부호화 해밍 웨이트를 이용한 가변 타원곡선 암호시스템의 안전성 향상

(Enhanced Security of Flexible Elliptic Curve Cryptosystems using Signed Hamming Weights)

이 문 규 †

(Mun-Kyu Lee)

요 약 스칼라 곱셈은 정수 k와 타원곡선 상의 한 점 P가 주어졌을 때 kP를 계산하는 연산이다. 스칼라 곱셈을 빠르게 하기 위한 일반적인 방법으로 Agnew, Mullin, Vanstone은 고정된 값의 해밍 웨이트를 갖는 스칼라 k를 이용하는 방법을 제안하였다. 본 논문에서는 고정된 값의 부호화 해밍 웨이트를 갖는 k를 이용하는 방법을 제안하고, 이 방법이 더 안전함을 보인다.

키워드 : 스칼라 곱셈, 타원 곡선, 비인접 표현형 (NAF), 부호화 해밍 웨이트

Abstract Scalar multiplication is to compute kP when an integer k and an elliptic curve point P are given. As a general method to accelerate scalar multiplication, Agnew, Mullin and Vanstone proposed to use k's with fixed Hamming weights. We suggest a new method that uses k's with fixed signed Hamming weights and show that this method is more secure.

Key words : scalar multiplication, elliptic curve, nonadjacent form (NAF), signed Hamming weight

1. Introduction

The use of elliptic curves in cryptography was suggested by Koblitz [1] and Miller [2], and extensive research has been done on elliptic curve cryptosystems (ECCs). The most time consuming operation in elliptic curve cryptosystems is scalar multiplication kP for an integer k and a point P. A well-known algorithm for this operation is to repeat doublings and additions, which we call the binary method. The binary method requires $\lceil \log k \rceil$ doublings and $HW(k)$ additions, where $HW(k)$ is the Hamming weight of k, i.e., the number of 1's in the binary representation of k.

Since $HW(k) = \lceil \log k \rceil / 2$ on the average, the average number of required point operations for a scalar multiplication kP is $3\lceil \log k \rceil / 2$.

· 본 연구는 과학기술부 신기술 융합 사업 M1-0326-08-0001의 지원으로 수행되었음

† 정 회 원 : 한국전자통신연구원 정보보호연구단
mklee@etri.re.kr

논문접수 : 2004년 6월 10일
심사완료 : 2004년 8월 9일

The binary method is the additive analogue of the repeated square-and-multiply algorithm for the exponentiation g^k in a general finite group. For an elliptic curve, however, one can use the fact that the complexity of a subtraction is almost the same as that of an addition. That is, the binary representation of k can be replaced with a signed binary representation which has fewer nonzero terms, and then scalar multiplication is done by repeating doublings and additions/subtractions. For example, if we represent k as the nonadjacent form (NAF) [3, 4], then the average number of nonzero terms is $\lceil \log k \rceil / 3$, and the average number of required point operations is reduced to $4\lceil \log k \rceil / 3$.

On the other hand, there are complex multiplication methods for fast scalar multiplication which use a special family of elliptic curves [5, 6, 7, 8]. However, we do not consider these methods, focusing on general methods which are applicable to any curve.

In many applications such as ECDSA signature

generation [9], we compute kP for randomly chosen k. In [10], Agnew, Mullin and Vanstone proposed to choose special k's that have small $HW(k)$ to reduce the number of additions. (We will call this approach the AMV method.) Scalar k generated by this method has lower security than a general one with the same length, since the size of search space is smaller.

However, the AMV method has an important application in an environment where various levels of security are required with one elliptic curve. For example, with an ECC hardware accelerator implementing only a single curve, $HW(k)$ can be used as a security parameter, i.e., we choose a relatively large $HW(k)$ to protect sensitive data but we use a smaller $HW(k)$ for less important data. Note that otherwise, one should have multiple curves to obtain many levels of security, which is not acceptable to space-constrained environments.

