Int. J. Communications, Network and System Sciences, **2010, 3, 639-644**
doi:10.4236/ijcns.2010.38086 Published Online August 2010 (http://www.SciRP.org/journal/ijcns)

Potential Vulnerability of Encrypted Messages: Decomposability of Discrete Logarithm Problems

Boris S. Verkhovsky
Computer Science Department, New Jersey Institute of Technology, Newark, USA
E-mail: *verb@njit.edu*
Received May 10, 2010; *revised June* 15, 2010; *accepted July* 21, 2010

Abstract

This paper provides a framework that reduces the computational complexity of the discrete logarithm problem. The paper describes how to decompose the initial DLP onto several DLPs of smaller dimensions. Decomposability of the DLP is an indicator of potential vulnerability of encrypted messages transmitted via open channels of the Internet or within corporate networks. Several numerical examples illustrate the framework and show its computational efficiency.

Keywords: Network Vulnerability, System Security, Discrete Logarithm, Integer Factorization, Multi-Level Decomposition, Complexity Analysis

1. Introduction and Problem Statement

The cryptoimmunity of numerous public key cryptographic protocols is based on the computational complexity of the discrete logarithm problems [1,2].

A DLP finds an integer *x* satisfying the equation

$$g^x \bmod p = h. \quad (1)$$

Here $2 \le g \le p-1;\ 1 \le h \le p-1$ (2)

and *p* is a large prime. In (1) *g*, *p* and *h* are inputs, and the unknown integer *x* must be selected on the interval $[1, p-1]$.

Two trivial cases: if *h* = 1, then *x* = *p* – 1; If *h* = *g*, then *x* = 1. If *h* is neither 1 nor *g*, then x must be selected on the interval [2, *p* – 2].

If *g* is a generator, then (1) always has a solution, otherwise the existence of a solution is not guaranteed.

For instance, if *p* = 7 and *g* = 2, then the DLP $2^x \bmod 7 = 5$ does not have a solution.

Various algorithms for solving the DLP were proposed and their computational complexities were analyzed over the last forty years [3-15].

This paper provides the algorithmic framework that reduces the computational complexity of the DLP.

The paper describes step-by-step procedure for decomposition of the initial DLP onto several DLPs with smaller dimensions. Several examples illustrate the decomposition algorithm and highlight its computational efficiency.

Let $g_1 := g;\ h_1 := h;\ x_1 := x;$

$$q_1 := p-1 \text{ and } p-1 = 2r_1r_2. \quad (3)$$

Here it is assumed that integer factors r_1 and r_2 in (3) are known or can be determined using existing algorithms for integer factorization [5,16,17].

Proposition: Let $R_1 := (p-1)/q;$ (4)

if $q \mid (p-1)$, then R_1 is an integer (4).

Let's define $g_2 := g_1^{R_1} \bmod p;$ (5)

$$h_2 := h_1^{R_1} \bmod p; \quad (6)$$

If an integer x_2 is a solution of equation

$$g_2^{x_2} \bmod p = h_2, \text{ where } x_2 \in [0, q], \quad (7)$$

then *q* divides $x_1 - x_2$.

Proof: Let's multiply both sides of the Equation (1) by $g_1^{-x_2} \bmod p$ [18], and find x_2, such that

$$h_1 g_1^{-x_2} \bmod p \quad (8)$$

has a root of power q.

By Euler's criterion [5] such a root exists if and only if

$$\left(h_1 g_1^{-x_2}\right)^{(p-1)/q} \bmod p = 1 \quad (9)$$

Using notations (4)-(6), rewrite (8) as

$$h_2 g_2^{-x_2} \bmod p = 1 \quad (10)$$

or as Equation (7). Q.E.D.

Therefore, the unknown x_1 can be represented as

$$x_1 = x_2 + qx_3 \quad (11)$$

where the integer x_3 must be on the interval

$$x_3 \in [0, (p-1)/q] = [0, q_3] \quad (12)$$

After x_2 is determined, we need to find an integer x_3, for which the following equation holds

$$g_1^{x_2+qx_3} \bmod p = h_1. \quad (13)$$

This equation can be rewritten as

$$\left(g_1^{q}\right)^{x_3} = h_1 g_1^{-x_2} \pmod p \quad (14)$$

where in contrast with the BSGS algorithm, the value of x_2 is already known.

