Lattice Reduction Attack on the Knapsack

Mark Stamp

1 Merkle–Hellman Knapsack

Every private in the French army carries a Field Marshal wand in his knapsack.
— Napoleon Bonaparte

The Merkle–Hellman knapsack cryptosystem [5] was one of the first proposed public key cryptosystems. This cipher utilizes a few elementary, but nonetheless clever mathematical ideas. Because of its historical significance and since it is easy to understand, we examine it first. The cipher is based on a mathematical problem which is known to be NP-complete [2].

The subset sum or *knapsack problem* can be stated as follows: Given a set of r weights

$$W = (w_0, w_1, \ldots, w_{r-1})$$

and a sum X, find $x_0, x_1, \ldots, x_{r-1}$, where each $x_i \in \{0,1\}$, so that

$$X = x_0 w_0 + x_1 w_1 + \cdots + x_{r-1} w_{r-1},$$

provided that this is possible. Note that the x_i simply select a subset of the weights.

For example, suppose that the weights are $W = (4, 3, 9, 1, 12, 17, 19, 23)$ and the given sum is $X = 35$. Then, a solution to the subset problem exists and is given by $x = (01011010)$, since

$$0 \cdot 4 + 1 \cdot 3 + 0 \cdot 9 + 1 \cdot 1 + 1 \cdot 12 + 0 \cdot 17 + 1 \cdot 19 + 0 \cdot 23 = 35.$$

For this set of weights, if $X = 6$, the problem does not have a solution.

While the general knapsack problem is NP-complete, a special type of knapsack known as a *superincreasing knapsack* can be solved efficiently. A superincreasing knapsack is a set W that, when ordered from least to greatest, has the property that each weight is greater than the sum of all the previous weights. For example,

$$W = (2, 3, 6, 13, 29, 55, 112, 220) \tag{1}$$

is a superincreasing knapsack.

It is straightforward to solve a superincreasing knapsack problem. For example, suppose that we are given the set of weights in (1) and the sum $X = 76$. Since X is less than 112, we must have $x_7 = x_6 = 0$. Then, since $X > 55$ and we have $2 + 3 + 6 + 13 + 29 < 55$, it must be the case that $x_5 = 1$. That is, if we do not select the weight 55, then we cannot possibly reach the desired sum, since the sum of all remaining weights is less than 55, due to the superincreasing property.

Now, let $X_1 = X - 55 = 21$. Since $13 < X_1 < 29$, we must have that $x_4 = 0$ and $x_3 = 1$. Continuing in this manner, we find $x = (10110100)$ which is easily verified to be correct since $76 = 2 + 6 + 13 + 55$. This process yields an efficient (linear time) algorithm to solve any superincreasing knapsack problem.

Merkle and Hellman's [5] idea was to disguise a superincreasing knapsack S through the use of a mathematical transformation to make it look like an arbitrary knapsack T. The disguised knapsack T is made public by Alice and T acts as Alice's public key. When Alice receives a ciphertext, she applies the inverse of the transformation to convert the problem back to the superincreasing case. Alice decrypts by solving the resulting superincreasing knapsack problem. Without knowledge of the transformation, it would appear that a cryptanalyst must solve a general knapsack, which is a hard problem. However, there is a shortcut attack, which we describe below. But first we discuss the the knapsack cryptosystem in more detail.

To create her public and private keys, Alice first chooses a superincreasing knapsack $S = (s_0, s_1, \ldots, s_{r-1})$. To convert S into T, she also chooses a conversion factor m and a modulus n, where $\gcd(m, n) = 1$ and n is greater than the sum of all elements of S. The transformed knapsack is computed as

$$T = (s_0 m \pmod{n}, s_1 m \pmod{n}, \ldots, s_{r-1} m \pmod{n})$$

and T is made public. Alice's private key consists of S and $m^{-1} \pmod{n}$. Suppose Bob wants to send a message of r bits to Alice. Bob first converts his plaintext into a binary block B. He then uses the 1 bits of B to select the elements of T, which are then summed to give the ciphertext block C. Alice recovers the plaintext B, by using the private key to compute $Cm^{-1} \pmod{n}$, and solves using her superincreasing knapsack. To encrypt longer messages, multiple blocks are encrypted.

