Parameterized Splitting Systems for the Discrete Logarithm

Sungwook Kim and Jung Hee Cheon

***Abstract*—Hoffstein and Silverman suggested the use of low Hamming weight product (LHWP) exponents to accelerate group exponentiation while maintaining the security level. With LHWP exponents, the computation costs on $\mathrm{GF}(2^n)$ or Koblitz elliptic curves can be reduced significantly, where the cost of squaring and elliptic curve doubling is much lower than that of multiplication and elliptic curve addition, respectively. In this paper, we present a parameterized splitting system with an additional property, which is a refinement version of the system introduced in PKC'08. We show that it yields an algorithm for the discrete logarithm problem (DLP) with LHWP exponents with lower complexity than that of any previously known algorithms.**

To demonstrate its application, we attack the GPS identification scheme modified by Coron, Lefranc, and Poupard in CHES'05 and the DLP with Hoffstein and Silverman's (2,2,11)-exponent. The time complexity of our key recovery attack against the GPS scheme is $2^{61.82}$, which was expected to be 2^{78}. Hoffstein and Silverman's (2,2,11)-exponent can be recovered with a time complexity of $2^{53.02}$, which is the lowest among the known attacks.

***Index Terms*—Discrete Logarithm Problem with Low Hamming Weight Product (LHWP) Exponents, Parameterized Splitting Systems.**

I. Introduction

Let g be a generator of a finite cyclic group G of order m. Given g and $h = g^x \in G$, the *discrete logarithm problem* (DLP) is to compute $x \in [0, m-1]$, which is denoted by $\log_g h$. The DLP is one of the most important underlying mathematical problems in cryptographic applications. The security of many of the current cryptosystems and cryptographic protocols is based on the hardness of the DLP.

The efficiency of DLP-based cryptosystems primarily relies on the speed at which exponentiation can be performed. One approach to achieve fast exponentiation is to use integers of low Hamming weight (LHW) as secret exponents [1], because the number of multiplications required for an exponentiation depends on the weight of the exponent. However, more efficient attacks on the DLP with LHW exponents have been proposed by Heiman-Odlyzko [10] and then by Coppersmith [3], [4], [15] and so the advantage of LHW exponents becomes insignificant. In fact, the time complexity, which means the number of required group operations, of Coppersmith's algorithm is about the square root of the size of the key space. It can be regarded as almost optimal in the sense that the complexity of the DLP on a group is lower bounded by the square root of the group order in the generic group model [20].

To resist previous attacks and achieve a greater speed-up, Hoffstein and Silverman suggested the use of low Hamming weight product (LHWP) exponents [11]. This was then applied to the GPS identification scheme, recommended by the NESSIE project [9], in which a secret key is taken as a product of two integers having low Hamming weights [8], [5]. In a general manner, this type of the DLP is a form of $h = g^{xy}$, where x is an integer of length n and Hamming weight t, and y is an element of a set Y. The essential part of the attack for this exponent is to split x into the sum of u and v and then apply the meet-in-the-middle technique for $h^{y^{-1}} g^{-u} = g^v$ so that the number of group operations required to compute the left-hand side of the above equation is almost equal to that of right-hand side. However, the splitting of x in Heiman-Odlyzko's or Coppersmith's algorithm has a fixed length n or a fixed weight $t/2$, respectively, and thus does not fit into this situation.

In this paper, we propose a more flexible splitting system, called a parameterized splitting system. It can be regarded as a generalization of Coppersmith's splitting system: given a bit string of length n and weight t and any positive integer $t_1 < t$, there exists a part of the string of length n_1 and weight t_1 where $\frac{n_1}{t_1} \approx \frac{n}{t}$. By exploiting this property, given an n-bit integer x one can find an n_1-bit integer u and an $(n-n_1)$-bit integer v of weight t_1 and $t-t_1$, respectively, such that $u + v2^{n_1} \equiv x2^k \pmod{2^n - 1}$ for some integer k. The concept of a parameterized splitting system was introduced in the preliminary version [12] of this paper. In this paper, we further refine it by adding a new property: when we split x into u and v, we can take an odd u while maintaining other properties, which reduces the attack complexity further.

We apply a parameterized splitting system to the private key of the GPS identification scheme in [5] and [8] and to Hoffstein and Silverman's (2,2,11)-exponent in [11], both of which are originally designated for 80-bit security. In [5], Coron, Lefranc, and Poupard proposed an attack with 2^{52} complexity to recover the private key of the modified GPS identification scheme from CHES'04 and suggested a new private key that they claimed had a security level of 2^{78}. But our parameterized splitting system reduces them to $2^{45.57}$ and $2^{64.53}$, respectively, and its randomized version reduces them to $2^{44.57}$ and $2^{61.82}$, respectively. In [2], Cheon and Kim introduced the notion of rotation-free elements and proposed an attack with $2^{55.9}$ group exponentiations to Hoffstein and

The work of S. Kim and J. H. Cheon was supported by the NRF grant by the Korea government (MEST) (No. R01-2008-000-11287-0, No. 2009-0058574, No. 2009-0063183). The work of S. Kim is also partially supported by the Seoul Scholarship Foundation by Seoul city. The material in this paper was presented in part at the PKC'08, Barcelona, Spain, March 2008.

S. Kim and J. H. Cheon are with ISaC & Department of Mathematical Sciences, Seoul National University (SNU), 599 Gwanangno, Gwanak-gu, Seoul 151-747, Korea (e-mail: {avell7, jhcheon}@snu.ac.kr).

Silverman's (2,2,11)-exponent. We reduce it further to $2^{53.02}$ by combining parameterized splitting systems and the notion of rotation-freeness.

The paper is organized as follows: in Section 2, we briefly introduce Heiman-Odlyzko's algorithm and Coppersmith's splitting system. In Section 3, we propose parameterized splitting systems. In Section 4, we describe the method to solve the DLP with LHWP exponents using parameterized splitting systems. In Section 5, we analyze the security of the GPS identification scheme and Hoffstein and Silverman's (2,2,11)-exponent. Finally, we conclude in Section 6.

