Security analysis of the Gennaro-Halevi-Rabin signature scheme

No Author Given

Abstract. We exhibit a feasible attack against a signature scheme recently proposed by Gennaro, Halevi and Rabin [8]. The scheme's security is based on two assumptions namely the strong RSA assumption and the existence of division-intractable hash-functions. For the latter, the authors conjectured a security level exponential in the hash-function's digest size whereas our attack is sub-exponential with respect to the digest size. Moreover, since the new attack is optimal, the length of the hash function can now be rigorously dimensionned. In particular, to get a security level equivalent to 1024-bit RSA, one should use a digest size of approximately 1024 bits instead of the 512 bits suggested in [8].

Key words : Gennaro-Halevi-Rabin signature scheme, Strong RSA problem, division intractability.

1 Introduction

This paper analyses the security of a signature scheme presented by Gennaro, Halevi and Rabin at Eurocrypt'99 [8]. The concerned scheme (hereafter GHR) uses a standard (public) RSA modulus n and a random public base s. To sign a message m, the signer computes the e-th root modulo n of s with $e = H(m)$ where H is a hash function. A signature σ is verified with $\sigma^{H(m)} = s \bmod n$.

The scheme is proven to be existentially unforgeable under chosen message attacks under two assumptions : the strong RSA assumption and the existence of division-intractable hash-functions. The originality of the construction lies in the fact that security can be proven without using the random oracle model [3].

In this paper we focus on the second assumption, *i.e.* the existence of division-intractable hash-functions. Briefly, a hash function is division-intractable if it is computationally infeasible to exhibit a hash value that divides the product of other hash values. Assimilating the hash function to a random oracle, it is conjectured [8] based on numerical experiments that the number of k-bits digests needed to find one that divides

the product of the others is approximately $2^{k/8}$. Here we show that the number of necessary hash-values is actually subexponential in k, namely $\exp((\sqrt{2\log 2}/2 + \circ(1))\sqrt{k \log k})$.

The paper is organised as follows. We briefly start by recalling the GHR scheme and its related security assumptions. Then we describe our attack, evaluate its asymptotical complexity and, by extrapolating from running times observed for small digest sizes, estimate the *practical* complexity of our attack. We also show that the attack is asymptotically optimal and estimate from a simple heuristic model the minimal complexity of finding a hash value that divides the product of the others. Finally, we show how to improve our attack for the specific hash function proposed in [8].

2 The Gennaro-Halevi-Rabin signature scheme

2.1 Construction

The GHR scheme is a hash-and-sign scheme that shares some similarities with the standard RSA signature scheme :

Key generation : Generate an RSA modulus $n = pq$, product of two primes p and q of about the same length and a random element $s \in \mathbb{Z}_n^*$. The public key is $\{n, s\}$ and the private key is $\{p, q\}$.

Signature generation : To sign a message m, compute an odd exponent $e = H(m)$. The signature σ is :

$$\sigma = s^{e^{-1} \bmod \phi(n)} \bmod n$$

where $\phi(n) = (p-1)(q-1)$ is Euler's function.

Signature verification : Check that :

$$\sigma^{H(m)} = s \bmod n$$

2.2 GHR's security proof

The originality of the GHR signature scheme lies in the fact that its security can be proven without using the random oracle model. Instead, the hash function must satisfy some well defined computational assumptions [8]. In particular, it is assumed that the hash function family is division-intractable.

Definition 1 (Division intractability [8]). *A hashing family $\mathcal{H}$ is division intractable if finding $h \in \mathcal{H}$ and distinct inputs $X_1, \ldots X_n$,Y such that $h(Y)$ divides the product of the $h(X_i)$ values is computationally infeasible.*

The GHR signature scheme is proven to be existentially unforgeable under an adaptive chosen message attack, assuming the strong RSA conjecture.

