x$ and return a ciphertext $y$ with probability $NHGD(y-b; |PS|, |CS|, x-a)$, where $a$ is the minimum value of $PS$ and $b$ is the minimum value of $CS$.

**Algorithm 2 Otm OPE**

Input: $SK$, $PS$, $CS$, $x$, $fid$.

Output: $y$.

1 While $PS$ contains more than one plaintext do
2 $\{PS, CS\} = B\text{inary } Search(SK, PS, CS, x)$ ;
3 End While
4 Let $coin = PRndGen(SK, l_1, (PS, CS, x, cc))$ ;
5 Choose a number $y$ in $CS$ with $coin$ ;
6 Return $y$ .

In Algorithm 2, for a given plaintext, it first determines the mapping interval of ciphertexts by calling Binary Search ( ) (Step 2). Once the interval is fixed, a random seed $coin$ is then generated by $PRndGen$ ( ) (Step 4). With the $coin$ , a number in the interval is randomly chosen as the ciphertext (Step 5). We stress that the result of Step 5 is not deterministic. In this way, the goal of one-to-many can be achieved, that is the same plaintext will be mapped to different ciphertexts.

## 3.Sampling Following the Negative Hypergeometric Distribution

To the best our knowledge, the exact sampling algorithm for generating variables following NHGD has not appeared. And it is believed as an open problem. In this section, we give a solution for this problem.

It is impossible to generate NHGD variables by direct sampling method as it has four parameters. We then turn to the acceptance-rejection sample method, requiring a continuous approximation for NHGD. Here a Gamma approximation is given. In this section, we first introduce the acceptance-rejection sample method. Then the upper bound for Gamma approximation is given. Based on above work, an exact sampling method for NHGD is proposed.

### *3.1. Acceptance-Rejection Method*

This method can generate variables following distribution function $f$ by utilizing another distribution $g$ , where $g$ is a continuous and there exists sampling method following distribution $g$ .

The method runs as follows [12]:

(1) Sample $v$ following $g$ ;

(2) Generate $\omega \sim U[0,1]$ ;

(3) If $\omega \le \dfrac{f(v)}{l \cdot g(v)}$ , where $l > 1$ , then output $v$ , else nothing is output and new $v$ be generated.

Note that distribution $g$ and distribution $f$ should be independent of each other. With this method, we know that the probability of outputting $v$ is $\dfrac{1}{l}$ . That means

$P\left(\omega \le \frac{f(v)}{l \cdot g(v)}\right) = \frac{1}{l}$. Obviously, the smaller the value of $l$, the higher the sampling efficiency. In general, let $l$ be maximum of $\frac{f(v)}{g(v)}$.

We wish to select variables according to negative hypergeometric distribution $NHGD_{k,A,B}$, which we bound with $NBD_{k,q}$, where $q = A/B$. The $NBD_{k,q}$ is being bounded by the Gamma distribution $Gamma_{k,\vartheta}$, where $\vartheta = -\ln(1-q)$. Here NBD is discrete, whereas Gamma is continuous.

$$NHGD_{k,A,B}(v) = \frac{\binom{v-1}{k-1}\binom{B-v}{A-k}}{\binom{B}{A}} \tag{1}$$

$$NBD_{k,q}(v) = \binom{v-1}{q-1} q^k (1-q)^{v-k} \tag{2}$$

$$Gamma_{k,\vartheta}(v) = \frac{\vartheta^k v^{k-1} e^{-\vartheta v}}{\Gamma(k)} \tag{3}$$

### *3.2. Upper-Bounding Distribution*

**Theorem 1.** If $A, B, k$ are integers, let $q = A/B$, where $A \le B, k \in \{1, 2, ..., A\}$, then:

$$\frac{NHGD_{k,A,B}(v)}{NBD_{k,q}(v)} \le \left(1 - \frac{k}{B-k}\right)^{B-A} \tag{4}$$

**Theorem 2.** If $k > 0, q > 0$, Let $\vartheta = -\ln(1-q)$, then:

$$\frac{NBD_{k,q}(v)}{Gamma_{k,\vartheta}(v)} \le \left(\frac{q}{\vartheta(1-q)}\right)^k \tag{5}$$

With Theorem 1 and Theorem 2, then:

$$\frac{NHGD_{k,A,B}(v)}{Gamma_{k,\vartheta}(v)} \le \left(1 - \frac{k}{B-k}\right)^{B-A} \cdot \left(\frac{q}{\vartheta(1-q)}\right)^k \tag{6}$$

### *3.3. Gamma Distribution*

If $\omega \sim U[0,1]$, the variable $v$ following the Gamma distribution and $\omega \le NHGD(v)/l \cdot Gamma(v)$, the integer part of $v$ we need, and then we have NHGD.