Notation: Throughout this paper we use the following notation. Let g be a generator of a finite cyclic group G of order m. We represent $\mathbb{Z}_m$ as a set $\{0, 1, 2, \ldots, m-1\}$. Then $n = \lceil \log_2 m \rceil$ bits are required to represent an element of $\mathbb{Z}_m$ as a binary string. For an integer x, $x \bmod n$ is used to denote the remainder of x when divided by n. We denote the Hamming weight of x by $wt(x)$, which is defined as the number of nonzero coefficients in its binary representation. Given a and b with $0 \le a, b < n$ and $a \ne b$, we define

$$[a,b)_n = \begin{cases} \{a, a+1, \ldots, b-1\} & \text{if } a < b, \\ [a,n)_n \cup [0,b)_n & \text{if } b < a. \end{cases}$$

We call a the starting element of the interval $[a,b)_n$.

II. DLP with LHW Exponents and Splitting Systems

Using LHW exponents is one approach to accelerate group exponentiations. For an exponent x of Hamming weight t over a group $\mathrm{GF}(2^n)$, only $t-1$ multiplications are required for exponentiation if a group element is represented with respect to a normal basis [1]. On Koblitz elliptic curves the same operations are required since the computation cost of doubling is almost free. But the use of LHW exponents may weaken the security of the scheme. Heiman-Odlyzko's algorithm [10] and Coppersmiths algorithm [3], [4], [15] were proposed to analyze the security of LHW exponents. In this section we briefly describe the two algorithms.

A. Heiman-Odlyzko's Algorithm

Given an integer x of weight t, and a non-negative integer $t_1 < t$ we want to express x as the sum of two integers x_1 and x_2, with weights t_1 and $t - t_1$, respectively. Such x_1 is easily obtained by choosing t_1 positions among the nonzero coefficients of the binary representation of x. Then we have

$$h = g^x = g^{x_1 + x_2},$$

and so

$$hg^{-x_1} = g^{x_2}. \tag{1}$$

Heiman-Odlyzko's algorithm works as follows: first we compute hg^{-x_1} for each $x_1 \in \mathbb{Z}_m$ of weight t_s, build a lookup table that contains all the pairs (hg^{-x_1}, x_1), and support an efficient search on the first component. Then we compute g^{x_2} for each $x_2 \in \mathbb{Z}_m$ of weight $t - t_s$ and use the lookup table to find a collision.

Note that the exponentiations can be performed incrementally so that each requires only a constant number of group operations. Neglecting logarithmic factors required to sort the table, the time complexity of Heiman-Odlyzko's algorithm is $O\left(\binom{n}{t_s} + \binom{n}{t-t_s}\right)$ group operations in G. Since we need to store only either the left- or right-hand side, the space complexity of Heiman-Odlyzko's algorithm is $O\left(\min\{\binom{n}{t_s}, \binom{n}{t-t_s}\}\right)$.

B. Coppersmith's Algorithm

Coppersmith's algorithm comes from the following observation called *Coppersmith's Splitting System*. For the proof, refer to [22].

Theorem 1 (Coppersmith's Splitting System): Suppose n and t are both even integers. Let $I = [0,n)_n$ and $\mathcal{B} = \{B_i : 0 \le i \le \frac{n}{2} - 1\}$, where $B_i = [i, i + \frac{n}{2})_n$ is an interval called a block. Then for every $T \subseteq I$ such that $|T| = t$, there exists a block $B \in \mathcal{B}$ such that $|T \cap B| = \frac{t}{2}$.

This system can be extended to odd integers n and t [22], where $\frac{n}{2}$ and $\frac{t}{2}$ are replaced by the nearest integers to $\frac{n}{2}$ and $\frac{t}{2}$, respectively.

Coppersmith's algorithm works as follows: given a binary representation $\sum_{i=0}^{n-1} x_i 2^i$ of $x \in \mathbb{Z}_m$, we define

$$\begin{aligned} u_k &:= \sum_{j=0}^{\frac{n}{2}-1} x_{k+j \bmod n} 2^{k+j \bmod n}, \\ v_k &:= x - u_k \end{aligned}$$

for $k = 0, \ldots, \frac{n}{2} - 1$. By Theorem 1, there exists i such that

$$x = \sum_{j=0}^{n-1} x_j 2^j = u_i + v_i,$$

where $wt(u_i) = wt(v_i) = \frac{t}{2}$. As in Eq. (1), we can compute x using

$$hg^{-u_i} = g^{v_i}.$$

This algorithm has a time complexity of $O\left(n\binom{\frac{n}{2}}{\frac{t}{2}}\right)$ and a space complexity of $O\left(\binom{\frac{n}{2}}{\frac{t}{2}}\right)$.

The randomized version of the above algorithm was invented by Coppersmith [4] and is described in [22]. In this version, a block B consists of randomly chosen $\frac{n}{2}$ elements in $[0,n)_n$. The time and space complexities of the randomized version are $O\left(\sqrt{t}\binom{\frac{n}{2}}{\frac{t}{2}}\right)$ and $O\left(\binom{\frac{n}{2}}{\frac{t}{2}}\right)$, respectively.

III. Parameterized Splitting Systems

In this section we propose *parameterized splitting systems*. A parameterized splitting system is a generalization of Coppersmith's splitting system for further applications. Given $T \subset I$, Coppersmith's splitting system gives $B \in \mathcal{B}$ such that $|T \cap B| = t/2$. Our parameterized splitting system, however, is flexible since it provides T with $|T \cap B| = t_s$ and $|B| = \lfloor \frac{t_s n}{t} \rfloor$ for any $1 \le t_s \le t$, which yields an efficient algorithm for the DLP with LHWP exponents. Furthermore, if we take $\mathcal{B}$ similar to Theorem 1, then it has an additional property that one of these Bs must have the starting element belonging to T.

