The Mist Exp^n Algorithm

\[\text{To compute: } ResultM \times M^n\]
StartM \leftarrow M;
ResultM \leftarrow 1;
While \ E > 0 \ do
Begin
  Choose a random "divisor" D;
  R \leftarrow E \mod D;
  If \ R = 0 \ then
    ResultM \leftarrow \text{StartM} \times \text{ResultM} ;
    \text{StartM} \leftarrow \text{StartM}^D ;
    E \leftarrow E \div D ;
  \end{If}
\{Invariant: } \text{StartM}^{\text{ResultM}} = \text{StartM} \times \text{ResultM} \}
End

Addition Sub-Chains

\text{We need instructions which include the update of ResultM:}
\text{If } i \text{ means multiply contents at addresses } i \text{ and } j \text{ and write result to address } k.
\text{Use 1 for location of StartM, 2 for TempM, 3 for ResultM:}
- (111) for \text{StartM} = (2,0)
- (112, 123) for \text{StartM} = (2,1)
- (112, 121) for \text{StartM} = (3,0)
- (112, 131, 121) for \text{StartM} = (3,1)
- (112, 231, 121) for \text{StartM} = (3,2)
- (121, 121) for \text{StartM} = (5,0)
- (121, 131, 121) for \text{StartM} = (5,1)
- (121, 231, 121) for \text{StartM} = (5,2)
- (121, 131, 131, 121) for \text{StartM} = (5,3)
- (121, 223, 121) for \text{StartM} = (5,4)

Average Add^n Chain Properties

- The probabilities of addition sub-chain lengths are:
  - length 1 is \( p_{1,1} = 0.354 \)
  - length 2 is \( p_{1,2} + p_{2,1} = 0.458 \)
  - length 3 is \( p_{2,2} + p_{2,3} + p_{3,2} = 0.139 \)
  - length 4 is \( p_{3,3} + p_{3,4} + p_{4,3} + p_{4,4} = 0.049 \)
- So average divisor sub-chain has length 1.883 mult
- Av decrease in \( E \) is 2038 per subchain
- So 0.757 log\_26 subchains \& 1.425 log\_26 mult
- This is faster than the binary exp^n algorithm and marginally slower than 4-ary exp^n

Choice of Divisor

\text{Initial choice:}
D \leftarrow 0 ;
If Random(8) < 7 then
If (E mod 2) = 0 then D \leftarrow 2 else
If (E mod 3) = 0 then D \leftarrow 5 else
If (E mod 3) = 0 then D \leftarrow 3 ;
If D = 0 then
Begin
p \leftarrow Random(8) ;
If p < 6 then D \leftarrow 2 else
If p < 7 then D \leftarrow 3 else
D \leftarrow 5
End
A^{\phi(n)}: 14247 \times log\_26 mult

Probability of each \text{pair } (D,R)

\text{This gives the probabilities:}
\text{If divisor } D
\text{If pair } (D,R)
- \( p_2 = 0.629 \)
- \( p_3 = 0.228 \)
- \( p_4 = 0.142 \)
Choice of Divisor

A semi-deterministic choice:

\[ D \leftarrow 0 \; ; \]
\[ \text{(Delete this line: if Random}(8)<7 \text{ then)} \]
\[ \text{if (mod } E 2) = 0 \text{ then } D \leftarrow 2 \text{ else } \]
\[ \text{if (mod } E 5) = 0 \text{ then } D \leftarrow 5 \text{ else } \]
\[ \text{if (mod } E 3) = 0 \text{ then } D \leftarrow 3 \; ; \]
\[ \text{if } D = 0 \text{ then End} \]

S&M Chains

- Assume an attacker can distinguish squares and multiplies from a single exponentiation (e.g. from Hamming weights of arguments deduced from power variation on bus).
- A division chain is the list of pairs \((D, R)\) used in an \(exp^a\) scheme. It determines the addition chain to be used, and hence the sequence of squares and multiplies which occur:
  \[
  (2, 0), (2, 1), (3, 0), (3, 1), (3, 2), (5, 0), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (5, 7), (5, 8), (5, 9), (5, 10)
  \]
- Divisor sub-chain boundaries are deduced from occurrences of \(S\) except for ambiguity between \((5, 4)\) and \((2, 0)(3, 0)\) or \((2, 0)(5, 0)\).

S&M Chains

- There is are:
  
  1 way to interpret \(S\).
  2 ways to interpret \(SS\),
  3 ways to interpret \(SSM\) with no preceding \(S\),
  4 ways to interpret \(SSMM\) with preceding \(S\),
  4 ways to interpret \(SSMMM\).
- Using known probabilities for each occurring:
  \[ \text{THEOREM: The search space for exponents with the same S&M sequence as } E \text{ has size approx } E^{0.5} \]
- For 4-ary \(exp^3\) it is much easier to average traces, easier to be certain of the S&M sequence, and the search space is only \(E^{0.125}\) which is smaller.

Operand Re-Use

- \[ \text{THEOREM: With MIST, the search space for exponents with the same operand sharing sequence as } E \text{ has size approx } E^{0.5} \]
  - this assumes \(opt^a\) sharing is determined with total accuracy from one exponentiation;
  - it also assumes unconstrained choice of divisors at each step;
  - in comparison, the search space for \(m\)-ary \(exp^a\) has size \(E^n\).
- It isn't clear if recovery from errors is possible.
- Selecting exact divisors will vastly decrease the search.

Deterministic Choices

- The deterministic constraints cut the search space for \(E\).
- By how much? Consecutive divisor choices are not independent, so theory simplified this way is inadequate.
- When the divisor is chosen semi-deterministically (as above) and these constraints are taken into account:
  \[ \text{THEOREM: The search space for exponents with the same S&M sequence as } E \text{ has size approx } E^{0.14} \]
- It is still computationally infeasible to recover \(E\).

Deterministic Choices

- Knowledge of \(opt^a\) sharing cuts the search space further.
- By how much? Simulations were used to find out.
- When the divisor is chosen deterministically and these constraints are taken into account:
  \[ \text{THEOREM: The search space for exponents with the same } opt^a \text{ sharing pattern as } E \text{ has size approx } E^{0.135} \]
- It may now be computationally feasible to recover \(E\).

768-bit exponents give search space of size \(2^{30}\).
1024-bit known RSA modulus with CRT has size only \(2^{20}\).