### 3.4. Description of Exact Sample Method of Negative Hypergeometric Distribution

**Algorithm 3 NHGDExactSample**

Input: $A$, $B$, $k$, $rd$.

Output: $v$.

1 Let $q = A/B$; $\vartheta = -\ln(1-q)$;

2 Let $l = \left(1-\frac{k}{B-k}\right)^{B-A} \cdot \left(\frac{q}{\vartheta(1-q)}\right)^{k}$;

3 Let $x = Gamma(k, \vartheta; rd)$[13];

4 Let $v = \lfloor x + 0.5 \rfloor$;

5 If $v > B+k-A$ or $v < k$, then go to step 3;

6 Let $NHGD(v) = \frac{\binom{v-1}{k-1}\binom{B-v}{A-k}}{\binom{B}{A}}$;

7 Let $Gamma(v) = \frac{\vartheta^k v^{k-1} e^{-\vartheta v}}{\Gamma(k)}$;

8 Generate $\omega \sim U[0,1]$;

9 If $\omega \le NHGD(v)/l \cdot Gamma(v)$ then return $v$;

10 Go to step 3.

According to the description of Algorithm 3, variable $v$ will be returned when $\omega \le NHGD(v)/l \cdot Gamma(v)$ with probability $1/l$. Next we will proof the correctness of Algorithm 3. In another words, we will illustrate $v$ follows exactly NHGD by proving $F(v)$ is equal to $F_{NHGD}$.

$$
\begin{aligned}
P\left(\omega \le \frac{NHGD(v)}{l \cdot Gamma(v)} \mid V \le v\right) &= \frac{P\left(\omega \le \frac{NHGD(v)}{l \cdot Gamma(v)}, V \le v\right)}{P(V \le v)} = \frac{P\left(\omega \le \frac{NHGD(v)}{l \cdot Gamma(v)}, V \le v\right)}{F_{Gamma}(\chi)} \\
&= \int_{-\infty}^{v} \frac{P\left(V \le \frac{NHGD(v)}{l \cdot Gamma(v)} \mid V = \lambda \le v\right)}{F_{Gamma}(v)} \cdot Gamma(\lambda)\, d\lambda \\
&= \frac{1}{l \cdot F_{Gamma}(v)} \int_{-\infty}^{v} \frac{NHGD(\lambda)}{Gamma(\lambda)} \cdot Gamma(\lambda)\, d\lambda \\
&= \frac{1}{l \cdot F_{Gamma}(v)} \int_{-\infty}^{v} NHGD(\lambda)\, d\lambda \\
&= \frac{F_{NGHD}(v)}{l \cdot F_{Gamma}(v)}
\end{aligned}
\tag{7}
$$

Algorithm 3 generates variable $v$ while $\omega \le NHGD(v)/l \cdot Gamma(v)$ is satisfied. Thus we have

$$P\left(V \le v \mid \omega \le \frac{NHGD(v)}{l \cdot Gamma(v)}\right) = \frac{P\left(V \le v, \omega \le \frac{NHGD(v)}{l \cdot Gamma(v)}\right)}{P\left(\omega \le \frac{NHGD(v)}{l \cdot Gamma(v)}\right)} = \frac{F_{Gamma}(v) \cdot \frac{F_{NGHD}(v)}{l \cdot F_{Gamma}(v)}}{1/l} = F_{NGHD}(v) \tag{8}$$

## 4. Security Analysis

We first evaluate security by analyzing the security of *PRndGen* in Algorithm 1.

*RndGen* is a PRF with variable length inputs and outputs. For adversar $A$, its advantage against *PRndGen* is defined as

$$\begin{aligned} &\mathrm{Adv}_{PRndGen}^{PRF}(A) \\ &= \Pr\left(A^{PRndGen(SK,l,\{0,1\}^*)} = 1\right) - \Pr\left(A^{O(l,\{0,1\}^*)} = 1\right) \end{aligned} \tag{9}$$

Where $SK$ is key, $l$ is output length, and $O$ is an oracle which outputs random numbers $\{0,1\}^l$.