Conjecture 1 (Strong-RSA [2]). *Given a randomly chosen RSA modulus n and a random $s \in \mathbb{Z}_n^*$, it is infeasible to find a pair (e, r) with $e > 1$ such that $r^e = s \mod n$.*

An opponent willing to forge a signature without solving the strong-RSA problem must find messages $m, m_1, \ldots, m_r$ such that $H(m)$ divides the least common multiple of $H(m_1), \ldots, H(m_r)$. In this case, we say that a *division-collision* for H was exhibited. Using Euclid's algorithm the opponent can obtain $a_1, \ldots, a_r, k$ such that :

$$\frac{a_1}{H(m_1)} + \ldots + \frac{a_r}{H(m_r)} = \frac{1}{\mathrm{lcm}(H(m_1), \ldots, H(m_r))} = \frac{1}{k \times H(m)}$$

and forge the signature σ of m from the signatures σ_i of messages m_i by :

$$\sigma = (\prod_{i=1}^{r} \sigma_i^{a_i})^k \bmod n$$

If $\mathcal{H}$ is division-intractable then it is infeasible for a polynomially bounded attacker to find a division collision for a hash function in $\mathcal{H}$. In particular, a random oracle is shown to be division-intractable in [8].

A natural question that arises is the complexity of finding a division collision, if one assumes that the hash function behaves as a random oracle. This question will condition the choice of the signature scheme's parameters. [8] conjectures (based on numerical experiments) a security level exponential is the length of the hash function namely that the number of hash calls necessary to obtain a division-collision behaves asymptotically as $2^{k/8}$ where k is the hash size. To get equivalent security to a 1024-bit RSA, [8] suggests to use a hash size of about 512 bits. In the next section, we exhibit a sub-exponential forgery and study its consequences for the recommanded digest size.

3 A sub-exponential attack

The outline of our attack is the following : we first look among many digests to find a smooth one, *i.e.* a hash value that factors into moderate-size primes p_i. Then for each of the p_i we look for a hash value divisible by p_i, so that the smooth hash value divides the least common multiple of the other hash values.

3.1 Background on smooth numbers

Let y be a positive integer. We say that an integer z is y-smooth if each prime dividing z is $\leq y$. An integer z is y-powersmooth if all primes powers dividing z are $\leq y$. Letting $\psi(x, y)$ denote the number of integers $z \leq x$ such that z is y-smooth, the following theorem gives an estimate on the density of smooth numbers [5] :

Theorem 1. *If ϵ is an arbitrary positive constant, then uniformly for $x \geq 10$ and $y \geq (\log x)^{1+\epsilon}$,*

$$\psi(x, y) = xu^{-u+\circ(u)} \quad \text{as } x \to \infty$$

where $u = (\log x)/(\log y)$.

In particular, setting $y = L_x[\beta] = \exp\left((\beta + \circ(1))\sqrt{\log x \log\log x}\right)$, the probability that a random integer between one and x is $L_x[\beta]$-smooth is :

$$\frac{\psi(x, y)}{x} = L_x[-\frac{1}{2\beta}]$$

The proportion of squarefree integers is asymptotically $6/\pi^2$. Letting $\psi_1(x, y)$ denote the number of *squarefree* integers $z \leq x$ such that z is y-smooth, theorem 3 in [9] implies that the same proportion holds for y-smooth numbers :

$$\psi_1(x, y) \sim \frac{6}{\pi^2}\psi(x, y) \tag{1}$$

under the growing condition :

$$\frac{\log y}{\log\log x} \to \infty, \qquad (x \to \infty)$$

Letting $\psi'(x, y)$ denote the number of integers $z \leq x$ such that z is y-powersmooth, we have for all $x, y > 0$:

$$\psi_1(x, y) \leq \psi'(x, y) \leq \psi(x, y)$$

which using (1) shows that for $y = L_x[\beta]$, the probability that a random integer between one and x is y-powersmooth is :

$$\frac{\psi'(x,y)}{x} = L_x[-\frac{1}{2\beta}]$$

3.2 The attack

In the following we assimilate the hash function to a random oracle which outputs random integers between one and x. Given a set $\mathcal{S}$ of random integers, we say that $\{e, e_1, \ldots, e_r\}$ is a *division-collision* for $\mathcal{S}$ if $e, e_1, \ldots, e_r \in \mathcal{S}$ and e divides the least common multiple of $e_1, \ldots, e_r$.

