Published in D. Naccache, Ed., *Cryptography and Security: From Theory to Applications (Quisquater Festschrift)*, vol. 6805 of *Lecture Notes in Computer Science*, pp. 3-7, Springer, 2012.

On Quisquater's Multiplication Algorithm

Marc Joye

Technicolor, Security & Content Protection Labs
1 avenue de Belle Fontaine, 35576 Cesson-Sévigné Cedex, France
marc.joye@technicolor.com

Smart card technologies have had a huge impact on the development of cryptographic techniques for commercial applications. The first cryptographic smart card was introduced in 1979. It implemented the *Telepass 1* one-way function using 200 bytes! Next came smart cards with secret-key and public-key capabilities, respectively in 1985 and 1988. Implementing an RSA computation on a smart card was (and still is) a very challenging task. Numerous tips and tricks were used in the design of the resulting smart-card chip P83C852 from Philips using the *CORSAIR* crypto-coprocessor [1, 12]. Among them was a new algorithm for the modular multiplication of two integers, the *Quisquater's multiplication algorithm* [10, 11]. This algorithm is also present in the subsequent crypto-coprocessors, namely the *FAME* crypto-coprocessor [4] and its various extensions.

1 Quisquater's Algorithm

The classical schoolboy method evaluates the quotient $q = \lfloor U/N \rfloor$ of the integer division of U by N in a digit-by-digit fashion; the remainder of the division is $r = U - qN$. Let $\beta = 2^k$ for some integer $k \geq 1$. If $N = \sum_{i=0}^{n-1} N_i \beta^i$ and $U = \sum_{i=0}^{n} U_i \beta^i$ (with $0 \leq N_i, U_i \leq \beta - 1$ and $N_{n-1}, U_n \neq 0$) denote the respective k-ary expansion of N and U then a good estimate for the quotient $q \in [0, \beta)$ is given by $\hat{q} = \min(\lfloor (U_n \beta + U_{n-1})/N_{n-1} \rfloor, \beta - 1)$; see e.g. [6, p. 271]. In particular, when $N_{n-1} \geq \beta/2$, it is easily verified that $\hat{q} - 2 \leq q \leq \hat{q}$. This means that the exact value for quotient q can then be obtained from $\hat{q}$ with at most two corrections.

In order to simplify the presentation, we further assume that N *is not a power of* 2 — remark that evaluating a reduction modulo a power of 2 is trivial. Quisquater's algorithm [9] relies on the observation that quotient $q = \lfloor U/N \rfloor$ is lower bounded by the approximated quotient

$$\hat{q} = \left\lfloor \frac{U}{2^c \beta^n} \right\rfloor \cdot \left\lfloor \frac{2^c \beta^n}{N} \right\rfloor$$

for some integer $c > 0$, which defines a remainder; namely,

$$\hat{r} := U - \hat{q}N = U - \left\lfloor \frac{U}{2^c \beta^n} \right\rfloor \cdot \left\lfloor \frac{2^c \beta^n}{N} \right\rfloor N \ .$$

Hence, letting $N' = \delta N$ where $\delta = \lfloor (2^c \beta^n)/N \rfloor$, we see that obtaining $\hat{r}$ merely requires a binary shift operation —i.e., a division by a power of 2, by evaluating $\hat{r}$ as $\hat{r} = U - \lfloor U/2^{kn+c} \rfloor N'$ (remember that $\beta = 2^k$). This of course supposes the precomputation of N'.

By construction, the c most significant bits of modulus N' are equal to 1. Indeed, from $N' = \delta N = \lfloor (2^c \beta^n)/N \rfloor N = 2^c \beta^n - (2^c \beta^n \bmod N)$ and since

1. $(2^c \beta^n \bmod N) \geq 1$ because N is assumed not to be a power of 2,
2. $(2^c \beta^n \bmod N) \leq N - 1 \leq \beta^n - 2$,

we get $2^c \beta^n - 1 \geq N' \geq 2^c \beta^n - (\beta^n - 2) > (2^c - 1)\beta^n$. This also shows that $|N'|_2 = kn + c$; i.e., that the bit-length of N' is $kn + c$. Such a modulus is called a *diminished-radix modulus* [8].