A. Parameterized Splitting Systems

We start with the definition of parameterized splitting systems.

Definition 1 (Parameterized Splitting Systems): Let n and t be integers such that $0 < t < n$ and $I = [0, n)_n$. For any t_s with $1 \le t_s \le t$, a subset $\mathcal{B}_n$ of $\{B \subset I : |B| = \lfloor \frac{t_s n}{t} \rfloor\}$ with cardinality N is called an $(N; n, t, t_s)$-parameterized splitting system of I if there exists a block $B \in \mathcal{B}$ such that $|T \cap B| = t_s$ for every $T \subseteq I$ with $|T| = t$.

Theorem 2: Let $1 \le t_s \le t < n$ be integers and $n_s = \lfloor \frac{t_s n}{t} \rfloor$. Then $\mathcal{B}_n = \{B_i = [i, i + n_s)_n : 0 \le i \le n - 1\}$ is an $(n; n, t, t_s)$-parameterized splitting system of $I = [0, n)_n$ with additional property: for any $T \subset I$ of cardinality t, there exists a block $B_i \in \mathcal{B}_n$ such that $i \in T$ and $|B_i \cap T| = t_s$.

Proof: Let $T = \{y_0, y_1, \ldots, y_{t-1}\}$. For $0 \le i \le t - 1$, we define

$$I_i := [y_{i \bmod t}, y_{i+1 \bmod t})_n$$

and

$$A_i := I_{i \bmod t} \cup \cdots \cup I_{i+t_s-1 \bmod t}.$$

Then $|T \cap A_i| = t_s$ for all i. Since $I_i = \bigcap_{j=0}^{t_s-1} A_{i-j \bmod t}$,

$$|A_0| + |A_1| + \cdots + |A_{t-1}| = t_s \sum_{i=0}^{t-1} |I_i| = t_s |I| = t_s n.$$

If $|A_i| = n_s$ for some i, then this block $A_i = [y_{i \bmod t}, y_{i+t_s \bmod t})_n$ is the desired one. Now suppose that $|A_i| \ne n_s$ for all i. If $|A_i| < n_s$ for all i, then

$$t_s n = \sum_{i=0}^{t-1} |A_i| < t n_s = t \left\lfloor \frac{t_s n}{t} \right\rfloor \le t_s n,$$

which is a contradiction. If $|A_i| > n_s$ for all i, then

$$t_s n = \sum_{i=0}^{t-1} |A_i| \ge t(n_s + 1) > t_s n,$$

which is a contradiction. Hence there exists i such that $|A_i| < n_s$ and $|A_{i+1 \bmod t}| > n_s$, which implies

$$|[y_{i+1 \bmod t}, y_{i+t_s \bmod t})_n| = |A_i \cap A_{i+1 \bmod t}| < n_s$$

and

$$|(A_i \cap A_{i+1 \bmod t}) \cup [y_{i+t_s \bmod t}, y_{i+t_s+1 \bmod t})_n| = |A_{i+1 \bmod t}| > n_s.$$

Then there exists $\ell \in [y_{i+t_s \bmod t}, y_{i+t_s+1 \bmod t})_n$ such that $|[y_{i+1 \bmod t}, \ell)_n| = n_s$. This block $[y_{i+1 \bmod t}, \ell)_n$ is what we want to find because

$$\begin{aligned} T \cap [y_{i+1 \bmod t}, \ell)_n &= T \cap A_{i+1 \bmod t} \\ &= \{y_{i+1 \bmod t}, \ldots, y_{i+t_s \bmod t}\} \end{aligned}$$

whose cardinality is equal to t_s. ∎

The above $(n; n, t, t_s)$-parameterized splitting system guarantees that for any given target string x of length n and weight t, by trying at most n blocks of n_s consecutive elements, we can split x into the sum of two strings, one of which is of length n_s and weight t_s, and starts from one of the fixed t positions.

B. A Randomized Version

We may consider a faster algorithm by using probabilistic approaches. Given n, t, n_s and t_s, we randomly choose $B \subset I$ such that $|B| = n_s$ and check whether $|T \cap B| = t_s$. The total number of blocks B such that $|B| = n_s$ and $|T \cap B| = t_s$ is $\binom{t}{t_s}\binom{n-t}{n_s-t_s}$. Hence given t_s, the probability of success is

$$p = \frac{\binom{t}{t_s}\binom{n-t}{n_s-t_s}}{\binom{n}{n_s}}.$$

We note that p is also equal to

$$\frac{\binom{n_s}{t_s}\binom{n-n_s}{t-t_s}}{\binom{n}{t}}.$$

In order to calculate the lower bound of p, we need Lemma 1 from [14].

Lemma 1: Suppose that n and λn are positive integers, where $0 < \lambda < 1$. Define

$$H(\lambda) = -\lambda \log_2 \lambda - (1 - \lambda) \log_2 (1 - \lambda).$$

Then

$$\frac{2^{nH(\lambda)}}{\sqrt{8n\lambda(1-\lambda)}} \le \binom{n}{\lambda n} \le \frac{2^{nH(\lambda)}}{\sqrt{2\pi n\lambda(1-\lambda)}}.$$

The lower bound of p can be easily obtained using Lemma 1 if $t | t_s n$ including the case that n is even and $t_s = t/2$ [22]. But if $t \nmid t_s n$, a more elaborate proof is required.

