Fixed-exponent Exponentiation

André Weimerskirch
escrypt Inc.

Related concepts and keywords

binary exponentiation, RSA, ElGamal decryption

Definition

The exponentiation of a random element $g \in G$, with G some group, by a fixed integer exponent e.

Theory There are several situations where an exponentiation g^e of an arbitrary element $g \in G$, with G some group, by a fixed exponent e needs to be performed. Fixed-exponent exponentiation algorithms aim to decrease the number of multiplications compared to general exponentiation algorithms such as the binary exponentiation algorithm. They are based on the fact that certain precomputations can be performed once for a fixed exponent. The problem of finding the smallest number of multiplications to compute g^e is equivalent to finding the shortest *addition chain* of e. An addition chain is a sequence of numbers such that each number is the sum of two previous ones and such that the sequence starts with 1. For instance, an addition chain of 19 is $V = (1, 2, 4, 5, 9, 10, 19)$.

Definition 1 *An* addition chain V *of length* s *for a positive integer* e *is a sequence* $u_0, u_1, \ldots, u_s$ *of positive integers, and an associated sequence* $w_1, \ldots, w_s$ *of pairs* $w_i = (i_1, i_2)$, $0 \leq i_1, i_2 < i$, *having the following properties: (i)* $u_0 = 1$ *and* $u_l = e$*; and (ii) for each* $u_i, 1 \leq i \leq s, u_i = u_{i_1} + u_{i_2}$.

Fixed exponents are considered here, and the time for precomputations is not taken into account here. However, determining the shortest addition chain for e is believed to be computationally infeasible in most cases for chains of relevant length. Thus in practice there are heuristics used to obtain nearly-optimal addition chains. Knuth [4] describes several such heuristics. It is wise to implement multiple heuristics in order to compare the results, and finally to choose the shortest addition chain. Such precomputation can be computationally very demanding, though. After an addition chain for an exponent e had been obtained, exponentiation can be performed as described in Algorithm 1.

Algorithm 1 Addition Chain Exponentiation
`INPUT`*: a group element g, an addition chain $V = (u_0, u_1, \ldots, u_s)$ of length s for a positive integer e, and the associated sequence $(w_1, \ldots, w_s)$, where $w_i = (i_1, i_2)$*
`OUTPUT`*: g^e*
1. $g_0 \leftarrow g$
2. `For` i `from` 1 `to` s `do`
   2.1 $g_i \leftarrow g_{i_1} \cdot g_{i_2}$
3. `Return` (g_s)

Algorithm 1 computes g^e for a precomputed addition chain for e of length s with s multiplications. For instance, for above example of $e = 19$ it is $V = (1, 2, 4, 5, 9, 10, 19)$. Algorithm 1 then works as follows:

i	0	1	2	3	4	5	6
w_i	-	$(0,0)$	$(1,1)$	$(0,2)$	$(2,3)$	$(3,3)$	$(4,5)$
g_i	g	g^2	g^4	g^5	g^9	g^{10}	g^{19}

In some cases *addition-subtraction chains* might be used to shorten the length of a chain. In such cases a number of the chain is the sum or subtraction of two previous elements in the chain. For instance, the shortest addition chain for 31 is $V = (1, 2, 3, 5, 10, 11, 21, 31)$ whereas there exists a shorter addition-subtraction chain $C' = (1, 2, 4, 8, 16, 32, 31)$ [3]. However, addition-subtraction chains only make sense in cases where an inversion in the underlying group is computationally cheap. Thus, addition-subtraction chains are not used for exponentiation in an RSA modulus but might be applied to elliptic curve operations.

There are two generalized versions of addition chains. An *addition sequence* is an addition chain $V = (u_0, \ldots, u_s)$ such that it contains a specified set of values $r_1, \ldots, r_t$. They are used when an element g needs to be raised to multiple powers $r_i, 1 \leq i \leq t$. Especially when the exponents $r_1, r_2, \ldots, r_t$ are far apart, an addition sequence might be faster in such a case. Finding the shortest addition sequence is known to be NP-complete [2] and thus heuristics are used to find short sequences.