Let $$g_3 := g_1^{(p-1)/q_3} \bmod p\,; \quad (15)$$

and $$h_3 := h_1 g_1^{-x_2} \bmod p\,. \quad (16)$$

2. Divide-and-Conquer Decomposition: Illustrative Example-1

Let's solve $$2^{x_1} \bmod 947 = 273, \quad (17)$$

i.e., here $g_1 = 2; p = 947; h_1 = 273$, and $x_1 \in [1, 946]$. Let $q_1 := p-1$.

Since $q_1 = 2r_1r_2 = 2 \times 11 \times 43$, select

$q_2 = \min_{0 \le z \le \sqrt{p-1}} \max\left(z, (p-1)/z\right) = 43$.

Then $R_1 := q_1 / q_2 = 22$; $g_2 := g_1^{R_1} \bmod p = 2^{22} \bmod 947 = 41$; and $h_2 := h_1^{R_1} \bmod p = 273^{22} \bmod 947 = 283$.

Therefore we need to solve the *DLP*(2):

$$41^{x_2} \bmod 947 = 283 \ \ (7), \quad (18)$$

where $x_2 \in [1, 42]$.

***Remark*1**: Notice that the interval of uncertainty [1, 42] for x_2 is much smaller than the corresponding interval of uncertainty [1, 946] for x_1.

Equation (18) can be solved using any algorithm for the DLP [3,6,8-10,12].

In this example $x_2 = 39$ and $q_2 = 43$.

Therefore $x_1 = 39 + 43x_3$, where

$$x_3 \in \left[0, \left(p-1/q_2\right)\right] = [0, 22].$$

To find x_3 solve the DLP(3):

$$\left(2^{43}\right)^{x_3} = 273 \times 2^{-39} \pmod{947},$$

which is equivalent to

$$367^{x_3} = 273 \times 111 = 946 \pmod{947}. \quad (19)$$

Therefore $x_3 = 11$.

Verification: $367^{11} \bmod 947 = 946$. (20)

Finally, $x_1 = 39 + 43 \times 11 = 512$.

3. Multi-Level Decomposition: Illustrative Example-2

***Initial DLP*(1)**: Find an integer x_1, such that

$$30^{x_1} \bmod 99991 = 45636, \quad (21)$$

where $x_1 \in [1, 99990]$.

Because 99990=303*330, select $q_2 = 330$ and represent the unknown x_1 as $x_1 = x_2 + 330x_3$.

Since $R_1 := (p-1)/q_2 = 303$;

then $g_2 := g_1^{303} \bmod 99991 = 151$;

and $h_2 := h_1^{303} \bmod 99991 = 64099$.

***Remark*2**: To better describe the concept of decomposition, a more suitable system of notations is considered below in the following **Table 1**. These notations are used to describe the process of solving three DLPs.

***DLP*(2)**: Solve $g_2^{x_2} \bmod 99991 = h_2$,

i.e., $151^{x_2} \bmod 99991 = 64099$,

where $$x_2 \in [0, 330]. \quad (22)$$

The solution is $x_2 = 115$; indeed

$$151^{115} \bmod 99991 = 64099.$$

Therefore $30^{x_1} = 30^{115+330x_3} \bmod 99991 = 45636$.

Consider the equation

$$\left(30^{330}\right)^{x_3} = 30^{-115} \times 45636 \pmod{99991}.$$

Let $g_3 := 30^{330} \bmod 99991 =$ 2593; and

$$\begin{aligned} h_3 &:= 30^{-115} \times 45636 \\ &= 96658^{115} \times 45636 \pmod{99991} \\ &= 49845 \end{aligned}$$

Therefore, we need to solve

***DLP*(3):** $2593^{x_3} \bmod 99991 = 49845$, where

$$x_3 \in [0, 303]. \quad (23)$$

It is easy to verify that $x_3 = 47$. Finally, $x_1 = x_2 + q_2x_3 = 115 + 330 \times 47 = 15625$.

***Decomposition of DLP*(2):** Solve

$$g_2^{x_2} \bmod p = h_2, \quad (24)$$

where $x_2 \in [0, q_2]$ = [0,330].

Table 1. Solutions of *DLP*(1) via the decomposition of *DLP*(2) and *DLP*(3).

