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

$$ g^x \mod p = h. $$

(1)

Here $2 \leq g \leq p-1$; $1 \leq h \leq p-1$ 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 $2^x \mod 7 = 5$ 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

$$ g_i = g; \ h_i = h; \ x_i = x; $$

$$ q_i = p - 1 \ \text{and} \ \ p - 1 = 2r_1r_2. $$

(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 $R_i := (p-1)/q$;

(4)

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

Let’s define

$$ g_2 := g_i^{r_1} \mod p; $$

(5)

$$ h_2 := h_i^{r_1} \mod p; $$

(6)

If an integer $x_2$ is a solution of equation

$$ g_2^{x_2} \mod p = h_2, \ \text{where} \ x_2 \in [0, q], $$

(7)

then $q$ divides $x_1 - x_2$.

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

$$ h_i g_i^{x_2} \mod p $$

(8)

has a root of power $q$.

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

$$ \left( h_i g_i^{x_2} \right)^{p-1/q} \mod p = 1 $$

(9)

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

$$ h_i g_i^{x_2} \mod p = 1 $$

(10)

or as Equation (7). Q.E.D.
Therefore, the unknown \( x_i \) can be represented as
\[
x_i = x_2 + qx_3
\] (11)
where the integer \( x_3 \) must be on the interval
\[
x_3 \in \left[ 0, \frac{(p - 1)}{q} \right] = \left[ 0, q_3 \right]
\] (12)
After \( x_2 \) is determined, we need to find an integer \( x_3 \), for which the following equation holds
\[
g_i^{x_2 + qx_3} \mod p = h_i.
\] (13)
This equation can be rewritten as
\[
\left( g_i^q \right)^{x_2} = h_i g_i^{-x_2} \mod p
\] (14)
where in contrast with the BSGS algorithm, the value of \( x_2 \) is already known.

\[
g_3 := g_1^{(p-1)/q_3} \mod p;
\] (15)
and
\[
h_3 := h_i g_i^{-x_2} \mod p.
\] (16)

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

Let’s solve
\[
2^{31} \mod 947 = 273,
\] (17)
i.e., here \( g_1 = 2; p = 947; h_1 = 273 \), and \( x_1 \in [1, 946] \).
Let \( q_3 := p - 1 \).
Since \( q_3 = 2r_3 = 2 \times 11 \times 33 \), select \( q_2 = \min_{0 \leq z \leq \sqrt{p-1}} \max \left( z, (p-1)/z \right) = 43 \).
Then \( R_2 := q_2 / q_3 = 22 \) ; \( g_2 := g_i^{86} \mod p = 2^{22} \mod 947 = 41 \); and \( h_2 := h_i^{22} \mod p = 273^{22} \mod 947 = 283 \).
Therefore we need to solve the DLP(2):
\[
41^{22} \mod 947 = 283 \] (7),
where \( x_2 \in [1, 42] \).

\textbf{Remark1}: 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 (8) can be solved using any algorithm for the DLP [3,6,8-10,12].
In this example \( x_2 = 39 \) and \( q_3 = 43 \).
Therefore \( x_1 = 39 + 43x_3 \), where \( x_3 \in \left[ 0, \frac{(p - 1)}{q_3} \right] = [0, 22] \).
To find \( x_3 \) solve the DLP(3):
\[
\left( 2^{43} \right)^{x_3} = 273 \times 2^{-39} \mod 947,
\] which is equivalent to
\[
367^{x_3} = 273 \times 111 = 946 \mod 947. \] (19)
Therefore \( x_3 = 11 \).

Verification: \( 367^{11} \mod 947 = 946 \). (20)
Finally, \( x_1 = 39 + 43 \times 11 = 512 \).

3. Multi-Level Decomposition: Illustrative Example-2

\textit{Initial DLP(1)}: Find an integer \( x_1 \), such that
\[
30^{x_1} \mod 99991 = 45636,
\] (21)
where \( x_1 \in [1, 99990] \).
Because \( 99990 = 303 \times 330 \), select \( q_3 = 330 \) and represent the unknown \( x_1 \) as \( x_1 = x_2 + 330x_3 \).
Since \( R_3 := (p-1)/q_3 = 303 \);
then \( g_3 := g_1^{330} \mod 99991 = 151 \);
and \( h_3 := h_1^{330} \mod 99991 = 64099 \).