Theorem 3: [1] Suppose $2 \le t \le n/2$ and $1 \le t_s \le t/2$. Then

$$p > \frac{1}{4}\sqrt{\frac{n\pi}{et(n-t)}} > \frac{\sqrt{\pi}}{4\sqrt{et}}.$$

Furthermore, if $t \mid t_s n$, then

$$p > \frac{1}{4}\sqrt{\frac{n\pi}{t(n-t)}} > \frac{\sqrt{\pi}}{4\sqrt{t}}.$$

Proof: Let $\lambda_1 = \frac{t_s}{t}$, $\lambda_2 = \frac{n_s - t_s}{n-t}$ and $\lambda = \frac{n_s}{n}$. Then we can write

$$p = \frac{\binom{t}{\lambda_1 t}\binom{n-t}{\lambda_2(n-t)}}{\binom{n}{\lambda n}}.$$

From Lemma 1,

$$p \ge \frac{2^{tH(\lambda_1)+(n-t)H(\lambda_2)-nH(\lambda)} \cdot \sqrt{\lambda(1-\lambda)} \cdot \sqrt{2\pi n}}{\sqrt{\lambda_1(1-\lambda_1)\lambda_2(1-\lambda_2)} \cdot 8\sqrt{t(n-t)}}.$$

If $t \mid t_s n$, then $n_s = t_s n/t$ and so $\lambda_1 = \lambda_2 = \lambda$. Hence, $tH(\lambda_1) + (n-t)H(\lambda_2) - nH(\lambda) = 0$. If $t \nmid t_s n$, then $\lambda_2 < \lambda < \lambda_1 \le \frac{1}{2}$. Since H is a continuous function, by the mean value theorem there exist $\lambda < c_1 < \lambda_1$ and $\lambda_2 < c_2 < \lambda$ such that $H(\lambda_1) - H(\lambda) = H'(c_1)(\lambda_1 - \lambda)$ and $H(\lambda_2) - H(\lambda) = H'(c_2)(\lambda_2 - \lambda)$. Hence we have

$$\begin{aligned} & tH(\lambda_1) + (n-t)H(\lambda_2) - nH(\lambda) \\ = \; & t(H(\lambda_1) - H(\lambda)) + (n-t)(H(\lambda_2) - H(\lambda)) \\ = \; & tH'(c_1)(\lambda_1 - \lambda) + (n-t)H'(c_2)(\lambda_2 - \lambda) \\ = \; & \frac{t_s n - t n_s}{n}(H'(c_1) - H'(c_2)). \end{aligned}$$

[1] We previously suggested a lower bound for p in Lemma 3 in [12]. But it turned out that the proof has a flaw. We have corrected it.

Again by the mean value theorem, there exists $c_2 < c < c_1$ such that $H'(c_1) - H'(c_2) = H''(c)(c_1 - c_2)$ since H' is also continuous. From the inequality $t_s n - t n_s \le t - 1$, we have

$$\begin{aligned} & \frac{t_s n - t n_s}{n}(H'(c_1) - H'(c_2)) \\ = & \frac{t_s n - t n_s}{n} H''(c)(c_1 - c_2) \\ \ge & \frac{t-1}{n} \cdot \frac{-\log_2 e}{\lambda_2(1-\lambda_2)} \cdot \frac{t-1}{(n-t)t} \\ > & -\log_2 \sqrt{e}, \end{aligned}$$

where the first inequality holds since $H''(x) = \frac{-\log_2 e}{x(1-x)}$ is increasing for $0 < x < 1/2$ and $c_1 - c_2 < \lambda_1 - \lambda_2$, and the second is obtained by using $1/(n-t) \le \lambda_2$ and $(t-1)/n < 1/2$.

Since

$$\lambda_1(1-\lambda_2) \le \lambda_1 = \frac{t_s}{t} \le \frac{1}{2},$$

$$\begin{aligned} \frac{\sqrt{\lambda(1-\lambda)}}{\sqrt{\lambda_1(1-\lambda_1)\lambda_2(1-\lambda_2)}} & \ge \frac{\sqrt{\lambda_2(1-\lambda_1)}}{\sqrt{\lambda_1(1-\lambda_1)\lambda_2(1-\lambda_2)}} \\ & = \frac{1}{\sqrt{\lambda_1(1-\lambda_2)}} > \sqrt{2}. \end{aligned}$$

Therefore

$$p > 2^{-\log_2 \sqrt{e}} \cdot \sqrt{2} \cdot \frac{\sqrt{2\pi n}}{8\sqrt{t(n-t)}} > \frac{\sqrt{\pi}}{4\sqrt{et}}.$$

■

Theorem 3 shows that the expected value of trials to find an appropriate block B such that $|T \cap B| = t_s$ is $O(\sqrt{t})$, regardless of n and t_s.

IV. Applications to the DLP with LHWP Exponents

In this section we describe how parameterized splitting systems can be used to solve the DLP with LHWP exponents. We further extend our method to the case where the order of a generator g is unknown.

A. The DLP with LHWP Exponents When the Order of g is Known

Let G be a cyclic group of order m generated by g. Given $h \in G$, the DLP is to find z such that $h = g^z$. We consider this problem when z is a product of two elements $x \in X$ and $y \in Y$ for two subsets X and Y of $\mathbb{Z}_m$.

If we apply the meet-in-the-middle technique for the equation $h^{y^{-1}} = g^x$, x and y can be computed in $O(|X| + |Y|)$. This might not be the best approach when $|X|$ is greater than $|Y|$. In this unbalanced case, it might be better to split x as $u + v$ for $u \in U$ and $v \in V$ where U and V are subsets of $\mathbb{Z}_m$ satisfying $X \subset U + V := \{u + v | u \in U, v \in V\}$. Then we check the following equality for each $y \in Y$ as in [5]:

$$h\,(g^y)^{-u} = (g^y)^v\,.$$

Then the complexity becomes $O(|Y|(|U| + |V|))$. When X is a set of LHW elements, the usable splitting systems include those of Heiman-Odlyzko [10] and Coppersmith [4]: the latter has a lower complexity.

To lower the complexity, we may consider the following the equation, as suggested in [11],

$$h^{y^{-1}} g^{-u} = g^v. \tag{2}$$

The meet-in-the-middle attack using the above equation has the complexity $O(|Y||U| + |V|)$, which is smaller than the previous when $|U| < |V|$. For the above X consisting of LHW elements, it is obtained by Heiman-Odlyzko's algorithm, but not by Coppersmith's algorithm, which supports only symmetric splitting with $|U| \approx |V|$.