To make things more concrete, consider the following example. Suppose that Alice chooses the superincreasing knapsack

$$S = (2, 3, 7, 14, 30, 57, 120, 251),$$

along with $m = 41$ and modulus $n = 491$. To transform S into a general knapsack T, Alice performs the following computations

$$\begin{aligned}
2m &= 2 \cdot 41 = 82 \pmod{491} \\
3m &= 3 \cdot 41 = 123 \pmod{491} \\
7m &= 7 \cdot 41 = 287 \pmod{491} \\
14m &= 14 \cdot 41 = 83 \pmod{491} \\
30m &= 30 \cdot 41 = 248 \pmod{491} \\
57m &= 57 \cdot 41 = 373 \pmod{491} \\
120m &= 120 \cdot 41 = 10 \pmod{491} \\
251m &= 251 \cdot 41 = 471 \pmod{491}.
\end{aligned}$$

Then Alice's public key is

$$T = (82, 123, 287, 83, 248, 373, 10, 471).$$

Alice's private key consists of

$$S = (2, 3, 7, 14, 30, 57, 120, 251)$$

and

$$m^{-1} \pmod{n} = 41^{-1} \pmod{491} = 12.$$

Now, suppose that Bob wants to encrypt the message $M = 150$ for Alice. He first converts 150 to binary, that is 10010110. He then uses the 1 bits to select the elements of T that are summed to give the ciphertext. In this example, Bob computes the ciphertext

$$C = 82 + 83 + 373 + 10 = 548$$

and sends C to Alice. To decrypt this ciphertext, Alice uses her private key to compute

$$Cm^{-1} \pmod{n} = 548 \cdot 12 \pmod{491} = 193.$$

She then solves the superincreasing knapsack S for 193 and she recovers the message in binary 10010110 or, in decimal, $M = 150$.

That this decryption process works can be verified by using elementary properties of modular arithmetic. In the particular example considered above, we have

$$\begin{aligned}548m^{-1} &= 82m^{-1} + 83m^{-1} + 37m^{-1} + 10m^{-1} \\ &= 2mm^{-1} + 14mm^{-1} + 57mm^{-1} + 120mm^{-1} \\ &= 2 + 14 + 57 + 120 \\ &= 193 \pmod{491}.\end{aligned}$$

In general, due to the linearity of the process used to convert from the superincreasing knapsack S into the public key knapsack T, knowledge of m^{-1} makes it easy to convert the ciphertext to the superincreasing case. Without Alice's private key, $(S, m^{-1} \pmod{n})$, the attacker Trudy needs to find a subset of T which sums to the ciphertext value C. This appears to be a general knapsack problem, which is intractable.

By converting the superincreasing knapsack into the general knapsack through the use of modular arithmetic, a trapdoor is introduced into the knapsack. Without m, it is not clear how to find the conversion factor m^{-1}. The one-way feature results from the fact that it is easy to encrypt with the general knapsack, but it is (hopefully) difficult to decrypt without the private key. But with the private key, the problem can be converted into a superincreasing knapsack, which is easy to solve and thus enables the intended recipient to easily decrypt.

However, this cryptosystem was shown to be insecure by Shamir [7] in 1983. It turns out that the "general knapsack" (the public-key) which arises in the Merkle–Hellman cryptosystem is not general enough. Instead, it is a highly structured case of the knapsack and Shamir's lattice reduction attack is able to take advantage of this fact. Shamir's ingenious method of attack is dicussed in the next section.

1.1 Lattice-Reduction Attack

Lattice reduction is a powerful technique which can be used to solve many different types of combinatorial problems. We first describe the lattice reduction method, as discussed in [8], and then illustrate how it can be used to attack the Merkle–Hellman knapsack cryptosystem. Some elementary linear algebra is used in this section; see the Appendix for an overview of the necessary linear algebra.

