Security Analysis of the Gennaro-Halevi-Rabin Signature Scheme

Jean-Sébastien Coron
Ecole Normale Supérieure
45 rue d'Ulm
Paris, F-75230, France
coron@clipper.ens.fr

David Naccache
Gemplus Card International
34 rue Guynemer
Issy-les-Moulineaux, F-92447, France
david.naccache@gemplus.com

Abstract. We exhibit an attack against a signature scheme recently proposed by Gennaro, Halevi and Rabin [9]. The scheme's security is based on two assumptions namely the strong RSA assumption and the existence of a division-intractable hash-function. 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 fixed. 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 [9].

1 Introduction

This paper analyses the security of a signature scheme presented by Gennaro, Halevi and Rabin at Eurocrypt'99 [9]. 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 [9] 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.

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 a RSA modulus $n = p \cdot q$, 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. In the random oracle model, the hash function is seen as an oracle which outputs a random value for each new query. Instead, the hash function must satisfy some well defined computational assumptions [9]. In particular, it is assumed that the hash function family is division-intractable.

Definition 1 (Division intractability [9]). *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 $r^e = s \bmod n$.*

An opponent willing to forge a signature without solving the strong-RSA problem can try to 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 :

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

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

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

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 [9].

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, *i.e.* outputs a random integer for each new query. This question will condition the choice of the signature scheme's parameters. [9] conjectures (based on numerical experiments) a security level exponential in the length of the hash function, namely that the number of hash calls necessary to obtain a division-collision is asymptotically $2^{k/8}$ where k is the digest size. To get equivalent security to a 1024-bit RSA, [9] suggests to use 512-bit digests. 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 $1 \leq z \leq x$ such that z is y-smooth, the following theorem gives an estimate of 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 $6/\pi^2$ [10]. Letting $\psi_1(x,y)$ denote the number of *squarefree* integers $1 \leq z \leq x$ such that z is y-smooth, theorem 3 in [10] 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 :

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

A squarefree y-smooth integer is y-powersmooth, so letting $\psi'(x,y)$ denote the number of integers $1 \leq z \leq x$ 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 there exist a probabilistic Turing machine which outputs a division-collision for $\mathcal{S}$ in time $L_x[\sqrt{2}/2]$ with non-negligible probability.*

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

An algorithm finding a division-collision :

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) [11] 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 :

$$e_k \mid \operatorname{lcm}(e_{j_1}, \ldots, e_{j_r})$$

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, an $L_x[\beta]$-powersmooth $H(m)$ is found in expected $L_x[1/(2\beta)] \cdot L_x[\beta/2] = L_x[1/(2\beta) + \beta/2]$ time. Using the ECM an $L_x[\beta]$-powersmooth $H(m)$ is found in $L_x[1/(2\beta)] \cdot L_x[\circ(1)] = L_x[1/(2\beta)]$ 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 The Attack's Practical Running Time

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

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

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 can estimate the time complexity for larger hash sizes.

We have experimented the attack on a Pentium 200 MHz for hash sizes of 128, 160, and 192 bits, using the MIRACL library [12]. In Table 1 we summarize the observed running time in seconds and the logarithm in base 2 of the number of operations (assuming that the Pentium 200 MHz performs $200 \cdot 10^6$ operations per second).

Table 1. Experimental running times in seconds and $\log_2$ complexity (number of operations) of the attack for various digest sizes

digest size in bits	time complexity in seconds	$\log_2$ complexity
128	$3.5 \cdot 10^2$	36
160	$3.6 \cdot 10^3$	39
192	$2.1 \cdot 10^4$	42

Assuming that the complexity of the attack (number of operations) can be expressed as $C \cdot \exp(\sqrt{2}/2\sqrt{\log x \log\log x})$, the experimental complexity for a 192-bits hash size gives $C = 6.1 \cdot 10^4$, from which we derive in Table 2 the estimated complexity for larger hash sizes. The estimate may be rather imprecise and only provides an order of magnitude of the attack's complexity. However, the results summarized in Table 2 suggest that in order to reach a security level equivalent to 1024-bit RSA, digests should also be approximately 1024-bit long. Finally, we describe in the full version of the paper [6] a slightly better attack for the particular hash function suggested in [9].

4 Minimal Number of Hash Calls Necessary to Obtain a Division-collision

In the previous section we have estimated the time complexity of the attack using the ECM, from its asymptotic running time (3) and the observed running times for small hash sizes. Consequently, our estimate depends on the practical implementations of the hash function and the ECM. However theorem 3 shows that there is a lower bound on the number of hash calls necessary to mount the attack : asymptotically the number of hash calls must be at least $L_x[\sqrt{2}/2]$ so

Table 2. Estimated $\log_2$ complexity (number of operations) of the attack for various digest sizes

digest size	$\log_2$ complexity (number of operations)
256	47
512	62
640	69
768	75
1024	86

that with non-negligible probability there exist a division-collision (*i.e.* one hash value divides the least common multiple of the others). In this section we obtain heuristically a more precise estimate of the minimal number of hash calls necessary to have a division-collision with given probability. As in the previous section we assume that the hash function behaves as a random oracle, *i.e.* it outputs a random integer for each new query. Consequently the problem is the following : given a set $\mathcal{S}$ of v random integers in $\{1, \ldots, x\}$, what is the probability $P(x, v)$ that one integer in $\mathcal{S}$ divides the least common multiple of the others ?