Theorem 2. *Let $\mathcal{S} = \{e_1, \ldots, e_v\}$ be a set of v random integers uniformly distributed between one and x. If $v = L_x[\sqrt{2}/2]$ then we can find a division-collision for $\mathcal{S}$ in time $L_x[\sqrt{2}/2]$.*

Proof. Using the following algorithm with $\beta = \sqrt{2}/2$, a division-collision is found in time $L_x[\sqrt{2}/2]$.

Finding a division-collision Algorithm :

Input : a set $\mathcal{S} = \{e_1, \ldots, e_v\}$ of $v = L_x[\sqrt{2}/2]$ random integers between one and x.

Output : a division-collision for $\mathcal{S}$.

Step 1 : Look for a powersmooth $e_k \in \mathcal{S}$ with respect to $y = L_x[\beta]$, using Pollard-Brent's Method [4] or Lenstra's Elliptic Curve Method (ECM) [10] to obtain :

$$e_k = \prod_{i=1}^{r} p_i^{\alpha_i} \quad \text{with} \quad {p_i}^{\alpha_i} \leq y \text{ for } 1 \leq i \leq r \tag{2}$$

Step 2 : For each prime factor p_i look for $e_{j(i)} \in \mathcal{S}$ with $j(i) \neq k$ such that $e_{j(i)} = 0 \bmod p_i^{\alpha_i}$, whereby :

$$e_k | \ \mathrm{lcm}(e_{j(1)}, \ldots, e_{j(r)})$$

Pollard-Brent's method finds a factor p of n in $\mathcal{O}(\sqrt{p})$ expected running time, whereas the ECM extracts a factor p of n in $L_p[\sqrt{2}]$ expected running time. Using Pollard-Brent's method at step 1, a $L_x[\beta]$-powersmooth $H(m)$ is found in expected $L_x[1/(2\beta)] \times L_x[\beta/2] = L_x[1/(2\beta) + \beta/2]$ time. Using the ECM a $L_x[\beta]$-powersmooth $H(m)$ is found in $L_x[1/(2\beta)] \times L_x[\circ(1)] = L_x[1/(2\beta)]$ operations. Since $p_i^{\alpha_i} \leq y$, the second stage requires less than $y = L_x[\beta]$ operations.

The overall complexity of the algorithm is thus minimal for $\beta = 1$ when using Pollard-Brent's method, resulting in a time complexity of $L_x[1]$. The ECM's minimum complexity occurs for $\beta = \sqrt{2}/2$ giving a time complexity of $L_x[\sqrt{2}/2]$.

Moreover, the following theorem shows that the previous algorithm is optimal.

Theorem 3. *Let $\mathcal{S} = \{e_1, \ldots, e_v\}$ be a set of v random integers uniformly distributed between one and x. If $v = L_x[\alpha]$ with $\alpha < \sqrt{2}/2$, then the probability that one integer in $\mathcal{S}$ divides the least common multiple of the others is negligible.*

Proof. See appendix A.

Consequently, assuming that the hash function behaves as a random oracle, the number of hash values necessary to exhibit a division-collision with non-negligible probability is asymptotically $L_x[\sqrt{2}/2]$ and this can be done in time $L_x[\sqrt{2}/2]$.

3.3 Attack's practical running time

Using the ECM, the attack has an expected time complexity of :

$$L_x[\sqrt{2}/2] = \exp\left((\frac{\sqrt{2}}{2} + \circ(1))\sqrt{\log x \log\log x}\right) \tag{3}$$

It appears difficult to give an accurate formula for the attack's practical running time since one would have to know the precise value of the term $\circ(1)$ in equation (3). However, extrapolating from (3) and the running times observed for small hash sizes, we have estimated the number of hash calls necessary to mount the attack. We obtained the following estimate for our implementation : the number of hash calls necessary is approximately given by :

$$\exp\left((\frac{\sqrt{2}}{2} + 0.62 \times (\log x)^{-0.31})\sqrt{\log x \log\log x}\right)$$

The results summarized in table 3.3 suggest that in order to reach a security level equivalent to a 1024-bit RSA, digests should also be approximately 1024-bit long.

digest size	$\log_2$ complexity
128	25
256	36
512	53
640	60
768	66
1024	77

Table 1. $\log_2$ complexity (number of hash calls) of the attack for various digest size.