Let us consider the subset X of $\mathbb{Z}_m$

$$X = \{x = \sum_{j=0}^{n-1} x_j 2^j : x_j = 0 \text{ or } 1, wt(x) = t\}.$$

We explain how to apply our parameterized splitting systems of Theorem 2 in more detail. Define

$$T = \{j : x_j = 1\} \subset I = [0, n)_n.$$

Given $t_s \in [0, \lceil \frac{t}{2} \rceil]$, there exists an $(n; n, t, t_s)$-parameterized splitting system $(I, \mathcal{B})$ by Theorem 2. Hence there is a block

$$B_i = [i, i + n_s \bmod n)_n \in \mathcal{B}$$

such that $|T \cap B_i| = t_s$. For this i, we set

$$u = \sum_{j=0}^{n_s - 1} x_{i+j \bmod n} 2^{i+j \bmod n}.$$

Then we have $wt(u) = t_s$ and $wt(v) = t - t_s$ for $v := x - u$. Furthermore we can force the first nonzero bit of u to be x_i.

The algorithm works as follows: for each i with $0 \le i \le n-1$, we define

$$U_i := \{u = \sum_{j=i}^{n_s+i-1} u_{j \bmod n} 2^{j \bmod n} : u_i = 1, wt(u) = t_s\}$$

and

$$\begin{aligned} V_i := \{v = \sum_{j=0}^{n-1} v_j 2^j : v_i = v_{i+1 \bmod n} = \dots \\ = v_{n_s+i-1 \bmod n} = 0, wt(v) = t - t_s\}. \end{aligned}$$

Then we compute the left-hand side of Eq. (2) for all $u \in U_i$ and $y \in Y$, and store them after sorting by the value $h^{y^{-1}} g^{-u}$. Second, we compute the right-hand side of Eq. (2) for each $v \in V_i$ and check if it is in the list from the first part.

We have to compute $|Y|\binom{n_s-1}{t_s-1}$ exponentiations in the first step, $\binom{n-n_s}{t-t_s}$ exponentiations in the second step, and repeat these two steps n times. Hence the time complexity is

$$O\left(n\left(|Y|\binom{n_s-1}{t_s-1} + \binom{n-n_s}{t-t_s}\right)\right).$$

Since we can store the smaller set among the sets from the first and the second step, the space complexity is

$$O\left(\min\left\{|Y|\binom{n_s-1}{t_s-1}, \binom{n-n_s}{t-t_s}\right\}\right).$$

The randomized version of this algorithm uses randomly chosen blocks that do not need to be sets of consecutive numbers. Theorem 3 guarantees that we can find an appropriate block in at most $\frac{4\sqrt{et}}{\sqrt{\pi}}$ trials. Thus, the running time of the randomized version is

$$O\left(\sqrt{t}\left(|Y|\binom{n_s}{t_s}+\binom{n-n_s}{t-t_s}\right)\right).$$

The space requirement is same as that of the deterministic version.

B. The DLP with LHWP Exponents When the Order of g is Unknown

We consider the DLP of LHWP exponents when the order G is not known. We assume it is known that the order of G is prime. Recall Eq. (2)

$$h^{y^{-1}}g^{-u} \quad = \quad g^{v}.$$

If the order of g is unknown, y^{-1} can not be computed from y and so we cannot use Eq. (2) directly. However, we can overcome this obstacle by the following trick in [5], and, earlier, proposed in [21]: let

$$\Upsilon = \prod_{y\in Y} y,\ \hat{g} = g^{\Upsilon}.$$

Since the order of g is prime, for any nonzero x the order of g^x is equal to that of g. Hence $\hat{g}$ is also a generator of G. From

$$(h^{y^{-1}}g^{-u})^{\Upsilon} = (g^{v})^{\Upsilon},$$

we have

$$h^{\prod_{y'\in Y-\{y\}} y'}\cdot \hat{g}^{-u} \quad = \quad \hat{g}^{v}. \tag{3}$$

Once we precompute and store $\hat{g}$, $\hat{g}^{-1}$, and $h^{\prod_{y'\in Y-\{y\}} y'}$, we can solve the DLP using parameterized splitting systems and a technique similar to that in the known-order case. According to the algorithm proposed in [5], $\{h^{\prod_{y'\in Y-\{y\}} y'} : y \in Y\}$ can be computed in $|Y|\log_2|Y|$ group exponentiations. Thus the total time and memory complexities for solving the DLP increase by $|Y|\log_2|Y|$ both in the deterministic or randomized versions. However, this increment is almost negligible because $\log_2|Y| \le n$ when $|X| \ge |Y|$.

V. Cryptanalysis

In this section, we apply our algorithms to the GPS identification scheme, introduced by M. Girault in [7] and proved secure by G. Poupard and J. Stern in [18], and the cryptosystem with LHWP exponents proposed by Hoffstein and Silverman [11].

A. The GPS Identification Scheme

The GPS identification scheme, the only identification scheme in the recommended portfolio of the NESSIE project [9], is an interactive protocol between a prover and a verifier which contains one or several rounds of three passes [8]. Let N be a product of two primes that is hard to factorize. The

Prover		**Verifier**
choose $r \in [0,\ 2^R)$		
compute $W = g^r \bmod N$	$\xrightarrow{W}$	
		choose $c \in [0,\ 2^k)$
	$\xleftarrow{c}$	
check $c \in [0,\ 2^k)$		
compute $z = r + x \times c$	$\xrightarrow{z}$	check $z \in [0,\ 2^R + 2^{k+S})$
		verify $g^z h^c = W$

Fig 1. The GPS Identification Scheme

GPS identification scheme is based on the DLP over $G = \langle g\rangle$, where g is an element of ${\mathbb{Z}_N}^*$ of maximal order m and the order of g is kept secret. When $h = g^{-x} \bmod N$, the private key of the prover is x and the public keys are $(N,\ g,\ h)$.

