Attacking The Knapsack Public-Key Cryptosystem

Rifaat Z. Khalaf¹, Thakwan Akram Jawad², Hamza B. Habib³

¹ Department of Mathematics, College of Science, University of Diyala, Diyala, Iraq,
² Department of System and Control Engineering, Ninevah University, Ninevah, Iraq,
³ Department of Mathematics, College of Science, University of Diyala, Diyala, Iraq

Abstract
Merkle-Hellman knapsack cryptosystem is a public-key cryptosystem, which means, two keys are used, a public key for the encryption and a private key for the decryption. This cryptosystem is not secure against the cryptosystem attacks, such as LLL Algorithm. In this paper, we introduce a new study to attack the knapsack problem. This study is based on discussing certain facts regarding the public key to extract the multiplier. Also, based on some facts we can find the modulus that is used in the cryptosystem. Therefore, the experimental outcome shows that this method is easier and faster to attack the knapsack cryptosystem.

Keywords: Merkle – Hellman knapsack, public key cryptosystem, cryptoanalysis.

Introduction
A public-key cryptosystem (PKC), which is a concept firstly introduced by Diffie and Hellman in their paper see (Diffie, W., & Hellman, M. E. 1976), is significant in the field of network information security. (Rivest, R.L., et al., 1978) and (Elgamal, T. 1985) present the PKCs, the RSA and Elgamal respectively. These PKCs face some essential issues, for example, the slow speed which affects their applications. Therefore, cryptographers were required to design faster PKCs, such as the knapsack cryptosystems.
(Merkle, R. C., & Hellman, M. E. 1978) developed the first Knapsack system, such that, the private key, which is the super increasing sequence \( S = \{s_i\}_{i=1}^k \), the multiplier \( a \), and the modulus \( n \) are kept secret with the receiver; however, the public key \( T = \{T_i\}_{i=1}^k \) is published. Many other Knapsack-type cryptosystems were found. Nevertheless, (Vaudenay, S. 2001) breaks the Chor-Rivest Cryptosystem, which is one of the few secure systems. In the literature, many techniques were developed, and many trapdoors were found to hide information. For example, (Merkle, R. C., & Hellman, M. E. 1978) present the 0-1 Knapsack problem, compact knapsack problem (Orton, G. A. 1995), multiplicative knapsack problem (Naccache, D., & Stern, J. A. 1997), group factorization problem (Nguyen, P., & Stern, J. 1997), Diophantine equations (Lin, C.H., et al., 1995), using the linear transformation of the secret sets with the Chinese remainder theorem (Murakami, Y., 2010), combining the Legendre symbol with the knapsack problem (Habib, H.B. et al., 2021), and so on. However, almost all the additive Knapsack-type cryptosystems are vulnerable to low–density subset-sum attacks (Coster, M.J., et al., 1991), GCD attacks (Brickell, E.F., & Odlyzko, A.M., 1992).

In this paper, we design a new study to attack the knapsack problem based on knowing the public key. By knowing the public key, we can apply some conditions to recover the multiplier. Then we can find a set of numbers, such that, each number will be tested in turn as a modulus to get the corresponding set. Nevertheless, not all of the corresponding sets are super-increasing sequences, there will be only one corresponding super increasing sequence to the used actual modulus from the set. The rest of this paper is given as follows, In section 2, the proposed attacking method is introduced. Finally, in Section 3 a conclusion is given.

The Standard Knapsack Cryptosystem Alice and Bob use this cryptosystem as follows, see (Merkle, R. C., & Hellman, M. E., 1978).

**The key generation process:** This process is performed by Alice as

1. Alice selects a super-increasing sequence \( S = \{s_i\}_{i=1}^k \).
2. Alice chooses the numbers \( n \) and \( a \), such that, \( n \geq \sum_i s_i \) and \( \gcd(a, n) = 1 \). Therefore, the private key is \((S, n, a)\).
3. Alice calculates \( T_i = a \times s_i \pmod{n} \), where \( 1 \leq i \leq k \), then \( T = \{T_i\}_{i=1}^k \) is the public key.

**The encryption process:** Bob converts the message \( M = (M_1, M_2, \cdots, M_k) \) to a binary form \( \beta = (b_1, b_2, \cdots, b_k) \). Then, the ciphertext is given by

\[
C_i = \sum_{j=1}^{k} T_j \ast M_{ij}
\]
and $C = \{C_i\}_{i=1}^k$ will be sent to Alice.

**The decryption process:** When the ciphertext is received, Alice follows the steps.

1. Alice finds $a^{-1}$ of $a$ modulo $n$ by applying the Euclidean Algorithm see (Rosen, K. H. 2011).
2. She computes $d_i = C_i * a^{-1} \pmod{n}$, where $1 \leq i \leq k$.
3. Alice performs $d_i - s_i$, where $s_i \leq d_i$, and continuing the subtraction for the rest of the elements of $S$ to obtain zero. This forms $b_i$ which represents $M_i$ in $M$.