In cases of simultaneous exponentiations such as in the digital signature standard (DSS), a generalized addition chain called *vector-addition chain* can be applied. These are used to compute $g_0^{e_0} g_1^{e_1} \ldots g_{k-1}^{e_{k-1}}$ where the g_i's are

arbitrary elements in G and the e_i's are fixed positive integers. In a vector addition chain, each vector is the sum of two previous ones. For instance, a vector-addition chain C of the vector $[15, 5, 12]$ is
$([1, 0, 0], [0, 1, 0], [0, 0, 1], [1, 0, 1], [2, 0, 2], [2, 1, 2],$
$[3, 1, 2], [5, 2, 4], [6, 2, 5], [12, 4, 10], [15, 5, 12])$.

Definition 2 *Let s and k be positive integers and let v_i denote a k-dimensional vector of non-negative integers. An ordered set $V = \{v_i : -k+1 \leq i \leq s\}$ is called a* vector-addition chain *of length s and dimension k if V satisfies the following: (i) Each $v_i, -k+1 \leq i \leq 0$, has a 0 in each coordinate position, except for coordinate position $i+k-1$, which is a 1. (Coordinate positions are labelled 0 through $k-1$.); and (ii) For each $v_i, 1 \leq i \leq s$, there exists an associated pair of integers $w_i = (i_1, i_2)$ such that $-k+1 \leq i_1, i_2 < i$ and $v_i = v_{i_1} + v_{i_2} (i_1 = i_2$ is allowed).*

Again, the time for precomputations is irrelevant. There is a 1-1 correspondence between vector-addition chains and addition sequences [6]. Hence determining the shortest vector-addition chain is NP-complete and thus heuristics are used to obtain short vector-addition chains [1]. Having a vector-addition chain, exponentiation can be performed as shown in Algorithm 2. The algorithm needs s multiplications for a vector-addition chain of length s to compute $g_0^{e_0} g_1^{e_1} \ldots g_{k-1}^{e_{k-1}}$ for arbitrary base and fixed exponents.

Algorithm 2 Vector-Addition Chain Exponentiation
`INPUT`*: group elements $g_0, g_1, \ldots, g_{k-1}$ and a vector-addition chain V of length s and dimension k with associated sequence $w_1, \ldots, w_s$, where $w_i = (i_1, i_2)$.*
`OUTPUT`*: $g_0^{e_0} g_1^{e_1} \ldots g_{k-1}^{e_{k-1}}$ where $v_s = (e_0, \ldots, e_{k-1})$.*
1. `For` i `from` $(-k+1)$ `to` 0 `do`
  1.1 $a_i \leftarrow g_{i+k-1}$
2.`For` i `from` 1 `to` s `do`
  2.1 $a_i \leftarrow a_{i_1} \cdot a_{i_2}$
3. `Return` (a_s)

An overview of addition chains was given by Knuth [4]. Further examples of fixed-exponent exponentiation can be found in [5] and [3]. A lower bound for the shortest length of addition chains was proven by Schönhage [7], an upper bound is obtained by constructing an addition chain of e from its binary representation, i.e., by the binary exponentiation algorithm. Yao proved

bounds for addition sequences [8].

Applications Fixed-exponent exponentiation can be used in RSA encryption using the fixed public exponent, and in RSA decryption using the fixed secret exponent. The mechanism can also be used in ElGamal decryption and elliptic curve operations. Simultaneous exponentiation with more than one fixed exponent can be applied to the digital signature standard (DSS).

References

[1] J. Bos and M. Coster. Addition Chain Heuristic. In *Proceedings of Crypto '89*, LNCS 435, Springer-Verlag, 1990.

[2] P. Downey, B. Leong, and R. Sethi. Computing sequences with addition chains. In *SIAM Journal on Computing*, vol. 10, 1981.

[3] D.M. Gordon. A Survey of Fast Exponentiation Methods. In *Journal of Algorithms*, vol. 27, pp. 129–146, 1998.

[4] D.E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, 3rd Edition, Addison-Wesley, 1997.

[5] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[6] J. Olivos. On vectorial addition chains. In *Journal of Algorithms*, vol. 2, pp. 13–21, 1981.

[7] A. Schönhage. A lower bound for the length of addition chains. In *Theoretical Computer Science*, vol. 1, pp. 1–12, 1975.

[8] A.C. Yao. On the evaluation of powers. In *SIAM Journal of Computing*, vol. 5, pp. 100–103, 1976.