The four security parameters of the GPS scheme are as follows [8]:

- S is the binary size of x. Typically, S=160.
- k is the binary size of the challenges sent to the prover and determines the level of security of the scheme.
- R is the binary size of the exponents used in the commitment computation. Typically $R = S + k + 80$.
- e is the number of rounds the scheme is iterated. Theoretically, e is a polynomial in the size of the security parameter. But, in practice, e is often chosen equal to 1.

Assuming the commitment is precomputed, the efficiency of the protocol from the prover side depends on the computation cost of $z = r + x \times c$ carried by the prover in Fig 1. For fast computation of the response, Girault and Lefranc suggested the use of a LHWP secret key [8]; that is, given a S-bit secret key x, we choose ℓ numbers, $x_1, \ldots, x_\ell$, where x_i has bit-length n_i and Hamming weight t_i. Here $S = \sum_{i=1}^{\ell} n_i$. If c is a k-bit number, computing $z = r + x \times c$ requires $S + k + \sum_{i=1}^{\ell} t_i \times (k + \sum_{j=1}^{i-1} n_j)$ bit additions.

As a concrete example, in [8], a private key x was proposed to be $x = x_1x_2$, where x_1 is a 19-bit number with 5 random bits equal to 1, chosen from among the 16 least significant ones and x_2 is a 142-bit number with 16 random bits equal to 1, chosen from among the 138 least significant ones. With this private key, the prover should perform 1168 bit additions for computing $z = r + x \times c$. Later, in order to strengthen the security, x_1 and x_2 were proposed to be a 30-bit number with 12 nonzero bits and a 130-bit number with 26 nonzero bits, respectively [5]. With this private key, the prover should perform 2188 bit additions.

Now we attack these parameters. We set $|X_1| = \binom{16}{5},\ n_2 = 138,\ t_2 = 16$ for the private keys from [8] and $|X_1| = \binom{30}{12},\ n_2 = 130,\ t_2 = 26$ for the private keys from [5]. Since N is public, we can easily compute $\hat{g}^{-1}$ of Eq. (3) using the extended Euclidean algorithm. We note that t_s is chosen to minimize the time complexity.

Tables 1 and 2 compares the complexities of the processes of recovering the private keys for the scheme suggested in [8] and [5], respectively. For the private key suggested [8], Coron *et al.* [5] presented an attack requiring 2^{52} group exponentiations. But the parameterized splitting system and its randomized

version reduce this further to $2^{45.57}$ and $2^{44.57}$, respectively.

Method	Exponentiations	Storage
[8]	2^{52}	2^{33}
Deterministic, $t_s = 7$	$2^{45.57}$	$2^{37.41}$
Probabilistic, $t_s = 7$	$2^{44.57}$	$2^{37.41}$

Table 1. Private Keys from [8]

Table 2 shows that a parameterized splitting system and its randomized version reduce the complexity of the DLP with the private key proposed in [5] from $2^{77.3}$ to $2^{64.53}$ and $2^{61.82}$, respectively. We can use the better bound of p in Theorem 3 because $t_2 \mid n_2$.

Method	Exponentiations	Storage
[5]	$2^{77.3}$	$2^{43.9}$
Deterministic, $t_s = 10$	$2^{64.53}$	$2^{54.58}$
Probabilistic, $t_s = 9$	$2^{61.82}$	$2^{56.09}$

Table 2. Private Keys from [5]

We note that the private keys with $n_1 + n_2 = 160$ and $t_1 + t_2 \leq 44$ can be revealed in 2^{70} group exponentiations. Under these condition the strongest private key, whose security level is $2^{69.92}$, is obtained when $n_1 = 3$, $t_1 = 1$, $n_2 = 157$ and $t_2 = 43$. And the private keys with $t_1 + t_2 \leq 52$ can be revealed in 2^{75} group exponentiations. Under these condition the strongest private key is obtained when $n_1 = 3$, $t_1 = 1$, $n_2 = 157$ and $t_2 = 51$. This private key achieves a security level of $2^{74.94}$. We get the above results by applying a randomized version to all keys under a given condition.

B. Hoffstein and Silverman's Exponents

Hoffstein and Silverman proposed the use of exponent $x = x_1x_2x_3 \in \mathbb{Z}_{2^{1000}-1}$, where x_1, x_2 and x_3 are integers of $wt(x_1) = 6$, $wt(x_2) = 7$ and $wt(x_3) = 7$, called a (6,7,7)-exponent, or $wt(x_1) = 2$, $wt(x_2) = 2$ and $wt(x_3) = 11$ [11], called a (2,2,11)-exponent. When ignoring squaring, which is much faster than a multiplication in binary fields, the computation of g^x requires 5+6+6=17 multiplications for a (6,7,7)-exponent and 1+1+10=12 multiplications for a (2,2,11)-exponent. For a (6,7,7)-exponent, all values of the Hamming weights are similar. Hence, splitting one of x_i does not afford an advantage. Therefore, we focus on a (2,2,11)-exponent.

Before attacking this exponent we introduce the concept of *rotation-free* elements [2]. An equivalent relation $\sim$ on $\mathbb{Z}_{2^n-1}$ is defined as follows: $a \sim b$ if and only if there exists a non-negative integer i such that $a = 2^i b$.

The idea behind Cheon and Kim's attack on LHWPs is to reduce the key search space by considering only one element from each equivalent class. However since there is no known algorithm to generate such representatives efficiently, they suggested the use of a set of rotation-free elements that contains at least one representative for each equivalent class. The set is only slightly larger than the number of equivalent classes and is easily generated.