It is worth noting that the two divisions in the expression of $\hat{q}$ are rounded by default so that the value of $\hat{q}$ will never exceed that of q and thus that $\hat{r}$ will never be negative. Further, the subtraction in the expression of $\hat{r}$ can advantageously be replaced with an addition using the 2-complemented value of N', $\overline{N'} = 2^{|N'|_2} - N'$, as

$$\hat{r} = U \bmod 2^{kn+c} + \left\lfloor \frac{U}{2^{kn+c}} \right\rfloor \cdot \overline{N'} \ .$$

It is also worth noting that $\hat{r} \equiv U \pmod{N'}$. Moreover, from the schoolboy method, it is very likely a correct estimate for $(U \bmod N')$ for a sufficiently large value for c. This is easy to check. Define $r' = U \bmod N'$. We have:

$$\hat{r} - r' = \left(\left\lfloor \frac{U}{N'} \right\rfloor - \left\lfloor \frac{U}{2^{kn+c}} \right\rfloor \right) N'$$

and

$$\left\lfloor \frac{U}{2^{kn+c}} \right\rfloor \leq \left\lfloor \frac{U}{N'} \right\rfloor \leq \begin{cases} \left\lfloor \dfrac{U}{2^{kn+c}} + \dfrac{1}{2^{c-(k+1)}} \right\rfloor & \text{if } c \propto k \,, \\ \left\lfloor \dfrac{U}{2^{kn+c}} + \dfrac{1}{2^{c+(c \bmod k)-(2k+1)}} \right\rfloor & \text{otherwise} \ . \end{cases}$$

For example, if $c = 2k$, $\hat{r}$ is expected to be equal to $(U \bmod N')$ with at least a probability of $1 - 2^{k-1}$; if not, then $\hat{r} - N'$ yields the value of $(U \bmod N')$.

Proof. The schoolboy method computes digit-by-digit the quotient (and corresponding remainder) of an $(\ell + 1)$-digit number by an ℓ-digit number. As Quisquater's algorithm replaces modulus N by modulus $N' = \delta N$, which is a $(n + \lceil c/k \rceil)$-digit, we assume that

$$U = \sum_{i=0}^{n+\lceil \frac{c}{k} \rceil} U_i \, \beta^i \quad \text{with } 0 \leq U_i < \beta \text{ and } U_{n+\lceil \frac{c}{k} \rceil} \neq 0 \ .$$

The relation on $\hat{r} - r'$ is immediate: $\hat{r} - r' = U - \lfloor U/(2^c\beta^n)\rfloor N' - (U \bmod N') = \lfloor U/N'\rfloor N' - \lfloor U/(2^c\beta^n)\rfloor N'$. For the second relation, $\lfloor U/N'\rfloor \geq \lfloor U/2^{kn+c}\rfloor$ since $N' < 2^{kn+c}$. Furthermore, since $N' > (2^c-1)\beta^n$ and $U < \beta^{n+\lceil \frac{c}{k}\rceil+1}$, we get

$$\begin{aligned}\left\lfloor \tfrac{U}{N'}\right\rfloor &\leq \left\lfloor \tfrac{U}{(2^c-1)\beta^n}\right\rfloor = \left\lfloor \tfrac{U}{2^c\beta^n} + \tfrac{U}{2^c(2^c-1)\beta^n}\right\rfloor \leq \left\lfloor \tfrac{U}{2^c\beta^n} + \tfrac{\beta^{\lceil \frac{c}{k}\rceil+1}}{2^c(2^c-1)}\right\rfloor \\ &\leq \left\lfloor \tfrac{U}{2^c\beta^n} + \tfrac{2^{k\lceil \frac{c}{k}\rceil+k}}{2^{2c-1}}\right\rfloor = \left\lfloor \tfrac{U}{2^c\beta^n} + \tfrac{1}{2^{2c-1-k\lceil \frac{c}{k}\rceil-k}}\right\rfloor \quad .\end{aligned}$$

Suppose first that $c \propto k$ (i.e., that $c \bmod k = 0$). Then we have $2c-1-k\lceil \frac{c}{k}\rceil - k = c-k-1$. Suppose now that $c \not\propto k$. Then $k\lceil c/k\rceil = k\lfloor c/k\rfloor + k = c + k - (c \bmod k)$ and therefore $2c - 1 - k\lceil \frac{c}{k}\rceil - k = c + (c \bmod k) - 2k - 1$. □

The description we gave is a high-level presentation of the algorithm. There is more in Quisquater's algorithm. We refer the reader to [10, 11] for low-level implementation details. See also [1, 4, 2]. In the next sections, we will discuss the normalization process (i.e., the way to get N') and some useful features satisfied by the algorithm.

