Pseudo-random Exponentiation Using the Lim-Lee Method

C.P. Schnorr

Fachbereich Mathematik/Informatik
Universität Frankfurt, Germany
schnorr@cs.uni-frankfurt.de

Abstract for rump and poster session

Suppose we want to compute $g^R$ for a pseudo-random $n$ bit exponent $R$. We first divide $R$ into $h$ blocks $R_i$, for $0 \leq i \leq h - 1$, of size $a = \lceil \frac{n}{h} \rceil$ and then subdivide each $R_i$ into $v$ smaller blocks $R_{i,j}$, for $0 \leq j \leq v - 1$ of size $b = \lceil \frac{n}{hv} \rceil$ with $R_{i,j}$ having bits $e_{i,jb+k}$ for $k = 0, ..., b - 1$. We have for $vh \mid n$:

$$R = R_{h-1} \ldots R_1 R_0 = \sum_{i=0}^{h-1} R_i 2^ia_i, \quad R_i = R_{i,v-1} \ldots R_{i,1} R_{i,0} = \sum_{j=0}^{v-1} R_{i,j} 2^jb_i,$$

$$R_{i,j} = e_{i,jb+b-1} \ldots e_{i,jb+1} e_{i,jb} = \sum_{k=0}^{b-1} e_{i,jb+k} 2^k,$$

$$R = \sum_{k=0}^{b-1} \sum_{j=0}^{v-1} L_{j,k} 2^k, \text{ where } L_{j,k} := \sum_{i=0}^{h-1} e_{i,jb+k} 2^ia_i + jb_i.$$

For each $j$ and $k$ there are $2^h$ combinations for the $h$ bits $e_{i,jb+k}$ for $i = 0, ..., h - 1$. For each $j$ there are $2^h - 1$ non-zero integers $\sum_{i=0}^{h-1} e_{i,jb+k} 2^ia_i + jb_i$. We select for each $j$ a subset $\mathcal{L}_j$ of $s \approx 2^{h/2} - 1$ of these integers. We precompute and store $g^L$ for $L \in \mathcal{L}_j$ for $j = 0, ..., v - 1$. Let $\mathcal{L} := \bigcup_{j=0}^{v-1} \mathcal{L}_j$. We generate random pairs in $\mathcal{L} \times \mathcal{L}$:

**Lim-Lee-pseudo-random exponentiation.**

$Z := 1, \quad L := 0$

for $k = b - 1$ to 0 step -1

$Z := Z \ast Z, \quad L := L + L$

for $j = v - 1$ to 0 step -1

pick $L_j \in_R \mathcal{L}_j$ at random

$Z := Z \ast g^{L_j}, \quad L := L + L_j$

return $(L, Z)$.

*Performance* for exponents $R$ of bit length $n = 160 / 1024$ at DL-complexity $2^{n/2}$. The number of multiplications is $a + b - 2$, where $a = n/h, b = n/(hv)$, we have $\#\mathcal{L} = \int_{L}^{L} f \int_{L}^{L}$. 

1
configuration | storage | \# multiplications | \# L
---|---|---|---
$h \times v$ | $s \times v$ | $n = 160$ | $n = 1024$ | 160 | 1024
$4 \times 1$ | $4 \times 1$ | 78 | 510 | $2^{80}$ | $2^{512}$
$4 \times 2$ | $4 \times 2$ | 58 | 372 | $2^{80}$ | $2^{512}$
$6 \times 3$ | $8 \times 3$ | 34 | 226 | $2^{81}$ | $2^{512}$

*Good choices for $|\mathcal{L}|$. Let $\mathcal{L}_j$ for $j = 0, \ldots, v - 1$ consist of the $s$ non-zero integers $L_j = \sum_{i=0}^{h-1} e_i 2^{ia+jb}$ of smallest (resp., highest) Hamming-weight $\sum_{i=0}^{h-1} e_i$. Then additive relations $u + v = w$ with $u, v, w \in \mathcal{L}$ are nearly excluded. However, fast generic DL-algorithms for $g^L$ require many additive relations in $\mathcal{L}$.  