***DLP*(1)**: $g_1^{x_1} \bmod p = h_1$	Problem A	Problem B	Problem C
Inputs $\{g_1; p; h_1\}$	{2; 947; 273}	{2; 947; 641}	{30; 99991; 45636}
$q_1 := p - 1 = 2r_1r_2...r_t$	$2\times 11\times 43$	$2\times 11\times 43$	$2\times 3^2\times 11\times 101$
***DLP*(2)**: $q_2 = \min_z \max(z, q_1/z)$	$q_2 = 43$	$q_2 = 43$	$q_2 = 330$
$R_2 := (p-1)/q_2$	$R_2 = 22$	$R_2 = 22$	$R_2 = 303$
$g_2 := g_1^{R_2} \bmod p$	$g_2 = 41$	$g_2 = 41$	$g_2 = 30^{303} \bmod 99991 = 151$
$h_2 := h_1^{R_2} \bmod p$	$h_2 = 283$	$h_2 = 283$	$h_2 = 45636^{303} \bmod 99991 = 64099$
$g_2^{x_2} \bmod p = h_2$, $x_2 \in [0, q_2]$	$x_2 \in [0,43]$; $x_2 = 39$	$x_2 \in [0,43]$; $x_2 = 23$	$x_2 \in [0,330]$; $x_2 = 115$
***DLP*(3)**: $q_1 = q_2 q_3$, $R_3 := (p-1)/q_3$	$R_3 = 43$	$R_3 = 43$	$R_3 = 330$
$g_3 := g_1^{R_3} \bmod p$	$g_3 = 367$	$g_3 = 367$	$g_3 = 30^{330} \bmod 99991 = 2593$
$h_3 := h_1 g_1^{-x_2} \bmod p$	$h_3 = 946$	$h_3 = 643$	$f = 30^{-1} \bmod p = 96658$, $h_3 = 96658^{x_2} \bmod p = 9381$
$g_3^{x_3} \bmod p = h_3$, $x_3 \in [0, q_3]$	$x_3 \in [0,22]$; $x_3 = 11$	$x_3 \in [0,22]$; $x_3 = 14$	$x_3 \in [0,303]$; $x_3 = 47$
Solution of *DLP*(1): $x_1 = x_2 + q_2 x_3$	$x_1 = 39 + 43 \times 11 = 512$	$x_1 = 23 + 43\times 14 = 625$	$x_1 = 115 + 330 \times 47 = 15625$

***Remark*3**: Notice that the interval of uncertainty in *DLP*(2) is not [1, p – 1], but $x_2 \in [1, q_2]$, which is much smaller than [1, p – 1].

Instead of solving (24) directly using an existing DLP algorithm, we can again apply the method of decomposition described above. Consider a factor q_4 of q_2 that is close to the square root of $q_2 = 330$:

$$q_4 = \min_{0 \le z \le \sqrt{q_2}} \max(z, q_2/z) = \min_z \max(z, 330/z) = 30 \quad (25)$$

Let's represent the unknown in (24) as

$$x_2 = x_4 + q_4 x_5, \quad (26)$$

where $x_4 \in [1, q_4] = [1, 30]$

and $x_5 \in [1, q_5 := q_2 / q_4] = [1, 11]$. (27)

Let us now investigate whether h_2 has an integer root of power 30 modulo p.

By Euler's criterion, such a root exists if and only if

$$h_2^{(p-1)/q_4} \bmod p = 1. \quad (28)$$

However, if $h_2^{(p-1)/q_4} \bmod p \ne 1$, find an integer x_4, which satisfies the equation

$$\left(h_2 g_2^{-x_4}\right)^{(p-1)/q_4} \bmod p = 1. \quad (29)$$

Let $$g_4 := g_2^{(p-1)/q_4} \bmod p; \quad (30)$$

and $$h_4 := h_2^{(p-1)/q_4} \bmod p. \quad (31)$$

Now we need to solve the equation

$$g_4^{x_4} \bmod p = h_4, \quad (32)$$

where $x_4 \in [0, 30]$. And again, the Equation (32) itself is also a DLP with a much smaller interval (27) for x_4 than the interval for x_2 in (24), and so on.

4. Multi-Level Decomposition: Illustrative Example-3

First level: Let's solve the equation $g_1^{x_1} \bmod p = h_1$, where g = 2, p = 4,000,000,003,231; and h = 3,024,336,139,227.

Then p – 1 = 863*2310*2006491, where 863 and 2,006,491 are primes.

In this case the initial *DLP*(1) $g_1^{x_1} \bmod p = h_1$; is decomposable into two sub-problems: *DLP*(2) and *DLP*(3).