4 Division-intractability in the random oracle model

In the previous section we have estimated the number of digests necessary to exhibit a division-collision with the ECM, from its asymptotic running time (3) and the observed running times for small hash sizes. Needless to say, our estimate depends on the practical implementations of the hash function and the ECM : the slower is the ECM implementation, the higher must be the number of hash-calls to exhibit a division-collision. In this section we derive heuristically the minimal number of digests necessary to find a division-collision, in the random oracle model.

The probability that given a set of random integers, one divides the least common multiple of the others, can be derived from a simple heuristic model called *random bissection*. In this model, the relative length of the first prime factor of a random number is obtained asymptotically by choosing a random λ uniformly in $[0,1]$, and then proceeding recursively with a random integer of relative size $1-\lambda$. This model is used in [1] to compute a recurrence for $F(\alpha) = \rho(1/\alpha)$, the asymptotic probability that all prime factors of n are smaller than n^α. In the above formula ρ is *Dickman's rho function* defined for real $t \geq 0$ by the relation [6] :

$$\rho(t) = \begin{cases} 1 & \text{if } 0 \leq t \leq 1 \\ \rho(n) - \displaystyle\int_n^t \frac{\rho(v-1)}{v} dv & \text{if } n \leq t \leq n+1 \end{cases} \tag{4}$$

For an n^α-smooth number n, all relative lengths λ chosen by random bisections are smaller than α, and the remaining integer of relative size $1-\lambda$ is also n^α-smooth. Consequently, we obtain equation (5) from which we derive (4) by substituting $t = (1-\lambda)/\alpha$ and $x = 1/\alpha$.

$$F(\alpha) = \int_0^\alpha F(\frac{\alpha}{1-\lambda}) d\lambda \tag{5}$$

Let $P_{\mathrm{DIV}}[u,v]$ denote the probability that a random u-bit integer a divides the least common multiple of 2^v other u-bit random integers. Let p be a prime factor of a of relative size λ. The probability P that p divides a u-bit integer is roughly $1/p$. Consequently, the probability P that p divides the least common multiple of 2^v u-bit integers is approximately :

$$P = 1 - (1 - \frac{1}{p})^{2^v} \simeq \begin{cases} 1 & \text{if } p \ll 2^v \\ \\ \dfrac{2^v}{p} & \text{if } p \gg 2^v \end{cases}$$

In the following, for a prime factor p of relative size λ, we assume that :

$$P = \begin{cases} 1 & \text{if } \lambda\, u \leq v \\ \\ 2^{v-\lambda\, u} & \text{if } \lambda\, u > v \end{cases}$$

Conditioning on the relative length λ of the first factor of a, we get

$$P_{\mathrm{DIV}}[u,v] = \begin{cases} 1 & \text{if } u \leq v \\ \\ \displaystyle\int_0^{\frac{v}{u}} P_{\mathrm{DIV}}[u(1-\lambda),v]d\lambda + & \\ \displaystyle\int_{\frac{v}{u}}^1 P_{\mathrm{DIV}}[u(1-\lambda),v]2^{v-\lambda\, u}d\lambda & \text{if } u > v \end{cases}$$

Letting $S(\alpha, v) = P_{\mathrm{DIV}}[\alpha v, v]$, we have :

$$S(\alpha,v) = \begin{cases} 1 & \text{if } \alpha \leq 1 \\ \\ \dfrac{1}{\alpha}\displaystyle\int_0^1 S(\alpha - s, v)ds + \dfrac{1}{\alpha}\int_1^\alpha S(\alpha - s, v)2^{v(1-s)}ds & \text{if } \alpha > 1 \end{cases}$$

We obtain :

$$\frac{\partial S}{\partial \alpha}(\alpha,v) = -v\log 2\Big(S(\alpha,v) - \frac{1}{\alpha}\int_0^1 S(\alpha - s, v)ds\Big)$$

and :

$$\frac{\partial^2 S}{\partial \alpha^2}(\alpha,v) = -\frac{v\log 2}{\alpha}S(\alpha-1,v) - (\frac{1}{\alpha} + v\log 2)\frac{\partial S}{\partial \alpha}(\alpha,v) \qquad (6)$$

$S(\alpha, v)$ for $\alpha \geq 0$ is thus defined as the solution with continous derivative of the delay differential equation (6) with initial condition $S(\alpha, v) = 1$ for $0 \leq \alpha \leq 1$.