Definition 2: (Rotation-Free Elements [2]) An element $z \in \mathbb{Z}_{2^n-1}$ is called a rotation-free element if there is a t-tuple $(a_1, a_2, \ldots, a_t)$ for a positive integer t satisfying

Method	Exponentiations	Storage
[2]	$2^{55.9}$	$2^{54.5}$
Ours , $t_s = 4$	$2^{53.02}$	$2^{49.80}$

Table 3. The Hoffstein and Silverman's (2,2,11)-Exponent

1) $a_i \geq a_1$ for $1 \leq i \leq t$;
2) $\sum_{i=1}^{t} a_i = n$;
3) $z = 2^{n-1} + 2^{n-1-a_1} + \cdots + 2^{n-1-(a_1+a_2+\cdots+a_{t-1})}$.

Let n, t be positive integers with $t < n$ and $\mathrm{RF}(n,t)$ be the number of rotation-free elements of weight t in $\mathbb{Z}_{2^n-1}$. Then $\mathrm{RF}(n,t)$ is given in [2] by

$$\mathrm{RF}(n,t) = \sum_{i=0}^{\lfloor \frac{n}{t} \rfloor - 1} \binom{n-2-ti}{t-2}.$$

Now we propose an improved attack to a (2,2,11)-exponent. According to the trick of [2], we convert the equation $y = g^{x_1x_2x_3}$ to $y^{2^t \bar{x}_1^{-1} \bar{x}_2^{-1}} = g^{x_3}$, where $0 \leq t < n = 1000$ and each of $\bar{x}_1$ and $\bar{x}_2$ is a rotation-free element in $\mathbb{Z}_{2^n-1}$. Then we split x_3 into $x_3 = x_4 + x_5$ using our parameterized splitting system with $wt(x_4) = t_s$ and $wt(x_5) = 11 - t_s$. We then have

$$y^{2^t \bar{x}_1^{-1} \bar{x}_2^{-1}} g^{-x_4} = g^{x_5}. \tag{4}$$

We take more operations to Eq. (4). By repeating squaring both sides of Eq. (4), we may assume that x_4 is just the first n_s bits of $2^{t'} x_3$ for some t'. Then the complexity of the splitting systems is reduced by n. That is, it is sufficient to consider a string of length n_s with t_s weights and starting from 1 for x_4. Therefore the total time complexity for $t_s = 4$ is equal to

$$n \cdot \binom{\mathrm{RF}(n,2)+1}{2} \cdot \binom{n_s - 1}{t_s - 1} + \binom{n - n_s}{t - t_s} \approx 2^{53.02}$$

group exponentiations and the space complexity is equal to $2^{49.80}$. The second term of the left-hand side is obtained from a combination with repetition of $\mathrm{RF}(n,2)$ elements choose 2. It is a deterministic algorithm, but has no less complexity than our randomized algorithm. We summarize the results in Table 3.

C. Implementations

The full implementation of the proposed attacks is not easy due to huge memory requirements. For example, the proposed attack in $\mathrm{GF}(2^{1000})$ for $(2,2,11)$-exponents requires $2^{49.80}$ memory, which amounts to about 2^{16} TBytes. It is too huge to store.

To verify the effectiveness of our attack and estimate the attack time in practice, we may try an implementation of our attacks for modified parameters requiring smaller time and storage complexity. We have chosen (2,2,11) exponents because the change of the size of the base field is enough to reduce the complexity within practical bound. On $\mathrm{GF}(2^{61})$, we take $t_s = 4$ and the lookup table for right-hand side of Eq. (4) consists of about $2^{23.87}$ elements, which requires 0.25 GBytes memory. The number of $y^{2^t \bar{x}_1^{-1} \bar{x}_2^{-1}}$ of Eq. (4) is about $2^{25.17}$. Hence the time complexity is about $2^{23.87} + 2^{25.17} \approx 2^{25.66}$.

The experiment was performed using the NTL [19] on a machine with a dual-core AMD Opteron 2.6 GHz CPU and 4 GBytes RAM. We have tested the attack for 200 number of randomly chosen h. The discrete log of each h was computed in 219.64 seconds on average. More precisely, computing exponentiations for $2^{23.87}$ exponents and constucting the lookup table took 103.6 senconds. And computing on-the-fly and finding a match on the loolup table took 116.04 seconds.

A multiplication in $\mathrm{GF}(2^{1000})$ could be 16^2 times slower than $\mathrm{GF}(2^{61})$ using a schoolbook multiplication method. Using a fast arithmetic, however, a multiplication in $\mathrm{GF}(2^{1000})$ is about 5 times slower than $\mathrm{GF}(2^{61})$ by our experiment. Hence the attack time on $(2, 2, 11)$-exponents in $\mathrm{GF}(2^{1000})$ is estimated to be about $219.64 \cdot 5 \cdot 2^{53.02-25.66} \approx 2^{37.6}$ seconds. We note that the attack on real parameters are possible only with sufficiently large memory allowing efficient read and write.

VI. Conclusion and Future Work

In this paper, we have proposed a parameterized splitting system, which is the refinement of a parameterized splitting system in [12]. The flexibility in the choice of the size of a block allows easier control of the trade-off between time and space complexity for solving the DLP with LHWP exponents. Moreover, the property that such a block starts with one reduces the time complexity further.

In the generic group model, the computational complexity of the DLP constrained to a subset S of a group G is known to be lower-bounded by the square root of the cardinality of S [20], [13]. In [6], Erdös and Newman asked for finding a set that is resistant to the baby-step giant-step algorithm, *i.e.,* the computational complexity of the DLP on S is larger than the square root of the cardinality of S. A set of LHWP exponents is a good candidate for this problem. The attack on LHWP exponents using a parameterized splitting system is the most efficient of any previously known algorithms, but is still larger than the square root bound of the key space. In particular, when the secret exponent is the product of three integers with almost-equal Hamming weights, our algorithm is far from the bound. It still remains open whether a set of LHWP exponents is an answer to the Erdös and Newman question.