***DLP*(2):** Compute

$$g_2 := g_1^{(p-1)/q_2} = 2^{1993530} \bmod 4000000003231 = 3278213345371;$$

and $$h_2 := h_1^{(p-1)/q_2} = 3024336139227^{1993530} \bmod 4000000003231 = 2084778340641.$$

Solve $g_2^{x_2} \bmod 4000000003231 = h_2$, where

$$0 \le x_2 \le q_2 = 2006491;$$

It is easy to verify the solution

$$x_2 = 1853979 \le 2006491.$$

***DLP*(3):**Compute

$$g_3 := g_1^{(p-1)/q_3} = 2^{2006491} \bmod 4000000003231 = 3767306619080;$$

and

$$h_3 := h_1 g_1^{-x_2}$$
$$= 3024336139227 \times 2000000001616^{1853979} \cdot$$
$$\bmod 4000000003231$$
$$= 3024336139227 \times 629308445687 \cdot$$
$$\bmod 4000000003231$$
$$= 2623468766941.$$

Solve $g_3^{x_3} = h_3 \pmod{p}$, where

$$0 \le x_3 = 14622 \le q_3 = (p-1)/q_2 = 1993530;$$

and $q_1 = q_2 q_3$.

Then

$$x_1 = x_2 + q_2 x_3$$
$$=1,853,979 + 2,006,491^* 14,622$$
$$=29,340,765,381.$$

It is easy to verify that the solution

$$x_3 = 14622 \le 1993530.$$

Comparison of complexities: While the size of the required memory/storage for *DLP*(1) equals

$$T_1 = \left\lfloor \sqrt{p-1} \right\rfloor = 2000000;$$

the corresponding memory requirement for *DLP*(2) and *DLP*(3) are respectively

$$T_2 = \left\lfloor \sqrt{q_2 - 1} \right\rfloor = \left\lfloor \sqrt{2006491} \right\rfloor = 1416$$

and $T_3 = \left\lfloor \sqrt{q_3 - 1} \right\rfloor = \left\lfloor \sqrt{1993530} \right\rfloor = 1411$.

Therefore the speed-up ratio

$$S = T_1 / (T_2 + T_3) = 2000000 / (1416 + 1411) = 707.$$

Thus the decomposition algorithm for solving *DLP*(1) via *DLP*(2) and *DLP*(3) is 707 times faster than a direct solution of the original *DLP*(1).

5. Second-Level Decomposition: Solution of *DLP*(3)

Remark4: The second problem, *DLP*(2), cannot be solved by decomposition since q2 = 2,006,491 is a prime integer. However, the third problem, *DLP*(3), is decomposable, therefore the speed-up ratio *S* can be further increased.

Indeed, select $q_6 := \min_{0 \le z \le \sqrt{q_3}} \max(q_3 / z, z) = 2310$.

Let's represent x_3 as $x_3 = x_6 + q_6 x_7$, where

$$0 < x_6 < q_6 = 2310 \quad \text{and} \quad 0 < x_7 < q_7 = 863,$$

and solve *DLP*(3) by decomposition into *DLP*(6) and *DLP*(7).

***DLP*(6)**: Compute $g_6 := g_3^{(p-1)/q_6} \bmod p$;
and $h_6 := h_3^{(p-1)/q_6} \bmod p$;
where $q_6 q_7 = q_3 = 1993530$;
and solve $g_6^{x_6} = h_6 (\bmod 1993531)$;
$\{ 0 < x_6 < q_6 = 2310 \}$.

***DLP*(7)**: Compute $g_7 := g_3^{(p-1)/q_7} \bmod p$;
and $h_7 := h_3 g_3^{-x_6} \bmod p$;
and solve $g_7^{x_7} = h_7 (\bmod 1993531)$;
$\{ 0 < x_7 < q_7 = 863 \}$.

Then $T_6 = \left\lfloor \sqrt{q_6} \right\rfloor = 48$ and $T_7 = \left\lceil \sqrt{q_7} \right\rceil = 29$.

Therefore

$$S = T_1 / (T_2 + T_6 + T_7)$$
$$=2000000/(1416 + 48 + 29)$$
$$=2000000/1493$$
$$=\mathbf{1339.6},$$

which implies that by decomposing the original problem *DLP*(1) into three sub-problems {*DLP*(2), *DLP*(6) and *DLP*(7)}, we can solve the initial *DLP*(1) 1340 times faster than if we directly solve it without employing decomposition.

In general, the speed-up increases as the size of *p* increases.