2 Normalization and Denormalization

Quisquater's algorithm requires that the c most significant of the modulus are equal to 1. For that purpose, an input modulus N is transformed into a *normalized modulus* $N' = \delta N$. As shown before, a valid choice for δ is $\delta = \lfloor 2^{|N|_2+c}/N\rfloor$.

We note that a full division by N is not necessary to obtain the value of normalization factor δ. If we let

$$\hat{\delta} = \left\lfloor \frac{2^{2c+2}}{\hat{N}} \right\rfloor$$

where $\hat{N}$ denotes the $(c+2)$ most significant bits of N, then $\delta \leq \hat{\delta} \leq \delta + 1$ [5, 3]. Hence, if we take $\hat{\delta}$ as an approximation for δ, the error is at most one. As a result, with only one test, we obtain the exact value of δ from the $(c+2)$ most significant bits of N.

The bit-length of the normalized modulus, $N' = \delta N$, is of $(kn + c)$ bits. If the word-size of the device implementing the algorithm is of k bits, it may be possible to increase the bit-length of N' without degrading the performance, provided that the word-length of the resulting modulus remains the same. As a consequence, it is smart to select c as a multiple of k. Doing so, the probability that $\hat{r}$ is the exact value for $(U \bmod N')$ will be maximal for a given word-length for N'.

If that probability is already high, another option would be to exploit the possible additional bits to diversify the normalized moduli. Application will be presented in the next section. The number of additional bits is given by $B := -c \bmod k$. The problem now consists in constructing normalization factors δ so that $N' = \delta N$ has at most $kn + c + B = k(n + \lceil c/k\rceil)$ bits and whose c most

significant bits are 1's. Letting as before $N = \sum_{i=0}^{n-1} N_i\,\beta^i$ the k-ary expansion of modulus N, we may define

$$\delta_{b,t} = \left\lfloor \frac{2^{c+b}\beta^n - t}{N} \right\rfloor \quad \text{for any } b \in \{0,\ldots,B\} \text{ and } t \in \{1,\ldots,(2^b-1)\beta^n + 2\} \ .$$

They are all valid normalization factors. Note that for such $\delta_{b,t}$, the expression for $\hat{r} = \hat{r}_{b,t}$ becomes $\hat{r} = U - \left\lfloor \frac{U}{2^{c+b}\beta^n} \right\rfloor \cdot (\delta_{b,t} N)$.

Proof. Define $N'_{b,t} = \delta_{b,t}\,N$ and $R_{b,t} = (2^{c+b}\beta^n - t) \bmod N$. Fix $b \in \{0,\ldots,B\}$. From the definition of $\delta_{b,t}$, we get $N'_{b,t} = 2^{c+b}\beta^n - t - R_{b,t}$. Hence, we have $N'_{b,t} \leq 2^{c+b}\beta^n - 1$ since $t \geq 1$ and $R_{b,t} \geq 0$. We also have $N'_{b,t} \geq 2^{c+b}\beta^n - (2^b - 1)\beta^n - 2 - (\beta^n - 2) = (2^c - 1)2^b\beta^n$ since $t \leq (2^b - 1)\beta^n + 2$ and $R_{b,t} \leq N - 1 \leq \beta^n - 2$. This shows that $N'_{b,t}$ has always its c most significant bits equal to 1. Moreover, $N'_{b,t} \leq 2^{c+b}\beta^n - 1 \leq 2^{c+B}\beta^n - 1$ implies that $N'_{b,t}$ has a length of at most $(kn + c + B)$ bits. □

Again the computation of the normalization factors can be sped up by considering only some highest part of N.

3 Application

The setting of Quisquater's multiplication suits particularly well an RSA computation [13]. Suppose for example that one has to compute the RSA signature $S = \mu(m)^d \bmod N$ on some message m, where d denotes the private signing key and μ represents some padding function. Signature S can be equivalently obtained using only modulo N' arithmetic as

$$S = \frac{\delta \cdot \left(\mu(m)^d \bmod N'\right) \bmod N'}{\delta} \ .$$

The correctness follows by noting that $\delta A \bmod \delta N = \delta(A \bmod N)$ for any integer A.