So far today, all known efficient algorithms for the DLP with LHWP exponents require the space complexity comparable to the time complexity while the ordinary DLP has space-efficient algorithms such as Pollard rho or kangaroo [16], [17]. It would be an important future research question to find a space-efficient algorithm for this problem.

Acknowledgement: The authors would like to thank the reviewers for their valuable comments. This work was supported by the NRF grant by the Korea government (MEST) (No. R01-2008-000-11287-0, No. 2009-0058574, No. 2009-0063183). The work of S. Kim is also partially supported by the Seoul Scholarship Foundation by Seoul city.

References

[1] G. Agnew, R. Mullin, I. Onyszchuk and S. Vanstone, "An Implementation for a Fast Public-Key Cryptosytem," J. Cryptology, Vol. 3, No. 2, pp. 63–79, 1991.

[2] J. Cheon and H. Kim, "Analysis of Low Hamming Weight Products," Discrete Appl. Math., Vol. 156, No. 12, pp. 2264–2269, Jun. 2008.

[3] D. Coppersmith and G. Seroussi, "On the Minimum Distance of Some Quadratic Residue Codes," IEEE Trans. Inf. Theory, Vol. 30, pp. 407–411, Mar. 1984.

[4] D. Coppersmith, "Private communication to Scott Vanstone," Dec. 1997.

[5] J. Coron, D. Lefranc and G. Poupard, "A New Baby-Step Giant-Step Algorithm and Some Application to Cryptanalysis," in Cryptographic Hardware and Embedded Systems - CHES 2005, Lecture Notes in Computer Science 3656, pp. 47–60, 2005.

[6] P. Erdös and D. Newman, "Bases for Sets of Integers," J. Number Theory, Vol. 9, No. 4, pp. 420–425, 1977.

[7] M. Girault, "Self-Certified Public Keys," in Advances in Cryptology - Eurocrypt 1991, Lecture Notes in Computer Science 547, pp. 490–497, 1991.

[8] M. Girault and D. Lefranc, "Public Key Authentication with One Single (on-line) Addition," in Cryptographic Hardware and Embedded Systems - CHES 2004, Lecture Notes in Computer Science 3156, pp. 413–427, 2004.

[9] M. Girault, G. Poupard and D. Lefranc, "Some Modes of Use of the GPS Identification Scheme," in Third NESSIE Workshop, Springer-Verlag, Nov. 2002.

[10] R. Heiman, "A Note on Discrete Logarithms with Special Structure," in Advances in Cryptology - Eurocrypt 1992, Lecture Notes in Computer Science 658, pp. 454–457, 1992.

[11] J. Hoffstein and J. Silverman, "Random Small Hamming Weight Products with Application to Cryptography," Discrete Appl. Math., Vol. 130, No.1, pp. 37–49, 2003

[12] S. Kim and J. Cheon, "A Parameterized Splitting System and its Application to the Discrete Logarithm Problem with Low Hamming Weight Product Exponents," in Public Key Cryptography - PKC 2008, Lecture Notes in Computer Science 4939, pp. 328–343, 2008.

[13] I. Mironov, A. Mityagin and K. Nissim, "Hard Instances of the Constrained Discrete Logarithm Problem," in Algorithm Number Theory Symposium - ANTS 2006, Lecture Notes in Computer Science 4076, pp. 582–598, 2008.

[14] F. J. MacWilliams and N. J. A. Sloane, "The Theory of Error-Correcting Codes," North-Holland, MR 57:5408a; MR 57:5408b, 1977, pp. 309.

[15] A. Menezes, P. van Oorschot and S. Vanstone, "Handbook of Applied Cryptography," CRC Press, 1997, pp. 128.

[16] J. Pollard, "Monte Carlo Methods for Index Computation $(\mathrm{mod}\, p)$," Mathematics of Computation, Vol. 32, pp. 918-924, 1978.

[17] J. Pollard, "Kangaroos, Monopoly and Discrete Logarithms," Journal of Cryptology, Vol 13, No. 4, pp. 437-447, 2000.

[18] G. Poupard and J. Stern, "Security Analysis of a Practical "on the fly" Authentication and Signature Generation," in Advances in Cryptology - Eurocrypt 1998, Lecture Notes in Computer Science 1403, pp. 422–436, 1998.

[19] V. Shoup. (2009, Nov.). NTL: A Library for doing Number Theory (Ver 5.4.1) [Online]. Available: http://www.shoup.net/ntl/

[20] V. Shoup, "Lower Bounds for Discrete Logarithms and Related Problems," in Advances in Cryptology - Eurocrypt 1997, Lecture Notes in Computer Science 1233, pp. 256–266, 1997.

[21] V. Shoup, "Practical Threshold Signatures," in Advances in Cryptology - Eurocrypt 2000, Lecture Notes in Computer Science 1807, pp. 207–220, 2000.

[22] D. Stinson, "Some Baby-Step Giant-Step Algorithms for the Low Hamming Weight Discrete Logarithm Problem," Mathematics of Computation, Vol. 71, No. 237, pp. 379–391, 2002.

Sungwook Kim Sungwook Kim is a graduate student in the department of mathematical sciences of Seoul National University (SNU). He had a B.S. degree in mathematics from SNU in 2005. His research interests include computational number theory, cryptography and information security.

Jung Hee Cheon Jung Hee Cheon is an associate professor in the department of mathematical sciences of Seoul National University (SNU). He received his B.S. and Ph.D. degrees in mathematics from KAIST in 1991, and 1997, respectively. From 1997, he worked for Electronics and Telecommunications Research Institute (ETRI) and then Information and Communications University (ICU). In 2000 he was a visiting scientist in Brown university. His research interests include computational number theory, cryptography and information security.

He is an associate editor of Journal of KIISC and CSI journal and co-chaired ICISC 2008. He served as program committee members for many conferences including Eurocrypt, Asiacrypt, PKC, and ICISC. He received the best paper award in Asiacrypt 2008.