\textbf{Remark2}: To better describe the concept of decomposition, a more suitable system of notations is considered below in the following \textbf{Table 1}. These notations are used to describe the process of solving three DLPs.

\textbf{DLP(2)}: Solve \( g_2^{x_2} \mod 99991 = h_2 \),
i.e., \( 151^{x_2} \mod 99991 = 64099 \),
where \( x_2 \in [0, 330] \). (22)
The solution is \( x_2 = 259 \); indeed \( 151^{259} \mod 99991 = 64099 \).

Therefore \( 30^{x_1} \mod 99991 = 45636 \).
Consider the equation
\[
\left( 30^{330} \right)^{x_3} = 30^{-115} \times 45636 \mod 99991 \).
\] Let \( g_3 := 30^{330} \mod 99991 = 2593 \); and
\[
h_3 := 30^{-115} \times 45636
\] = 96658\textsuperscript{15} \times 45636 \mod 99991
\] = 49845.
Therefore, we need to solve \textbf{DLP(3)}:
\[
2593^{x_3} \mod 99991 = 49845, \text{ where}
\] \( x_3 \in [0, 303]. \) (23)
It is easy to verify that \( x_3 = 47 \). Finally, \( x_1 = x_2 + q_3 x_3 = 115 + 330 \times 47 = 15625 \).

\textbf{Decomposition of DLP(2)}: Solve
\[
g_2^{x_2} \mod p = h_2,
\] (24)
where \( x_2 \in [0, q_3] = [0, 330] \).
Table 1. Solutions of DLP(1) via the decomposition of DLP(2) and DLP(3).

<table>
<thead>
<tr>
<th>DLP(1): $g_i^x \mod p = h_i$</th>
<th>Problem A</th>
<th>Problem B</th>
<th>Problem C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs ${g_i; ; p; ; h_i}$</td>
<td>[2; 947; 273]</td>
<td>[2; 947; 641]</td>
<td>[30; 99991; 45636]</td>
</tr>
<tr>
<td>$q_i := p - 1 = 2r_i - 1$, $0 &lt; r_i &lt; p$</td>
<td>$2 \times 11 \times 43$</td>
<td>$2 \times 11 \times 43$</td>
<td>$2 \times 3 \times 11 \times 101$</td>
</tr>
</tbody>
</table>

**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 $q_2 = 330$:

$$q_4 = \min_{0 < x < q_2} \max \left( z, \frac{q_2}{z} \right) = \min \left( \max \left( \frac{z \cdot q_2}{z}, 30 \right) \right)$$

(25)

Let’s represent the unknown in (24) as

$$x_2 = x_4 + q_4 x_5,$$

(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 $h_i$ has an integer root of power 30 modulo $p$.

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

$$h_i^{(p-1)/q_4} \mod p = 1.$$ 

(28)

However, if $h_i^{(p-1)/q_4} \mod p \neq 1$, find an integer $x_4$, which satisfies the equation

$$\left( h_i g_2 \right)^{(p-1)/q_4} \mod p = 1.$$ 

(29)

Let

$$g_i := g_2^{(p-1)/q_4} \mod p;$$ 

(30)

and

$$h_i := h_i^{(p-1)/q_4} \mod p.$$ 

(31)

Now we need to solve the equation

$$g_i^{x_4} \mod p = h_i,$$

(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 $g_1^x \mod p = h_1$, where $g = 2, \; p = 4,000,000,003,231; \; \text{and} \; h = 3,024,336,139,227$.

Then $p - 1 = 863\times2310\times2006491$, where 863 and 2,006,491 are primes.

In this case the initial $DLP(1)$ $g_1^x \mod p = h_1$ is decomposable into two sub-problems: $DLP(2)$ and $DLP(3)$.

**DLP(2):** Compute

$$g_2 := g_1^{(p-1)/q_2} = 2^{3999350} \mod 4000000003231 = 3278213345371;$$

and

$$h_2 := h_1^{(p-1)/q_2} = 3024336139227^{3999350} \mod 4000000003231 = 2084778340641.$$ 

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

$$0 \leq x_2 < q_2 = 2006491;$$

It is easy to verify the solution

$$x_2 = 1853979 \leq 2006491.$$ 

**DLP(3):** Compute

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

and
\[ h_i := h_i g_i^{-x_i} = 3024336139227 \times 2000000001616^{385979} \mod 4000000003231 = 3024336139227 \times 629308445687 \mod 4000000003231 = 2623468766941. \]

Solve \( g_i^{x_i} = h_i \mod m \), where \( 0 \leq x_i = 14622 \leq q_i = (p - 1)/q_i = 1993530 \); and \( q_i = q_i q_2 \).