Motivated by the AMV method, we propose in this paper to use k's with small $SHW(k)$, where $SHW(k)$ is the signed Hamming weight of k, i.e., the number of nonzeros in the signed binary representation of k. We show that our method is more secure than that of [10] by showing that the size of search space for k in our method is much larger when the amount of computation is the same.

2. Proposed method

Our method is based on the nonadjacent form (NAF) of k, i.e., a signed binary representation with the property that no two consecutive digits are nonzero [3]. For example, '35' is uniquely represented as $(1, 0, 0, 1, 0, -1)$ in NAF, since $35=2^5+2^2-1$.

We want to generate a random k of length m in NAF such that $SHW(k)$ is equal to a predefined security parameter w. A naive approach is to randomly select w locations for nonzero digits out of m digits, and then to assign '1' or '-1' to each of these digits. (The remaining digits will be '0's.) However, k's generated in this way do not always satisfy the NAF property. This problem can be solved by a modification, where we use '10' and '-10' as single nonzero units instead of 1 and -1. Algorithm 1 implements this observation, and Fig.1 shows an example for $m=7$, $w=3$.

Algorithm 1. Random Generation of k

1: Initially there is an array of $m-w+1$ consecutive slots.
2: Assign two-digit binary number 10 to the first slot. (This is to guarantee that $k>0$ and that k has exactly m digits.)
3: Choose $w-1$ random slots out of the remaining $m-w$ slots and assign 10 or -10 randomly to each of them.
4: Assign 0 to each remaining slot.
5: Concatenate all slots to get a number k' with $m+1$ signed binary digits. Note that k' is even.
6: Set $k \leftarrow k'/2$.

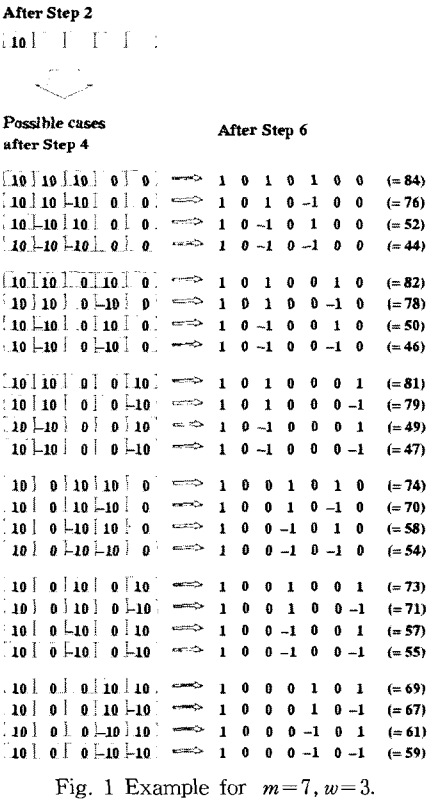

Fig. 1 Example for $m=7, w=3$.

Theorem 1. *Algorithm 1 forms a uniform distribution of k's that have exactly m digits in NAF and satisfy $k>0$, $SHW(k)=w$.*

Proof. Let A be the set of k's generated by the algorithm, and let B be the set of k's that have exactly m digits in NAF and satisfy $k>0$, $SHW(k)=w$.

Then it is sufficient to show that $A=B$; and that every element in A occurs with the same probability. It is easy to prove the second statement by the fact that every other choice in Step 3 is mapped into a unique k, and the probability of each choice is the same.