For computing $\mathrm{Adv}_{PRndGen}^{PRF}(A)$, we set

$$\begin{aligned} &\mathrm{Adv}_{PRndGen}^{PRF}(A) \\ &= \Pr\left(A^{PRndGen(SK,l,\{0,1\}^*)} = 1\right) - \Pr\left(A^{O_F(SK,l,\{0,1\}^*)} = 1\right) \\ &\quad + \Pr\left(A^{O_F(SK,l,\{0,1\}^*)} = 1\right) - \Pr\left(A^{O(l,\{0,1\}^*)} = 1\right) \end{aligned} \tag{10}$$

where oracle $O_F(SK,l,\{0,1\}^*)$ take as inputs key $SK$, output length $l$ and $\{0,1\}^*$ and then returns $G(1^l,s)$ with random $s \in \{0,1\}^k$.

Construct an adversary $B$ as following, where $B$ is PRF.

Adversary $B^{O(\cdot)}$

(1) Choose $m \in [1,k]$ randomly;

(2) Run $A$ and answer its queries as:

a. Deliver $G(l,F(SK,r))$ to $A$ with first $q-1$ queries, where $r$ is seed;

b. Deliver $G(l,O(r))$ to $A$ with the $q\,th$ queries;

c. Deliver $G(l,s)$ to $A$ with $q+1,\ldots,k$ queries;

(3) Return the outputs of $A$.
Then,

$$\begin{aligned}\Pr\left(A^{PRndGen(SK,l,\{0,1\}^*)}=1\right)-\Pr\left(A^{O_F(SK,l,\{0,1\}^*)}=1\right)&=\sum_{t=1}^{k}\left(\Pr\left(B^{F(SK,\cdot)}=1|\,E\right)-\Pr\left(B^{R(\cdot)}=1|\,E\right)\right)\\&=k\cdot\frac{1}{k}\cdot\sum_{t=1}^{k}\left(\Pr\left(B^{F(SK,\cdot)}=1|\,E\right)-\Pr\left(B^{R(\cdot)}=1|\,E\right)\right)\\&=k\cdot\sum_{t=1}^{k}\left(\Pr\left(B^{F(SK,\cdot)}=1\text{ and }E\right)-\Pr\left(B^{R(\cdot)}=1\text{ and }E\right)\right)\\&\le k\cdot\left(\Pr\left(B^{F(SK,\cdot)}=1\right)-\Pr\left(B^{R(\cdot)}=1\right)\right)\\&=k\cdot\mathrm{Adv}_F^{prf}(B)\end{aligned}\tag{11}$$

where we denote by $E:\ m=t$.
Construct an adversary $C$ against blockcipher $O_E$, where $C$ is PRG. We then have

$$\begin{aligned}\Pr\left(A^{O_F(SK,l,\{0,1\}^*)}=1\right)-\Pr\left(A^{O(l,\{0,1\}^*)}=1\right)&=\sum_{t=1}^{k}\left(\Pr\left(C^{O_E(SK,l,\cdot)}=1|\,E\right)-\Pr\left(C^{R(\cdot)}=1|E\right)\right)\\&=k\cdot\frac{1}{k}\cdot\sum_{t=1}^{k}\left(\Pr\left(C^{O_E(SK,l,\cdot)}=1|\,E\right)-\Pr\left(C^{R(\cdot)}=1|E\right)\right)\\&=k\cdot\sum_{t=1}^{k}\left(\Pr\left(C^{O_E(SK,l,\cdot)}=1\text{ and }E\right)-\Pr\left(C^{R(\cdot)}=1\text{ and }E\right)\right)\\&\le k\cdot\left(\Pr\left(C^{O_E(SK,l,\cdot)}=1\right)-\Pr\left(C^{R(\cdot,\cdot)}=1\right)\right)\\&=k\cdot\mathrm{Adv}_{O_E}^{prg}(C)\end{aligned}\tag{12}$$

With Equation (1) and (2), then
$\mathrm{Adv}_{PRndGen}(A)\le k\cdot(\mathrm{Adv}_F^{prf}(B)+\mathrm{Adv}_{O_E}^{prg}(C))+\lambda$, where the value of $\lambda$ depends on the sampling accuracy of NHGD.

As analysis in Section 2, random seed *coin* is used to choose the ciphertext in final selection. Therefore different ciphertexts will be chosen to be the different mapping of the same plaintext. Our Otm OPE is not deterministic and has higher security than BCLO scheme.

## 5. Efficiency Analysis

From Algorithm 1 and 2, it is easy to observe the computation cost of the scheme depends on the running time of Binary Search.

There is $P$ elements in the plaintext space. To reach one element, on average, the expected recursions is $M=\frac{1}{P}\cdot\left[\sum_{k=1}^{log\,P}2^{k-1}k+(log\,P+1)\right]$. Next we estimate the value of $M$. Let $\mathrm{A}=log\,P$, $\mathrm{A}\ge 1$, then

$$M = M_{\mathrm{A}} = \frac{1}{2^{\mathrm{A}}} \cdot \left[ \sum_{k=1}^{\mathrm{A}} 2^{k-1} k + (\mathrm{A}+1) \right] \tag{13}$$

Mathematical induction is employed on $\mathrm{A}$ to prove $\mathrm{A}-1 \le M_{\mathrm{A}} \le \mathrm{A}$.