Consider, for example, the vectors

$$c_0 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \text{ and } c_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Since c_0 and c_1 are linearly independent, any point in the plane can be uniquely represented by $\alpha_0 c_0 + \alpha_1 c_1$, where α_0 and α_1 are real numbers. If we restrict the coefficients to integers, that is, we require that α_0 and α_1 are integers, then we obtain a *lattice* consisting of discrete points in the plane. Figure 1 illustrates the lattice spanned by c_0 and c_1. In general, a lattice $\mathcal{L}$ is the set of all linear combinations of a set of column vectors c_i with integer coefficients.

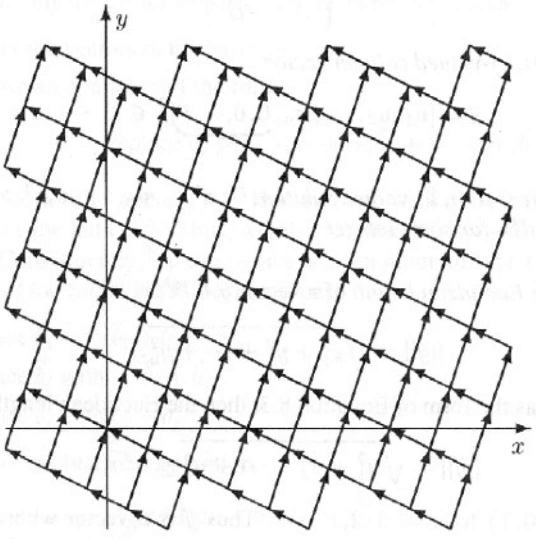

Figure 1: A lattice in the plane.

Given an $m \times n$ matrix A and an $m \times 1$ matrix B, suppose we want to find a solution U to the matrix equation $AU = B$, with the restriction that U consists entirely of 0s and 1s. If U is a solution to $AU = B$, then the block matrix equation

$$MV = \begin{bmatrix} I_{n\times n} & 0_{n\times 1} \\ A_{m\times n} & -B_{m\times 1} \end{bmatrix} \begin{bmatrix} U_{n\times 1} \\ 1_{1\times 1} \end{bmatrix} = \begin{bmatrix} U_{n\times 1} \\ 0_{m\times 1} \end{bmatrix} = W \tag{2}$$

holds true, since $MV = W$ is equivalent to $U = U$ and $AU - B = 0$. Consequently, finding a solution V to the block matrix equation $MV = W$ is equivalent to finding a solution U to the original matrix equation $AU = B$. Note that the columns of M are linearly independent, since the $n \times n$ identity matrix appears in the upper left and the final column begins with n zeros.

Let $c_0, c_1, c_2, \ldots, c_n$ be the $n + 1$ columns of the matrix M in (2) and let $v_0, v_1, v_2, \ldots, v_n$ be the elements of V. Then

$$W = v_0 c_0 + v_1 c_1 + \cdots + v_n c_n. \tag{3}$$

We have $MV = W$, where

$$W = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{n-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} U \\ \vec{0} \end{bmatrix} \tag{4}$$

and we want to determine U. Instead of solving linear equations to obtain V, we will find U by determining W. Note that because of (3), W is in the lattice $\mathcal{L}$, spanned by the columns of M.

The Euclidean length of a vector $Y = [y_0, y_1, \ldots, y_{n+m-1}]^T$ is

$$\|Y\| = \sqrt{y_0^2 + y_1^2 + \cdots + y_{n+m-1}^2}.$$

However, the length of a vector W in (4) is

$$\|W\| = \sqrt{u_0^2 + u_1^2 + \cdots + u_{n-1}^2} \leq \sqrt{n},$$

which is much "shorter" than a typical vector in $\mathcal{L}$. Furthermore, W has a very special form, since its first n entries consist of 0s and 1s with its last m entries being all 0. Is it possible to take advantage of this special structure to find W?