To prove that $A=B$, we only have to show that $B\subset A$. (It is straightforward to show that $A\subset B$.) We will show that any element in B is also in A. First, note that an element l in B is of the following form:

$$l = 1\overbrace{0\cdots0}^{n_1\text{zeros}} \pm 1 \overbrace{0\cdots0}^{n_2\text{zeros}} \pm 1 \cdots \pm 1 \overbrace{0\cdots0}^{n_w\text{zeros}},$$

where n_i's satisfy

$$\begin{aligned} &n_1+n_2+\cdots+n_w=m-w,\\ &n_1, n_2, \ldots, n_{w-1}\geq 1,\ n_w\geq 0. \end{aligned} \tag{1}$$

If we set $l'=2l$, then l' can be written as

$$l' = 1\overbrace{0\cdots0}^{n_1\text{zeros}} \pm 1 \overbrace{0\cdots0}^{n_2\text{zeros}} \pm 1 \cdots \pm 1 \overbrace{0\cdots0}^{(n_w+1)\text{zeros}}$$

Now, we partition l' as

$$l' = \boxed{10}\overbrace{\boxed{0}\cdots\boxed{0}}^{(n_1-1)\text{ zero slots}}\boxed{\pm10}\overbrace{\boxed{0}\cdots\boxed{0}}^{(n_2-1)\text{ zero slots}}\boxed{\pm10}\cdots\boxed{\pm10}\overbrace{\boxed{0}\cdots\boxed{0}}^{n_w\text{ zero slots}}$$

Then, it is easy to see that any l' of the above form can be generated by Steps 1-4 of Algorithm 1. Note that the number of zero slots is

$$(n_1-1)+(n_2-1)+\cdots+(n_{w-1}-1)+n_w=(m-w)-(w-1)$$

by (1), and the number of nonzero slots is w. Thus the total number of slots is $m-w+1$, which is the same as the value given in the initial step of Algorithm 1. Hence, for every element $l\in B$, we can show that $l\in A$. This completes the proof. □

3. Security

The security of elliptic curve schemes is based on the difficulty of the elliptic curve discrete logarithm problem (ECDLP), i.e., finding k when points P and $Q=kP$ are given. The best general-purpose algorithm known for the ECDLP is the elliptic curve version of Pollard's ρ algorithm [11], which uses a pseudo-random sequence generated from P and Q. Its expected running time is $O(\sqrt{n})$, and the amount of required memory is a constant, where n is the order of P. (Note that $k\approx n$.) The pseudo-random sequence in Pollard's ρ algorithm does not seem to contain any useful information about $HW(k)$ and $SHW(k)$. Hence, the expected running time of Pollard's ρ algorithm is still $O(\sqrt{n})$ for k's with specific $HW(k)$ or $SHW(k)$, although the size of search space for these k's is much smaller than n. Thus, our selection method and the method in [10] seem to be as secure against Pollard's ρ attack as random selection of general k.

The next choice of an attacker is the baby-step giant-step algorithm, which is a time-memory trade-off of the exhaustive search. Actually, this is the most powerful attack in the context of fixed HW and SHW, if the attacker has sufficient memory. Now, we describe several baby-step giant-step algorithms that can be applied to our setting, and we analyze the security of our method. (Although most of the known results deal with only the discrete logarithm problem (DLP), they can be easily transformed into the algorithms for ECDLP.)

The original and general baby-step giant-step algorithm is the Shanks' method (see [12], pp. 9, 575-576). It has time complexity $O(\sqrt{n})$ (if hashing is used), where n is the group order. Heiman [13] proposed the first baby-step giant-step algorithm to search the scalar space for k's that have m bits and a fixed Hamming weight w. Coppersmith ([14], p.128) and Stinson [15] observed that this space can be searched deterministically in $O\left(m\binom{m/2}{w/2}\right)$ steps by dividing k into two equal pieces so that the Hamming weight of each piece is $w/2$. Stinson also showed that the average-case complexity of this algorithm is $O\left(w\sqrt{w}(\log m)\binom{m/2}{w/2}\right)$, and that there is a Las Vegas algorithm with complexity $O\left(\sqrt{w}\binom{m/2}{w/2}\right)$. Ignoring some minor factors, we get $\binom{m/2}{w/2}$ for all of these cases, and this is approximately the square-root of the size of search space[1] for k, i.e., $\binom{m-1}{w-1}$. Therefore, the baby-step

1) In [14] and [15], the size of search space is $\binom{m}{w}$, which is from the setting where the most significant digit can be zero. However, this slight difference almost does not affect our overall computation.

giant-step algorithms given in [14] and [15] are square-root algorithms.