6. Computational Considerations

It is quite reasonable to ask under what conditions should we stop the decomposition of a *DLP*(*k*) and try to solve it directly. Here are the major issues that must be taken into the consideration:

1) Feasibility of factoring $q_k = q_{2k} q_{2k+1}$ in such a way that

$$g_{2k} := g_k^{(p-1)/q_{2k}} \bmod p \ne \pm 1. \tag{33}$$

For instance, if $q_2 q_4 \mid 2(p-1)$, then

$$\begin{aligned} w_4 := w_2^{(p-1)/q_4} &= \left\lceil w_1^{(p-1)/q_2} \right\rceil^{(p-1)/q_4} \\ &= \left\lceil w_1^{2(p-1)/q_2 q_4} \right\rceil^{(p-1)/2} = \pm 1 (\bmod p) \end{aligned}, \tag{34}$$

where *w* = {*g*, *h*}. In such a case Equation (32) has only trivial solutions {0 or 1} or no solution
if $g_4 = 1$ and $h_4 = -1$.

2) Magnitude of the overhead computations required to find g_{2k} and g_{2k+1} and then to solve these two DLPs, provided that these intermediate computations do not

become too "costly".

Remark 4: Analogously, we can solve *DLP*(3) by decomposing it into two DLPs with smaller intervals of uncertainty for the corresponding unknowns.

7. Algorithmic Decomposition of *DLP*(*k*)

Suppose that we need to solve *DLP*(*k*)

$$g_k^{u_k} \bmod p = h_k, \quad (33)$$

where $u_k \in [0, q_k]$.

If q_k is a prime or if factors of q_k are unknown, then (33) can be solved by an algorithm for DLP such as: BSGS, Pollard's rho-algorithm, Lenstra's number field algorithm etc. However, if $q_k = cd$, where both c and d are integers, then the *DLP*(*k*) can be reduced to solving two less complex DLPs: *DLP*(2*k*) and *DLP*(2*k* + 1).

Let $q_k = q_{2k} q_{2k+1}$;

***DLP*(2*k*)**: Solve $g_{2k}^{u_{2k}} \bmod p = h_{2k}$; (34)

where $q_{2k} := c$ and $u_{2k} \in [0, c]$; (35)

$$R_k := (p-1)/q_k; \quad (36)$$

$$g_{2k} := g_k^{R_k} \bmod p; \quad (37)$$

and $h_{2k} := h_k^{R_k} \bmod p$; (38)

***DLP*(2*k*+1):** Solve

$$g_{2k+1}^{u_{2k+1}} \bmod p = h_{2k+1}; \quad (39)$$

where $u_{2k+1} \in [0, q_k / c]$, (40)

$$R_{2k+1} := (p-1)/q_{2k+1}; \quad (41)$$

$$g_{2k+1} := g_k^{R_{2k+1}} \bmod p; \quad (42)$$

and $h_{2k+1} := h_k g_k^{-u_{2k}} \bmod p$. (43)

8. Conclusions

Provided that we know how to factor $p - 1$, we can reduce the initial *DLP*(1) to two discrete logarithm problems: *DLP*(2) and *DLP*(3), for solution of which the best known algorithms can be implemented. The decomposition can be implemented recursively for solution of the *DLP*(*k*) by reducing it to a pair of *DLP*(2*k*) and *DLP*(2*k* + 1).

9. Acknowledgements

I express my appreciation to R. Rubino and P. Fay for their comments and suggestions that improved the style of this paper.

10. References

[1] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, Vol. 22, No. 6, 1976, pp. 644-654.

[2] T. ElGamal, "A Public Key Cryptosystem and a Digital Signature Scheme Based on Discrete Logarithms," *IEEE Transactions on Information Theory*, Vol. 31, No. 4, 1985, pp. 469-472.

[3] L. M. Adleman and J. DeMarrais, "A Sub-Exponential Algorithm for Discrete Logarithms over all Finite Fields," *Mathematics of Computation*, Vol. 61, No. 203, 1993, pp. 1-15.

[4] E. Bach, "Discrete Logarithms and Factoring," *Technical Report*: *CSD*-84-186, University of California, Berkeley, 1984.

[5] R. Crandall and C. Pomerance, "Prime Numbers: A Computational Perspective," *The Quadratic Sieve Factorization Method*, Springer, Berlin, 2001, pp. 227- 244.