In 1982, Lenstra, Lenstra and Lovàsz [4] discovered the so-called LLL Algorithm, which provides an efficient method to find short vectors in a lattice. In Table 1, we give an outline of their algorithm in pseudo-code, where GS(M) refers to the Gram–Schmidt process, which returns an orthonormal basis for the subspace spanned by the columns of M. The Gram–Schmidt process appears in Table 2. Note that a small number of lines of pseudo-code suffices to specify the entire LLL Algorithm.

Table 1: LLL Algorithm

```
        // find short vectors in the lattice spanned
        // by the columns of $M = (b_0, b_1, \ldots, b_n)$
        repeat
            $(X, Y) = \mathrm{GS}(M)$
            for $j = 1$ to $n$
                for $i = j - 1$ to 0
                    if $|y_{ij}| > 1/2$ then
                        $b_j = b_j - \lfloor y_{ij} + 1/2 \rfloor b_i$
                    end if
                next $i$
            next $j$
            $(X, Y) = \mathrm{GS}(M)$
            for $j = 0$ to $n - 1$
                if $\|x_{j+1} + y_{j,j+1} x_j\|^2 < \frac{3}{4}\|x_j\|^2$ then
                    swap$(b_j, b_{j+1})$
                    goto abc
                end if
            next $j$
            return$(M)$
abc:        continue
        forever
```

With clever insight, Shamir [7] realized that lattice reduction could be used to attack the Merkle–Hellman knapsack cryptosystem. Suppose that Bob's public knapsack is given by $T = (t_0, t_1, \ldots, t_{r-1})$, and Alice sends Bob a ciphertext block C, encrypted with Bob's public knapsack. Since the attacker, Trudy, knows the public knapsack T and C, she can break the system if she is able to solve the matrix equation $TU = C$, where U is an $r \times 1$ column matrix consisting of 0s and 1s.

Table 2: Gram–Schmidt Process

```
// Gram–Schmidt $M = (b_0, b_1, \ldots, b_n)$
GS($M$)
    $x_0 = b_0$
    for $j = 1$ to $n$
        $x_j = b_j$
        for $i = 0$ to $j - 1$
            $y_{ij} = (x_i \cdot b_j)/||x_i||^2$
            $x_j = x_j - y_{ij} x_i$
        next $i$
    next $j$
    return$(X, Y)$
end GS
```

Trudy can rewrite the matrix equation $TU = C$ in block matrix form as

$$MV = \begin{bmatrix} I_{r\times r} & 0_{r\times 1} \\ T_{1\times r} & -C_{1\times 1} \end{bmatrix} \begin{bmatrix} U_{r\times 1} \\ 1_{1\times 1} \end{bmatrix} = \begin{bmatrix} U_{r\times 1} \\ 0_{1\times 1} \end{bmatrix} = W$$

and apply the LLL Algorithm to the matrix M. The resulting short vectors which are obtained can be checked to see if they have the special form required of W, which is a column vector where the first r entries are all 0 or 1 and last entry is 0. The LLL Algorithm will not always produce the desired vector and therefore, the attack is not always successful. However, in practice, the lattice reduction attack is highly effective against the original Merkle–Hellman knapsack.

To illustrate the lattice reduction attack, suppose Alice constructs her knapsack key pair from the superincreasing knapsack

$$S = (s_0, s_1, \ldots, s_7) = (2, 3, 7, 14, 30, 57, 120, 251),$$

with $m = 41$ and modulus $n = 491$. Then, $m^{-1} = 12 \pmod{491}$. The corresponding general knapsack T is obtained by computing $t_i = 41 s_i \pmod{491}$, for $i = 0, 1, 2, \ldots, 7$, which was found above to be

$$T = (t_0, t_1, \ldots, t_7) = (82, 123, 287, 83, 248, 373, 10, 471).$$

Alice's knapsack key pair is defined by

Public key: T

and

$$\text{Private key: } S \text{ and } m^{-1} \pmod n.$$

Suppose Bob wants to encrypt the message $M = 10010110$ for Alice. Then, as discussed above, Bob computes

$$1 \cdot t_0 + 0 \cdot t_1 + 0 \cdot t_2 + 1 \cdot t_3 + 0 \cdot t_4 + 1 \cdot t_5 + 1 \cdot t_6 + 0 \cdot t_7 = 548$$

and sends ciphertext $C = 548$ to Alice.