The situation for the fixed signed Hamming weight case is the same, i.e., there exists a square-root algorithm. We will show this fact by computing the size of search space and the complexity of algorithm, and then by comparing these two values. First, the number of k's that have m signed binary digits and satisfy $SHW(k)=w$ is [2] $\binom{m-w}{w-1}\times 2^{w-1}$. Using Stirling's formula, $n!\approx\left(\frac{n}{e}\right)^n\sqrt{2\pi n}$, we have

$$\begin{aligned}\binom{m-w}{w-1}\times 2^{w-1} &= \frac{(m-w)!\cdot 2^{w-1}}{(m-2w+1)!\,(w-1)!}\\ &\approx \frac{\left(\frac{m-w}{e}\right)^{m-w}\cdot\sqrt{2\pi(m-w)}\cdot 2^{w-1}}{\left(\frac{m-2w+1}{e}\right)^{m-2w+1}\cdot\sqrt{2\pi(m-2w+1)}\cdot\left(\frac{w-1}{e}\right)^{w-1}\cdot\sqrt{2\pi(w-1)}}\\ &= \frac{(m-w)^{m-w}\cdot 2^{w}}{(m-2w+1)^{m-2w+1}\cdot(w-1)^{w-1}}\cdot\frac{\sqrt{m-w}}{2\sqrt{2\pi(w-1)(m-2w+1)}}.\end{aligned} \tag{2}$$

Next, we consider the time complexity of a baby-step giant-step algorithm to find an m-digit k with $SHW(k)=w$. As in [14] and [15], we can use the time-memory trade-off approach, by dividing k into two equal pieces[3] so that the signed Hamming weight of each piece is $w/2$. Then the time complexity is $\binom{m/2-w/2}{w/2-1}\times 2^{w/2}$, if some minor factors are ignored.[4] Using Stirling's formula again, we have

$$\begin{aligned}&\binom{m/2-w/2}{w/2-1}\times 2^{w/2}\\ &\approx\left(\frac{(m-w)^{m-w}\cdot 2^{w}}{(m-2w+2)^{m-2w+2}\cdot(w-2)^{w-2}}\right)^{1/2}\\ &\quad\cdot\frac{\sqrt{m-w}}{\sqrt{\pi(w-2)(m-2w+2)}}.\end{aligned} \tag{3}$$

Comparing (2) and (3), we can see that the time complexity is approximately the square-root of the size of search space.

We have seen that there exist square-root algorithms solving the ECDLP for both types of scalars with fixed HW and fixed SHW, and that these are the best attacks. Therefore, it makes sense that we compare the security of these two types of scalars by directly comparing the sizes of search spaces.

Table 1 shows that the size of search space of our method is much larger than that of [10] when these two methods are applied to the same values of m and w in typical settings. (Note that the required amounts of computation for a scalar multiplication are the same for both methods, if m and w are fixed.) In another point of view, our method uses less computation than that of [10] if the same security level is required. For example, the size of search space for $m=160$, $w=40$ of [10] has the same order as that for $m=160$, $w=30$ of our method which uses 25% fewer additions or subtractions (excluding doublings).

Table 1 The sizes of search spaces for k.