[6] A. Enge and P. Gaudry, "A General Framework for Sub-Exponential Discrete Logarithm Algorithms," *Research Report* LIX/RR/00/04, Luxembourg Internet eXchange (LIX), Luxembourg Kirchberg, Vol. 102, June 2000, pp. 83-103.

[7] B. A. LaMacchia and A. M. Odlyzko, "Computation of Discrete Logarithms in Prime Fields," *Designs*, *Codes and Cryptography*, Vol. 19, No. 1, 1991, pp. 47-62.

[8] A. K. Lenstra and J. H. W. Lenstra, "The Development of the Number Field Sieve," *Lecture Notes in Mathematics*, Springer-Verlag, Berlin, Vol. 1554, 1993, pp. 95-102.

[9] V. Müller, A. Stein and C. Thiel, "Computing Discrete Logarithms in Real Quadratic Congruence Function Fields of Large Genus," *Mathematics of Computation*, Vol. 68, No. 226, 1999, pp. 807-822.

[10] O. Schirokauer, "Using Number Fields to Compute Logarithms in Finite Fields," *Mathematics of Computation*, Vol. 69, No. 231, 2000, pp. 1267-1283.

[11] D. Shanks, "Class Number, a Theory of Factorization and Genera," *Proceedings of Symposium in Pure Mathematics*, Vol. 20, American Mathematical Society, Providence, 1971, pp. 415-440.

[12] J. Silverman, "The *xedni* Calculus and the Elliptic Curve Discrete Logarithm Problem," *Designs*, *Codes and Cryptography*, Vol. 20, No. 1, 2000, pp. 5-40.

[13] D. C. Terr, "A Modification of Shanks' Baby-Step Giant-Step Algorithm," *Mathematics of Computation*, Vol. 69, No. 230, 2000, pp. 767-773.

[14] B. Verkhovsky, "Generalized Baby-Step Giant-Step Algorithm for Discrete Logarithm Problem," *Advances in Decision Technology and Intelligent Information Systems*, International Institute for Advanced Studies in Systems Research and Cybernetics, Baden-Baden, 2008, pp. 88-89.

[15] R. Zuccherato, "The Equivalence between Elliptic Curve and Quadratic Function Field Discrete Logarithms in Characteristic 2," *Algorithmic Number Theory Seminar*

ANTS-III, *Lecture Notes in Computer Science*, Springer, Berlin, Vol. 1423,1998, pp. 621-638.

[16] J. P. Pollard, "A Monte Carlo Method for Factorization," *BIT Numerical Mathematics*, Vol. 15, No. 3, 1975, pp. 331-334.

[17] C. Pomerance, J. W. Smith and R. Tuler, "A Pipeline Architecture for Factoring Large Integers with the Quadratic Sieve Algorithm," *SIAM Journal on Computing*, Vol. 17, No. 2, 1988, pp. 387-403.

[18] B. Verkhovsky, "Multiplicative Inverse Algorithm and its Complexity," *Proceedings of International Conference on System Research*, *Informatics* & *Cybernetics*, Baden-Baden, 28-30 July 1999, pp. 62-67.

APPENDIX

Numeric example as an exercise

Let p = 5,000,491; then $p-1=990\times 5051$ Let

$$g_1 = 2 \text{ and } h_1 = 1020305 .$$

In this case *DLP*(1) is $2^{x_1} = 1020305 \pmod{5000491}$, where the unknown $x_1 \in [1, p-1]$.

The *DLP*(1) is decomposable into two sub-problems:

DLP(2): $g_2^{x_2} = h_2 \pmod{p}$ {see (4)-(6)}, where

$$x_2 \in [1, q_2] = [1, 5051];$$

and *DLP*(3): $g_3^{x_3} = h_3 \pmod{p}$ {see (15) and (16)}, where

$$x_3 \in [1, q_3] = [1, 990].$$

Therefore $x_1 = x_2 + q_2 x_3$.

***Remark*5**: The reader now has an opportunity to solve this problem himself since values required for the decomposition are purposely omitted.

From *DLP*(2) and *DLP*(3) we find that

$$x_2 = 1947 < 5051;$$

and

$$x_3 = 470 < 990 .$$

Finally,

$$x_1 = 1947 + 5051 \times 470 = 2375917.$$

Overall complexity: the storage requirement for *DLP* (2) and *DLP*(3) equal to 71 and 31 respectively, yet the size of required storage for the *DLP*(1) is 2236, *i.e*. almost 32 times larger.