Thus, the gets the original message $M = (M_1, M_2, \ldots, M_k)$.

Proposed Attacking Algorithm

The proposed algorithm works on finding the multiplier $a$ and the modulus $n$ by discussing certain conditions and facts see Figure 1.

Theoretical Part

**First condition:** If $T_1$ is an even number, then the multiplier $a$ is given by

$$a = \frac{T_1}{2} \quad (1)$$

**Second Condition:** If $T_1$ is an odd and $T_2$ is an even, then

$$a = \frac{T_2}{2} + T_k \quad (2)$$

**Third condition:** If both $T_1$ and $T_2$ are odd, then there are two cases.

**Case 1:** If $T_1$ and $T_2$ have the same number of digits, then

$$a = (|T_1 - T_2|) + T_1. \quad (3)$$

**Case 2:** If $(|T_1 - T_2|)$ is two or more digits, then

$$a = (|T_1 - T_2|). \quad (4)$$
Moreover, knowing the multiplier $a$ along with the facts below will provide Miro with the sorted set $G$, such that, the used modulus in the cryptosystem belongs to $G$. The facts are as follows:

**Fact 1:** $\gcd(a, n) = 1$.

**Fact 2:** If $T_j$, where $1 \leq j \leq k$, is the largest number in $T$, then $n > T_j$.

**Fact 3:** The corresponding set $S$ should be a super increasing sequence.

Miro needs to consider the numbers in $G$ in turn. Each number will be tested as a modulus to get a corresponding set, see Eq. (5) below.

\[
T_i \cdot a^{-1} \equiv S_i \pmod{n}. \tag{5}
\]

To ensure that the corresponding set is the correct super-increasing sequence, Miro uses Eq. (6),

\[
\text{CT}_i \cdot a^{-1} \equiv S_i \pmod{n}, \tag{6}
\]

where $\text{CT}_i$ is the $i^{th}$ ciphertext block.

*Figure 1: The figure illustrates the proposed attacking Algorithm*
The Experimental Part
Suppose that the super increasing sequence is given as \( S = \{3, 5, 11, 20, 41\} \). Let \( a = 47, \ n = 83 \), then \( a^{-1} = 53 \) implies the public key is \( T = \{58, 69, 19, 27, 18\} \). Also, suppose the transmitted message is “AHMED”. Then, the encryption and decryption processes are shown in Table 1.

Table 1: The table shows the Encryption and Decryption Processes

<table>
<thead>
<tr>
<th>The Alphabet</th>
<th>Corresponding Integers</th>
<th>Binary form</th>
<th>Ciphertext</th>
<th>( S )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>00001</td>
<td>18</td>
<td>41</td>
</tr>
<tr>
<td>H</td>
<td>8</td>
<td>01000</td>
<td>69</td>
<td>5</td>
</tr>
<tr>
<td>M</td>
<td>13</td>
<td>01101</td>
<td>106</td>
<td>57</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
<td>00101</td>
<td>37</td>
<td>52</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>00100</td>
<td>19</td>
<td>11</td>
</tr>
</tbody>
</table>

Now, suppose that Miro wants to attack the given cryptosystem, it is obvious that she knows the public key, \( T = \{58, 59, 19, 27, 18\} \). Since \( T_1 \) is an even, then by the first condition of the proposed algorithm

\[
\frac{58}{2} + 18 = 29 + 18 = 47
\]

Miro considers the facts below.

**Fact 1:** \( n \) is the modulus, then \( \gcd(47, n) = 1 \).

**Fact 2:** The largest number in \( T \) is 59, then \( n > 59 \).

**Fact 3:** \( S \) is a super-increasing sequence.

Then, she can conclude that the modulus \( n \) is one of the elements of the set \( G \). That is,

\[
G = \{60, 61, \ldots, 82, 83, \ldots\}.
\]

To find the actual modulus \( n \), Miro needs to discuss the numbers in \( G \) in turn as follows.

**Case 1:** Let \( n = 60, \ a = 47 \) and \( a^{-1} = 23 \), then

\[
58 \times 23 \equiv 14 \pmod{60}
\]

\[
69 \times 23 \equiv 27 \pmod{60}
\]
Thus, $S = \{14, 27, 17, 21, 54\}$ which is not a super-increasing sequence. So Miro moves to the next number in $G$ which is 61.

**Case 2:** Let $n = 61$, $a = 47$ and $a^{-1} = 13$, then by the same way she can get $S = \{22, 43, 3, 46, 51\}$, which is not a super-increasing sequence.

Then, she continues testing the rest of the numbers until she gets to case 6.