A division-collision occurs if at least one integer divides the least common multiple of the others. We assume these events to be statistically independent. Consequently, the probability $P_{\mathrm{DC}}[u, v]$ that a division-collision occurs in a set of 2^v random u-bit integer can be expressed as :

$$P_{\mathrm{DC}}[u, v] = 1 - \left(1 - S(\frac{u}{v}, v)\right)^{2^v} \tag{7}$$

The number a random u-bit integer required to obtain a division collision with probability β is thus 2^v where v is obtained by solving $P_{\mathrm{DC}}[u, v] = \beta$ is equation (7).

The function $S(\alpha, v)$ can be computed by numerical integration from (6) and $S(\alpha, v) = 1$ for $0 \leq \alpha \leq 1$. We used Runge-Kutta method of order 4 to solve differential equation (6). We summarize in table 4 the number of u-bit integers required to obtain a division-collision with probability 1/2. One can see that the number of integers needed to obtain a division-collision (table 4) is actually smaller than the number of hash calls necessary to mount the attack of section 3.2 (table 3.3).

integer size in bits	$\log_2$ number of integers
128	18
256	28
512	43
640	50
768	56
1024	67
1280	76

Table 2. $\log_2$ number of random integers required to obtain a division-collision with probability 1/2 as a function of their size.

In [8] numerical experiments were performed to estimate the number of integers needed to get a division-collision with probability 10^{-2}. Figure 1 shows on a base two logarithmic scale the requested number of integers as a function of their size in bits for :

- The numerical experiments in [8] depicted by crosses.
- The $2^{k/8}$ conjecture in [8], depicted by dots.
- The actual value, computed from (7), depicted by a plain line.

The minimal number of digests necessary to obtain a division-collision with sufficient probability is thus much smaller than the $2^{k/8}$ conjectured in [8].

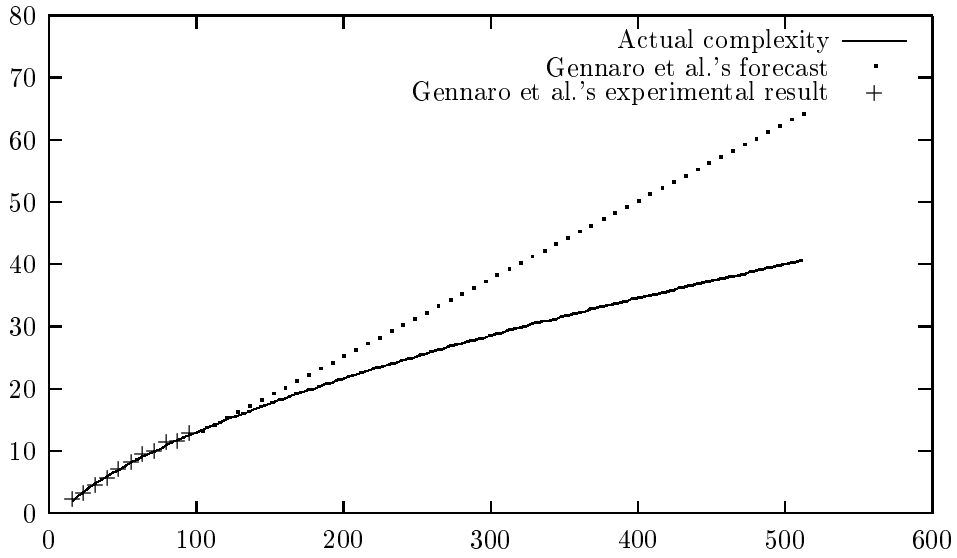

Fig. 1. $\log_2$ complexity (number of hash calls) of finding a division-collision with probability $\geq 10^{-2}$ as a function of the digest size for Gennaro *et al.*'s forecast vs. actual.