		AMV [10]	Our method
		$\binom{m-1}{w-1}$	$\binom{m-w}{w-1}\times 2^{w-1}$
m=160	w=20	1.8×10^{24}	7.2×10^{28}
	w=30	5.2×10^{31}	4.2×10^{37}
	w=40	2.2×10^{37}	3.1×10^{43}
m=192	w=20	7.1×10^{25}	4.6×10^{30}
	w=30	1.7×10^{34}	5.0×10^{40}
	w=40	6.9×10^{40}	1.6×10^{48}
m=224	w=20	1.5×10^{27}	1.4×10^{32}
	w=30	2.1×10^{36}	1.5×10^{43}
	w=40	5.5×10^{43}	7.5×10^{51}
	w=50	6.4×10^{49}	3.2×10^{58}

4. Discussion

We proposed to choose special k's with fixed small $SHW(k)$ when computing kP, and we showed that it is more secure than the original method that uses k's with fixed $HW(k)$. Note that our method inherits a desirable property, i.e., flexible security, from the original method. Finally, we remark that the overhead to implement our method (Algorithm 1) is negligible.

References

[1] Koblitz, N., "Elliptic curve cryptosystems," Mathematics of Computation, Vol.48, pp.203-209, 1987.
[2] Miller, V., "Use of elliptic curves in crypto-

2) Note that we do not multiply 2^w, but multiply 2^{w-1}, since the most significant digit is always '1'.
3) For convenience, we assume m and w are even. If this is not the case, the algorithms will be altered in a straightforward manner.
4) This time, we don't multiply $2^{w/2-1}$, but multiply $2^{w/2}$, since the most significant digit could be '-1' in each piece.

graphy," CRYPTO '85, LNCS, Vol.218, pp.417-428, Springer, 1986.
[3] Morain, F. and Olivos, J., "Speeding up the computations on an elliptic curve using addition-subtraction chains," Theoretical Informatics and Applications, Vol.24, pp.531-543, 1990.
[4] Solinas, J.A., "An improved algorithm for arithmetic on a family of elliptic curves," CRYPTO '97, LNCS, Vol.1294, pp.357-371, Springer, 1997.
[5] Koblitz, N., "CM-curves with good cryptographic properties," CRYPTO '91, LNCS, Vol.576, pp.279-287, Springer, 1991.
[6] Smart, N.P., "Elliptic curve cryptosystems over small fields of odd characteristic," Journal of Cryptology, Vol.12, pp.141-151, 1999.
[7] Gallant, R.P., Lambert, R.J., and Vanstone, S.A., "Faster point multiplication on elliptic curves with efficient endomorphisms," CRYPTO 2001, LNCS, Vol.2139, pp.190-200, Springer, 2001.
[8] Park, T.J., Lee, M.K., Kim, E., and Park, K., "A general expansion method using efficient endomorphisms," ICISC 2003, LNCS, Vol.2971, pp.112-126, Springer, 2004.
[9] ANSI X9.62, Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), 1999.
[10] Agnew, G.B., Mullin, R.C., and Vanstone, S.A., "An implementation of elliptic curve cryptosystems over $F_{2^{155}}$," IEEE Journal on Selected Areas in Communications, Vol.11, No.5, pp.804-813, 1993.
[11] Harper, G., Menezes, A., and Vanstone, S.A., "Public-key cryptosystems with very small key lengths," EUROCRYPT '92, LNCS, Vol.658, pp.163-173, Springer, 1993.
[12] Knuth. D.E., The Art of Computer Programming, Vol. 3: Sorting and Searching, Addison-Wesley, Reading, Mass., 1973.
[13] Heiman, R., "A note on discrete logarithms with special structure," EUROCRYPT '92, LNCS, Vol.658, pp.454-457, Springer, 1993.
[14] Menezes, A., van Oorschot, P.C., and Vanstone, S.A., Handbook of Applied Cryptography, CRC Press, 1996.
[15] Stinson, D.R., "Some baby-step giant-step algorithms for the low hamming weight discrete logarithm problem," Mathematics of Computation, Vol.71, pp.379-391, 2002.

이 문 규

1996년 2월 서울대학교 컴퓨터공학과 학사. 1998년 2월 서울대학교 컴퓨터공학과 석사. 2003년 8월 서울대학교 전기컴퓨터공학부 박사. 2003년 8월~현재 한국전자통신연구원 정보보호연구단. 관심분야는 컴퓨터이론, 암호학