Then \( x_i = x_i + q_i x_2 = 1853,979,200,6,491,14,622 = 29,340,765,381. \)

It is easy to verify that the solution \( x_i = 14622 \leq 1993530 \).

**Comparison of complexities**: While the size of the required memory/storage for \( DLP(1) \) equals \( T_1 = \lfloor \sqrt{p - 1} \rfloor = 2000000 \); the corresponding memory requirement for \( DLP(2) \) and \( DLP(3) \) are respectively \( T_2 = \lfloor \sqrt{q_2 - 1} \rfloor = \lfloor \sqrt{2006491} \rfloor = 1416 \) and \( T_3 = \lfloor \sqrt{q_3 - 1} \rfloor = \lfloor \sqrt{1993530} \rfloor = 1411. \)

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 \( DLP(3) \)

**Remark 4**: The second problem, \( DLP(2) \), cannot be solved by decomposition since \( q_2 = 2,006,491 \) is a prime integer. However, the third problem, \( DLP(3) \), is decomposable, therefore the speed-up ratio \( S \) can be further increased.

Indeed, select \( q_6 := \min_{0 \leq x \leq \sqrt{q_6}} \max (q_j / z, z) = 2310. \)

Let’s represent \( x_j \) as \( x_j = x_j + q_j x_j \), where \( 0 < x_j < q_j = 2310 \) and \( 0 < x_j < q_j = 863 \), and solve \( DLP(3) \) by decomposition into \( DLP(6) \) and \( DLP(7) \).

**DLP(6)**: Compute \( g_6 := g_3^{(p^2 - 1)/q_6} \mod p \);

and \( h_6 := h_3^{(p^2 - 1)/q_6} \mod p \);

where \( q_6 q_2 = q_3 = 1993530 \);

and solve \( g_6^{x_6} = h_6 \mod 1993531 \);

\( 0 < x_6 < q_6 = 2310. \).

**DLP(7)**: Compute \( g_7 := g_3^{(p^2 - 1)/q_7} \mod p \);

and \( h_7 := h_3^{(p^2 - 1)/q_7} \mod p \);

and solve \( g_7^{x_7} = h_7 \mod 1993531 \);

\( 0 < x_7 < q_7 = 863 \).

Then \( T_6 = \lfloor \sqrt{q_6} \rfloor = 48 \) and \( T_7 = \lfloor \sqrt{q_7} \rfloor = 29 \).

Therefore \( S = T_6 / (T_6 + T_7) = 2000000 / (1416 + 1493) = 1339.6 \)

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

\[ g_{2k} := g_k^{(p^2 - 1)/q_2} \mod p \neq \pm 1. \]

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

\[ w_4 := w_2^{(p^2 - 1)/q_4} = \left\lfloor \frac{w_1^{(p^2 - 1)/q_4} - 1}{(p - 1)/q_4} \right\rfloor (p - 1)/q_4 = \pm 1 \mod p. \]

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 \( g_{2j} \) and \( g_{2j+1} \) 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 $DLP$s with smaller intervals of uncertainty for the corresponding unknowns.

7. Algorithmic Decomposition of $DLP(k)$

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

$$g_k^n \mod p = h_k,$$

(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 $DLP$s: $DLP(2k)$ and $DLP(2k+1)$.

Let

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

$DLP(2k)$: Solve

$$g_{2k}^{x_k} \mod p = h_{2k};$$

(34)

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

(35)

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

(36)

g_{2k} := g_k^{q_k} \mod p;

(37)

and

$h_{2k} := h_k^{q_k} \mod p.$

(38)

$DLP(2k+1)$: Solve

$$g_{2k+1}^{u_{2k+1}} \mod p = h_{2k+1};$$

(39)

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

(40)

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

(41)

g_{2k+1} := g_k^{u_{2k+1}} \mod p;

(42)

and

$h_{2k+1} := h_k g_k^{u_{2k}} \mod p.$

(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 $DLP(2k)$ and $DLP(2k+1)$.

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


APPENDIX

Numeric example as an exercise

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

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

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

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 $x_i = x_2 + g_2 x_3$.

Remarks: 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.