5 Faster attacks on GHR's specific hash function

Finally, we show that our attack can even be improved for the *specific* hash function suggested in [8] :

$$H(R_1; R_2; R_3; R_4; m) = f_1(m, R_1) \cdot f_2(m, R_2) \cdot f_3(m, R_3) \cdot f_4(m, R_4)$$

where $a \cdot b$ stands for the concatenation of a and b and :

$$\begin{aligned} f_1(m, R) &= 1 \cdot \mathrm{SHA}(m \cdot 1 \cdot R) \\ f_2(m, R) &= \quad \mathrm{SHA}(m \cdot 2 \cdot R) \\ f_3(m, R) &= \quad \mathrm{SHA}(m \cdot 3 \cdot R) \\ f_4(m, R) &= \quad \mathrm{SHA}(m \cdot 4 \cdot R) \cdot 1 \end{aligned}$$

To show that this particular function can be broken faster than the general case described in section 3.2, we use the attack described in section 3.2 as a black-box and feed it with shorter inputs, thereby reducing its running time.

Fix a target m and proceed as follows :

• Select a moderate-size (*e.g.* 40-bit) odd number ω, such that reductions modulo ω are simple. In practice we recommend $\omega = 2^{40} - 1$ for which reductions are simple.

• For $i = 1, 2, 3$ and 4, exhaustive search ℓ random $R_i[0],\ldots,R_i[\ell]$ such that $f_i(M, R_i[j]) = 0 \bmod \omega$.

We can now generate, by simple concatenation, ℓ^4 different hash values such that :

$$H(R_1; R_2; R_3; R_4; m) = 0 \bmod \omega$$

Since all digests have a common 40-bit multiple, the attack described in the previous sections can be run on 602-bit digests (the digests divided by ω) instead of 642-bit ones. Since for 602-bit hash values our attack is expected to require 2^{58} different hash values (instead of 2^{60} for 642-bit hash values), it appears that $\ell = 2^{14.5} \cong 23000$ and we can expect to hash $4 \times 23000 \times 2^{40} \cong 2^{56.5}$ different $R_i[j]$ values before being able to construct the requested 2^{58} hash values. Memory requirement is $4 \times 23000 = 92000$ table entries containing the various $R_i[j]$ values.

This shows that the security of the *specific* hash function suggested in section 5.2 of [8] is lower than the general subexponential case analyzed in the previous sections of this paper (where the hash function's output was assumed to be random). However, since 2^{58} hash values are required when using 602-bit integers instead of 2^{60} when using 642-bits integer, the attack's efficiency is only increased by a factor of 4 for this specific hash function.

6 Conclusion

We have analysed the security of the Gennaro-Halevi-Rabin signature scheme of Eurocrypt'99. In particular, we exhibited a sub-exponential attackal that forces to increase the security parameters beyond 512 or 642 bits up to appoximately 1024 bits in order to get a security equivalent to 1024-bits RSA. Another variant of the scheme described in [8] consists in generating prime digests only, by performing primality tests on the digests until a prime is obtained. In this case, a division-collision is equivalent to a collision in the hash function, but the signature scheme becomes less attractive from a computational standpoint.

References

1. E. Bach and R. Peralta, *Asymptotic semismoothness probabilities*, Mathematics of computation, vol. 65, no. 216, pp. 1701–1715, 1996.

2. N. Barić and B. Pfitzmann, *Collision-free accumulators and Fail-stop signature scheme without trees*, proceedings of Eurocrypt'97, LNCS vol. 1233, Springer-Verlag, 1997, pp. 480-494.
3. M. Bellare and P. Rogaway, *Random Oracles are practical : a paradigm for designing efficient protocols*, proceedings of the 1st CCCS, pp. 62-73. ACM press, 1993.
4. R. Brent, *An improved Monte Carlo factorization algorithm*, Nordisk Tidskrift för Informationsbehandling (BIT) 20 (1980) pp. 176-184.
5. E. Canfield, P. Erdös and C. Pomerance, *On a problem of Oppenheim concerning 'Factorisatio Numerorum'*, J. Number Theory, vol. 17, 1983, PP. 1-28.
6. K. Dickman, *On the frequency of numbers containing prime factors of a certain relative magnitude*, Arkiv för matematik, astronomi och fysik, vol. 22A, no. 10, pp. 1–14, 1930.
7. G. Hardy and E. Wright, *An introduction to the theory of numbers*, Fifth edition, Oxford, 1979, pp. 354-359, 368-370.
8. R. Gennaro, S. Halevi, T. Rabin, *Secure hash-and-sign signatures without the random oracle*, proceedings of Eurocrypt'99, LNCS vol. 1592, Springer-Verlag, 1999, pp. 123-139.
9. A. Ivić and G. Tenenbaum, *Local densities over integers free of large prime factors*, Quart. J. Math. Oxford (2), 37 (1986), pp. 401-417.
10. H. W. Lenstra, Jr., *Factoring integers with elliptic curves*, Ann. of Math. (2) 126 (1987) pp. 649-673.