4.1 A Heuristic Model

The probability $P(x, v)$ can be derived from a simple heuristic model called *random bisection*. 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 a random x are smaller than x^α. In the above formula ρ is *Dickman's rho function* defined for real $t \geq 0$ by the relation [7] :

$$\rho(t) = \begin{cases} 1 & \text{if } \ 0 \leq t \leq 1 \\ \rho(n) - \displaystyle\int_n^t \frac{\rho(w-1)}{w} dw & \text{if } \ n \leq t \leq n+1 \text{ for } n \in \mathbb{N} \end{cases} \tag{4}$$

For an x^α-smooth integer x, the relative length λ chosen by random bisection is smaller than α, and the remaining integer of relative size $1 - \lambda$ is also x^α-smooth. Consequently, we obtain equation (5) from which we derive (4).

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

Let $Q(x, v)$ denote the probability that a random integer z comprised between one and x divides the least common multiple of v other random integers in $\{1, \ldots, x\}$. Let $X = \log_2 x$ and $V = \log_2 v$. Let p be a prime factor of z of

relative size λ (*i.e.* $p = x^\lambda$). The probability that p divides a random integer in $\{1, \ldots, x\}$ is roughly $1/p$. Consequently, the probability P that p divides the least common multiple of v random integers in $\{1, \ldots, x\}$ is roughly :

$$P = 1 - (1 - \frac{1}{p})^v \simeq 1 - \exp(\frac{-v}{p}) \text{ for large } p$$

If $\lambda \leq V/X$, then $p \leq v$ and we take $P = 1$. Otherwise if $\lambda \geq V/X$ then $p \geq v$ and we take $P = v/p$. Consequently, we obtain :

$$Q(x,v) = \begin{cases} 1 & \text{if } x \leq v \\ \int_0^{\frac{V}{X}} Q(x^{1-\lambda}, v)d\lambda + \int_{\frac{V}{X}}^1 Q(x^{1-\lambda}, v)\frac{v}{x^\lambda} & \text{if } x > v \end{cases}$$

Letting $S(\alpha, V) = Q(v^\alpha, v)$, we have :

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

We obtain :

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

$S(\alpha, V)$ for $\alpha \geq 0$ is thus defined as the solution with continuous 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 those events to be statistically independent. Consequently, we obtain :

$$P(x,v) \simeq 1 - \left(1 - S(\frac{X}{V}, V)\right)^v \quad (7)$$

4.2 Numerical Experiments

Table 3. Number of random integers required to obtain a division-collision with probability 1% as a function of their size (numerical experiments and heuristic model)

integer size	16	32	48	64	80	96
number of integers (experiments)	4	25	119	611	1673	7823
$\log_2$ number of integers (experiments)	2.0	4.6	6.9	9.3	10.7	12.9
$\log_2$ number of integers (model)	2.0	4.7	7.0	9.1	10.9	12.6

We performed numerical experiments to estimate the number of k-bit integers required so that a division-collision appears with good probability. We considered bit-lengths between $k = 16$ to $k = 96$ in increments of 16, and as in [9] for each bit length we performed 200 experiments in which we counted how many random integers were chosen until one divides the least common multiple of the others. As in [9], we took the second smallest result of the 200 experiments as an estimate of the number of integers required so that a division-collision appears with probability 1%. The results are summarized in Table 3.

Table 4. $\log_2$ number of random integers required to obtain a division-collision with probability 1% as a function of their size

integer size in bits	$\log_2$ number of integers
128	15.6
256	25.6
512	40.6
640	46.8
768	52.4
1024	63.2
1280	72.1

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 the differential equation (6). We summarize in Table 3 the $\log_2$ number of k-bit integers required to obtain a division-collision with probability 1% for $k = 16$ to $k = 96$, from the heuristic model. We see that the values predicted by the model are close to the experimental values. In Table 4 we use the model to estimate the number of k-bit integers required to obtain a division-collision with probability 1% for large values of k. As in section 3.3 we see that in order to get a security level of a 1024-bits RSA, one should use a hash function of size approximately 1024 bits.

5 Conclusion

We have analysed the security of the Gennaro-Halevi-Rabin signature scheme of Eurocrypt'99. In particular, we exhibited a sub-exponential attack that forces to increase the security parameters beyond 512 or 642 bits up to approximately 1024 bits in order to get a security level equivalent to 1024-bits RSA. Another variant of the scheme described in [9] 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 First Annual Conference on Computer and Commmunications Security, ACM, 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. J.S. Coron and D. Naccache, *Security analysis of the Gennaro-Halevi-Rabin signature scheme*, full version of this paper, available at `http://www.eleves.ens.fr:8080/home/coron`, 2000.
7. 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.
8. G. Hardy and E. Wright, *An introduction to the theory of numbers*, Fifth edition, Oxford, 1979, pp. 354-359, 368-370.
9. R. Gennaro, S. Halevi and T. Rabin, *Secure hash-and-sign signatures without the random oracle*, proceedings of Eurocrypt'99, LNCS vol. 1592, Springer-Verlag, 1999, pp. 123-139.
10. A. Ivić and G. Tenenbaum, *Local densities over integers free of large prime factors*, Quart. J. Math. Oxford (2), 37 (1986), pp. 401-417.
11. H. W. Lenstra, Jr., *Factoring integers with elliptic curves*, Ann. of Math. (2) 126 (1987) pp. 649-673.
12. M.I.R.A.C.L. library, Shamus Software Ltd., 94 Shangan Road, Ballymun, Dublin, Ireland.

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 one and x. Denote by $P(x, v)$ 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 $P(x, v)$ 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 :

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

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 :

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

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

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

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 :

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

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