Now, suppose that Trudy wants to recover the plaintext that corresponds to ciphertext $C = 548$. Since Trudy knows the public key T and ciphertext $C = 548$, she needs to find a set of u_i, for $i = 0, 1, \ldots, 7$, with the restriction that each $u_i \in \{0, 1\}$, and

$$82u_0 + 123u_1 + 287u_2 + 83u_3 + 248u_4 + 373u_5 + 10u_6 + 471u_7 = 548.$$

This can be written as the matrix equation

$$T \cdot U = 548,$$

where T is Alice's public knapsack and $U = (u_0, u_1, \ldots, u_7)$, and the a_i are unknown, but each must be either 0 or 1. This is of the form $AU = B$ (as discussed above), so Trudy rewrites the matrix equation as $MV = W$ and applies the LLL Algorithm to M. In this case, Trudy finds

$$M = \begin{bmatrix} I_{8\times 8} & 0_{8\times 1} \\ T_{1\times 8} & -C_{1\times 1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 82 & 123 & 287 & 83 & 248 & 373 & 10 & 471 & -548 \end{bmatrix}.$$

The LLL Algorithm outputs a matrix M', consisting of short vectors in the lattice spanned by the columns of the matrix M. In this example, LLL yields

$$M' = \begin{bmatrix} -1 & -1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & -1 & 1 & 2 \\ 1 & -1 & -1 & 1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -2 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 \\ 1 & -1 & 1 & 0 & 0 & 1 & -1 & 2 & 0 \end{bmatrix}.$$

The entries in the fourth column of M' have the correct form to be a solution to this knapsack problem. Therefore, Trudy obtains the putative solution

$$U = (1,0,0,1,0,1,1,0).$$

Using the public key and ciphertext $C = 548$, she can easily verify that U is indeed the original plaintext sent by Bob.

1.2 Knapsack Conclusion

Much research has been done on the knapsack problem since the Merkle–Hellman cryptosystem was broken. Several different knapsack variants have been created and some of these appear to yield secure cryptosystems. However, people have been reluctant to use these systems, since "knapsack" continues to be equated with "broken," even to this day. For more information on knapsack cryptosystems, see [1, 3, 6].

References

[1] Y. Desmedt, What happened with knapsack cryptographic schemes?, *Performance Limits in Communication, Theory and Practice*, J. K. Skwirzynski, ed., Kluwer, pp. 113–134, 1988
Cited on page 10

[2] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP–Completeness*, W. H. Freeman & Company, 1979
Cited on page 1

[3] M. K. Lai, Knapsack cryptosystems: the past and the future, March 2001, at `www.cecs.uci.edu/~mingl/knapsack.html`
Cited on page 10

[4] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovàsz, Factoring polynomials with rational coefficients, *Mathematische Annalen*, Vol. 261, No. 4, 1982, pp. 515–534
Cited on page 6

[5] R. Merkle and M. Hellman, Hiding information and signatures in trapdoor knapsacks, *IEEE Transactions on Information Theory*, Vol. IT-24, No. 5, 1978, pp. 525–530
Cited on pages 1 and 2

[6] A. M. Odlyzko, The rise and fall of knapsack cryptosystems, at `www.research.att.com/~amo/doc/arch/knapsack.survey.pdf`
Cited on page 10

[7] A. Shamir, A polynomial-time algorithm for breaking the basic Merkle–Hellman cryptosystem, *IEEE Transactions on Information Theory*, Vol. IT-30, No. 5, September 1984, pp. 699–704
Cited on pages 4 and 7

[8] M. Stamp, *Information Security: Principles and Practice*, Wiley-Interscience, 2005
Cited on page 4