A Proof of theorem 3

Proof. Let $\mathcal{S} = \{e_1, \ldots, e_v\}$ with $v = L_x[\alpha]$ and $\alpha < \sqrt{2}/2$ be a set of v random integers uniformly distributed between 1 and x. Denote by $P(v, x)$ the probability that one integer in $\mathcal{S}$ divides the least common multiple of the others and by B the event in which e_1 divides the least common multiple of $\{e_2, \ldots, e_v\}$. The proof's outline is the following : we consider the possible smoothness degrees of e_1 and compute the probability of B for each smoothness degree. Then we show that $\Pr[B]$ is smaller than $L_x[-\sqrt{2}/2 + \epsilon]$ for $\epsilon > 0$ and conclude that $P(v, x)$ is negligible.

The possible smoothness degrees of e_1 are denoted :

- Sm : e_1 is $L_x[\sqrt{2}/2]$-smooth. This happens with probability

$$\Pr[\mathrm{Sm}] = L_x[-\sqrt{2}/2]$$

and consequently :

$$\Pr[B \wedge \mathrm{Sm}] = \mathcal{O}(L_x[-\sqrt{2}/2]) \tag{8}$$

• $\mathrm{Sm}(\gamma, \epsilon)$: e_1 is $L_x[\gamma + \epsilon]$-smooth without being $L_x[\gamma]$ smooth, for $\sqrt{2}/2 < \gamma < \sqrt{2}$ and $\epsilon > 0$. This happens with probability :

$$\Pr[\mathrm{Sm}(\gamma, \epsilon)] = L_x[\frac{-1}{2 \times (\gamma + \epsilon)}] - L_x[\frac{-1}{2 \times \gamma}] = L_x[\frac{-1}{2 \times (\gamma + \epsilon)}] \qquad (9)$$

In this case, e_1 contains a prime factor greater than $L_x[\gamma]$, which appears in the factorization of another e_i with probability $\mathcal{O}(L_x[-\gamma])$. Consequently e_1 divides the least common multiple of $\{e_2, \ldots, e_v\}$ with probability :

$$\Pr[B|\mathrm{Sm}(\gamma, \epsilon)] = \mathcal{O}(L_x[\alpha - \gamma])$$

With (9) and $\gamma + \frac{1}{2(\gamma+\epsilon)} \geq \sqrt{2} - \epsilon$ for all $\gamma > 0$, we get :

$$\Pr[B \wedge \mathrm{Sm}(\gamma, \epsilon)] = \mathcal{O}(L_x[-\frac{\sqrt{2}}{2} + \epsilon]) \qquad (10)$$

• $\neg\mathrm{Sm}$: e_1 is not $L_x[\sqrt{2}]$-smooth. Consequently e_1 contains a factor greater than $L_x[\sqrt{2}]$ and thus :

$$\Pr[B \wedge \neg\mathrm{Sm}] = \mathcal{O}(L_x[\alpha - \sqrt{2}]) = \mathcal{O}(L_x[-\frac{\sqrt{2}}{2}]) \qquad (11)$$

Partitioning the segment $[\sqrt{2}/2, \sqrt{2}]$ into segments $[\gamma, \gamma + \epsilon]$ and using equations (8), (10) and (11), we get :

$$\Pr[B] = \mathcal{O}(L_x[-\frac{\sqrt{2}}{2} + \epsilon])$$

Since $\alpha < \sqrt{2}/2$ we can choose $\epsilon > 0$ such that $\sqrt{2}/2 - \alpha - \epsilon = \delta > 0$ and obtain :

$$P(v, x) = \mathcal{O}(L_x[\alpha - \sqrt{2}/2 + \epsilon]) = \mathcal{O}(L_x[-\delta])$$

which shows that $P(v, x)$ is negligible. □