(1) When $\mathrm{A}=1$, then $M_1=1$ and $0 \le M_1 \le 1$;

$$M_w = \frac{1}{2^w} \cdot \left[ \sum_{k=1}^{w} 2^{k-1} k + (w+1) \right] \text{ and } w-1 \le M_w \le w \tag{14}$$

(2) Suppose $\mathrm{A} = w$ and assume

(3) when $\mathrm{A} = w+1$, it gets

$$M_{w+1} = \frac{1}{2^{w+1}} \cdot \left[ \sum_{k=1}^{w+1} 2^{k-1} k + (w+2) \right] = \frac{1}{2} M_w + \frac{1}{2} w + \frac{1}{2} + \frac{1}{2^{w+1}} \tag{15}$$

Obviously, $w \le M_{w+1} \le w+1$.

As analysis above, the procedure Binary Search ( ) would be called about $\log P$ rounds to map a plaintext to a randomized interval. While in BCLO scheme, $\log C$ recursive calls are required to determine the mapping interval. In real applications, the value of $P$ is much smaller than that of $C$, so our Otm OPE has higher efficiency.

## 6. Conclusions

In this paper, we focus on how to enable OPE to support effectively searching with higher security. To this end, we derive a practical sampling algorithm following NHGD distribution. Based on this work, an efficient order-preserving and one-to-many searchable encryption is proposed. In our scheme, a value in the ciphertext bucket is chosen as the mapping of a given plaintext. Comparing with BCLO scheme, we prove that our Otm scheme is more secure. We also show that our work is more efficient by performance analysis.

## Acknowledgement

This work was supported by the National Natural Science Foundations of China (No. 61702238, No.61862027), Jiangxi Provincial Natural Science Foundation (20202BABL202041, 20192BAB207008), The Technology Project of Jiangxi Education Department (No. GJJ170316, GJJ180264) and The 15th Student Research Project of JXUFE (20200611170950596).

## References


[1] C. Gentry. Fully Homomorphic Encryption Using Ideal Lattices[C]. Proceedings of the 41st Annual ACM Symposium on Theory of Computing, ACM Press, 2009, 169-178.

[2] Agrawal R, Kiernan J, Srikant R, et al. Order preserving encryption for numeric data. Proceedings of the 2004 ACM SIGMOD international conference on Management of data. 2004:563-574.

[3] Boldyreva A, Chenette N, Lee Y, et al. Order-preserving symmetric encryption. Advances in Cryptology-EUROCRYPT 2009. Springer Berlin Heidelberg, 2009: 224-241.

[4] Popa R A, Li F H, Zeldovich N. An ideal-security protocol for order-preserving encoding. Security and Privacy (SP), 2013 IEEE Symposium on. IEEE, 2013: 463-477.

[5] Xiao L, Yen I L. Security analysis for order preserving encryption schemes[C]. Information Sciences and Systems (CISS), 2012 46th Annual Conference on. IEEE, 2012: 1-6.

[6] Z. Liu et al. New Order Preserving Encryption Model for Outsourced Databases in Cloud Environments. Journal of Network and Computer Applications. 59 (2016) :198-207.

[7] Lewi K, Wu D J. Order-revealing encryption: New constructions, applications, and lower bounds[C]. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016: 1167-1178.

[8] Boldyreva A, Chenette N, O'Neill A. Order-preserving encryption revisited: Improved security analysis and alternative solutions[C]. Annual Cryptology Conference. Springer, Berlin, Heidelberg, 2011: 578-595.

[9] Popa R A, Redfield C, Zeldovich N, et al. CryptDB: protecting confidentiality with encrypted query processing[C]. Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles. ACM, 2011: 85-100.

[10] Wang C, Cao N, Li J, et al. Secure ranked keyword search over encrypted cloud data[C]. 2010 IEEE 30th international conference on distributed computing systems. IEEE, 2010: 253-262.

[11] Liu D, Wang S. Nonlinear order preserving index for encrypted database query in service cloud environments[J]. Concurrency and Computation: Practice and Experience, 2013, 25(13): 1967-1984.

[12] Schmeiser B W, Shalaby M A. Acceptance/rejection methods for beta variate generation[J]. Journal of the American Statistical Association, 1980, 75(371): 673-678.