**Case 6:** Let $n = 65$, $a = 47$ and $a^{-1} = 18$, then $S = \{4, 7, 17, 31, 64\}$. To make sure this is the correct super-increasing sequence, Miro uses Eq. (6), then

$$
\begin{align*}
18 \times 18 &\equiv 64 \pmod{65} \\
69 \times 18 &\equiv 7 \pmod{65} \\
106 \times 18 &\equiv 23 \pmod{65} \\
37 \times 18 &\equiv 16 \pmod{65} \\
19 \times 18 &\equiv 17 \pmod{65}
\end{align*}
$$

Therefore, $S$ here is not the correct super-increasing sequence because both of the resulting numbers 23 and 16 are not in $S$.

Miro will continue in turn with the rest of the numbers in $G$, for more details see Table 2, till she reaches case 24.

**Case 24:** $n = 83$, $a = 47$ and $a^{-1} = 53$, then

$$
\begin{align*}
58 \times 53 &\equiv 3 \pmod{83} \\
69 \times 53 &\equiv 5 \pmod{83} \\
19 \times 53 &\equiv 11 \pmod{83} \\
27 \times 53 &\equiv 20 \pmod{83} \\
18 \times 53 &\equiv 41 \pmod{83}
\end{align*}
$$

Then, $S = \{3, 5, 11, 20, 41\}$. Now, using Formula (6)

$$
\begin{align*}
18 \times 53 &\equiv 41 \pmod{83} \\
69 \times 53 &\equiv 5 \pmod{83}
\end{align*}
$$
\[106 \times 53 \equiv 57 \pmod{83}\]
\[37 \times 53 \equiv 52 \pmod{83}\]
\[19 \times 53 \equiv 11 \pmod{83}\]

Thus, \( S = \{3, 5, 11, 20, 41\} \) is the actual used super-increasing sequence.

Table 2: The table shows the corresponding sets to each modulus

<table>
<thead>
<tr>
<th>Cases</th>
<th>( n )</th>
<th>( S )</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>60</td>
<td>{14,27,17,21,54}</td>
<td>Not Super</td>
</tr>
<tr>
<td>2</td>
<td>61</td>
<td>{22,43,3,46,51}</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>62</td>
<td>{54,45,7,23,36}</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>63</td>
<td>{20,39,50,18,54}</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>64</td>
<td>{38,11,29,21,14}</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>65</td>
<td>{4,7,17,31,64}</td>
<td>Yes/Fail</td>
</tr>
<tr>
<td>7</td>
<td>66</td>
<td>{56,45,65,9,6}</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>67</td>
<td>{44,20,56,2,46}</td>
<td>No</td>
</tr>
<tr>
<td>9</td>
<td>68</td>
<td>{62,55,25,57,38}</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>69</td>
<td>{35,0,65,27,18}</td>
<td>No</td>
</tr>
<tr>
<td>11</td>
<td>70</td>
<td>{34,67,57,11,54}</td>
<td>No</td>
</tr>
<tr>
<td>12</td>
<td>71</td>
<td>{39,6,14,61,17}</td>
<td>No</td>
</tr>
<tr>
<td>13</td>
<td>72</td>
<td>{38,3,5,45,54}</td>
<td>No</td>
</tr>
<tr>
<td>14</td>
<td>73</td>
<td>{9,17,47,13,33}</td>
<td>No</td>
</tr>
<tr>
<td>15</td>
<td>74</td>
<td>{28,55,13,73,24}</td>
<td>No</td>
</tr>
<tr>
<td>16</td>
<td>75</td>
<td>{14,27,2,66,69}</td>
<td>No</td>
</tr>
<tr>
<td>17</td>
<td>76</td>
<td>{74,71,57,41,2}</td>
<td>No</td>
</tr>
<tr>
<td>18</td>
<td>77</td>
<td>{34,67,43,53,61}</td>
<td>No</td>
</tr>
<tr>
<td>19</td>
<td>78</td>
<td>{56,33,17,57,12}</td>
<td>No</td>
</tr>
<tr>
<td>20</td>
<td>79</td>
<td>{13,25,71,51,34}</td>
<td>No</td>
</tr>
<tr>
<td>21</td>
<td>80</td>
<td>{54,27,77,21,14}</td>
<td>No</td>
</tr>
<tr>
<td>22</td>
<td>81</td>
<td>{65,48,59,54,9}</td>
<td>No</td>
</tr>
<tr>
<td>23</td>
<td>82</td>
<td>{78,73,51,25,44}</td>
<td>No</td>
</tr>
<tr>
<td>24</td>
<td>83</td>
<td>{3,5,11,20,41}</td>
<td>Yes/Success</td>
</tr>
</tbody>
</table>

Conclusion
Attacking knapsack cryptosystem is presented in this paper. The attack is based on applying some conditions to the public key to find the multiplier. Finding the multiplier along with some fundamental facts regarding the structure of the knapsack cryptosystem provide a set of guess numbers. This set contains the used modulus in the cryptosystem to encrypt and decrypt data, and...
then the required super increasing sequence is found. As a result of the uniqueness of the used super increasing sequences which satisfies the knapsack cryptosystem, the proposed attacking method is faster and easy to be applied on the cryptosystem.

References


http://www.webology.org