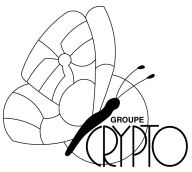
Quisquater's algorithm results in an increase of the modulus size. At first sight, this may appear as an issue but, for protected implementations, it turns out that it is not. The usual countermeasure to thwart DPA-type attacks [7] consists in randomizing the process for evaluating a cryptographic computation. Applied to the computation of the above RSA signature, this can be achieved as

$$S^* = (\mu(m) + r_1 N)^{d + r_2\phi(N)} \bmod N'$$

for certain random integers r_1 and r_2, and where ϕ denotes Euler's totient function (i.e., $\phi(N) = \#\mathbb{Z}_N^*$). Moreover, it is even possible to freely randomize the value of N' by randomly choosing the normalization factor δ as one of the valid $\delta_{b,t}$'s when defining N'. Signature S is then recovered as $S = (\delta S^* \bmod N')/\delta$.

Acknowledgments

I chose to discuss Quisquater's algorithm not only because it is one of the best known methods to evaluate a modular exponentiation but also because it is the first topic I worked on as a graduate student under the supervision of Jean-Jacques. This was in the early nineties when the UCL Crypto Group was formed. Since then, many students benefited from the advices of Jean-Jacques, the scientist of course and, maybe more importantly, the person. *Merci Jean-Jacques!*



References

1. de Waleffe, D., Quisquater, J.J.: CORSAIR: A smart card for public-key cryptosystems. In: Menezes, A.J., Vanstone, S.A. (eds.) Advances in Cryptology – CRYPTO '90. Lecture Notes in Computer Science, vol. 537, pp. 503–513. Springer-Verlag (1991)
2. Dhem, J.F.: Design of an efficient public-key cryptographic library for RISC-based smart cards. Ph.D. thesis, Université catholique de Louvain, Louvain-la-Neuve (May 1998)
3. Dhem, J.F., Joye, M., Quisquater, J.J.: Normalisation in diminished-radix modulus transformation. Electronics Letters 33(23), 1931 (1997)
4. Ferreira, R., Malzahn, R., Marissen, P., Quisquater, J.J., Wille, T.: FAME: A 3rd generation coprocessor for optimising public-key cryptosystems in smart-card applications. In: Hartel, P.H., et al. (eds.) Proceedings of the 2nd Smart Card Research and Advanced Applications Conference (CARDIS '96). pp. 59–72 (1996)
5. Joye, M.: Arithmétique algorithmique: Application au crypto-système à clé publique RSA. Master's thesis, Université catholique de Louvain, Louvain-la-Neuve (Jan 1994)
6. Knuth, D.E.: The Art of Computer Programming, Volume 2: Seminumerical Algorithms. Addison-Wesley, 3rd edn. (1997)
7. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) Advances in Cryptology – CRYPTO '99. Lecture Notes in Computer Science, vol. 1666, pp. 388–397. Springer-Verlag (1999)
8. Orton, G., Peppard, L., Tavares, S.: Design of a fast pipelined modular multiplier based on a diminished-radix algorithm. Journal of Cryptology 6(4), 183–208 (1993)
9. Quisquater, J.J.: Fast modular exponentiation without division. Rump session of EUROCRYPT '90, Aarhus, Denmark (May 21–24, 1990)
10. Quisquater, J.J.: Procédé de codage selon la méthode dite RSA par un microcontrôleur et dispositifs utilisant ce procédé. Demande de brevet français, No. de dépôt 90 02274 (Feb 1990)
11. Quisquater, J.J.: Encoding system according to the so-called RSA method, by means of a microcontroller and arrangement implementing this system. U.S. Patent #5,166,978 (Feb 1991)
12. Quisquater, J.J., de Waleffe, D., Bournas, J.P.: CORSAIR: A chip with fast RSA capability. In: Chaum, D. (ed.) Smart Card 2000. pp. 199–205. Elsevier Science Publishers (1991)
13. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21(2), 120–126 (1978)