Understanding the mathematics of the Knapsack public key cryptosystem

SUPREME PAUDEL, Luther College, USA

The Merkle–Hellman knapsack cryptosystem is an early public-key cryptosystem based on the NP-complete subset sum problem. Although the original scheme has been proven insecure, it remains a valuable subject of study for understanding the mathematical foundations of cryptographic systems. This paper is a guide for studying the mathematics of the Merkle-Hellman knapsack cryptosystem, beginning with a review of relevant modular arithmetic and proceeding through key theorems and proofs. At the end, we examine its vulnerabilities and explain why the cryptosystem is no longer considered secure.

1 MODULAR ARITHMETIC

Suppose you divide two integers a and n, then we will have the following equation:

$$\frac{a}{n} = q \text{ remainder } r$$

where a is the dividend, n is the divisor, q is the quotient, and r is the remainder. In this case, $a \bmod n = r$.
Another way to write this is $a = n \cdot q + r$ where $0 \leq r < n$. In fact, this way is more useful when it comes to dealing with modulus arithmetic as we will see.

Theorem 1: Proof of $(a + n \cdot k) \bmod n = a \bmod n$ *where* $a, k \in \mathbb{Z}, n \in \mathbb{Z} - \{0\}$.
Let $(a + n \cdot k) \bmod n = r$. As we saw earlier, we can write that as $a + n \cdot k = n \cdot q_1 + r$. So,

$$\begin{aligned} a &= n \cdot q_1 - n \cdot k + r \\ &= n \cdot (q_1 - k) + r \\ &= n \cdot q_2 + r \end{aligned}$$

So, $a \bmod n = r$. Therefore, $(a + n \cdot k) \bmod n = a \bmod n$ ■

Theorem 2: Proof of $((a \bmod n) + (b \bmod n)) \bmod n = (a + b) \bmod n$ *where* $a, b \in \mathbb{Z}$.
Let us reuse the idea from the *Theorem 1* proof. let $a \bmod n = r_1$, and $b \bmod n = r_2$. So, $a = n \cdot k_1 + r_1$ and $b = n \cdot k_2 + r_2$. Let $r_1 + r_2 = q \cdot n + r_3$, where $0 \leq r_3 < n$. Let us look at $a + b$.

$$\begin{aligned} a + b &= n \cdot k_1 + n \cdot k_2 + r_1 + r_2 \\ &= n \cdot (k_1 + k_2) + r_1 + r_2 \\ &= n \cdot k_3 + r_3 \end{aligned}$$

So, $(a + b) \bmod n = r_3$. Now, let us look at the left-hand side of the equation we are trying to prove.

$$\begin{aligned} ((a \bmod n) + (b \bmod n)) \bmod n &= (r_1 + r_2) \bmod n \\ &= q \cdot n + r_3 \bmod n \\ &= r_3 \bmod n \\ &= r_3 \end{aligned}$$

Hence, $((a \bmod n) + (b \bmod n)) \bmod n = (a + b) \bmod n = r_3$■

Author's address: Supreme Paudel, Luther College, Decorah, USA, paudsu01@luther.edu.

Theorem 3: Proof of $((a \bmod n) \cdot (b \bmod n)) \bmod n = (a \cdot b) \bmod n$ where $a, b \in \mathbb{Z}$.
Let $a \bmod n = r_1$, and $b \bmod n = r_2$. So, $a = n \cdot k_1 + r_1, b = n \cdot k_2 + r_2$.
Now $r_1 \cdot r_2 = n \cdot k_3 + r_3,$ where $0 \leq r_3 < n$.

$$\begin{aligned}((a \bmod n) \cdot (b \bmod n)) \bmod n &= (r_1 \cdot r_2) \bmod n \\ &= (n \cdot k_3 + r_3) \bmod n \\ &= r_3 \bmod n \\ &= r_3\end{aligned}$$

Now, let us consider the right-hand side of the equation we are trying to prove.

$$\begin{aligned}(a \cdot b) \bmod n &= n^2 \cdot k_1 \cdot k_1 + n \cdot k_1 \cdot r_2 + n \cdot k_2 \cdot r_1 + r_1 \cdot r_2 \\ &= (n \cdot k_4 + r_1 \cdot r_2) \bmod n \\ &= (r_1 \cdot r_2) \bmod n \\ &= r_3\end{aligned}$$

Hence, $((a \bmod n) \cdot (b \bmod n)) \bmod n = (a \cdot b) \bmod n = r_3$■

Definitions of multiplicative inverse and totient function.

> Multiplicative inverse: A multiplicative inverse of $x \bmod n$, denoted by $x^{-1} \bmod n$ would be the number such that $(x \cdot x^{-1}) \bmod n = 1 \bmod n$.
> Totient function: $\Phi(n)$ is the count of the positive integers less than n that are relatively prime to n.

2 THE EUCLIDEAN ALGORITHM AND BÉZOUT'S IDENTITY

A lot of public cryptosystems like the Merkle–Hellman knapsack cryptosystem rely on having two numbers that are relatively prime to each other because the inverse $x^{-1} \bmod n$ exists only when x and n are relatively prime to each other. This inverse can help during the decoding process because $x \cdot x^{-1} \bmod n = 1 \bmod n$.
Let us first define $gcd(a, b)$ before we start using it. The greatest common divisor of integers a and b is the product of their shared primes to the minimum order. Note that the $gcd(a, b) = 1$ if a and b are relatively prime to each other since they share no primes.

Theorem 4: If $a = b \cdot q + r$, then $gcd(a, b) = gcd(b, r)$.
Let D be the set of divisors d such that $d|a$ and $d|b$. On the other hand, let D' be the set of divisors d' such that $d'|b$ and $d'|r$. Now, for $d \in D$, $d|(a - bq)$ since $d|a$ and $d|b$. Since $d|(a - bq)$ and $r = a - bq$, $d|r$. So, $D \subseteq D'$.
Similarly, for $d' \in D'$, $d'|(bq + r)$ since $d'|b$ and $d'|r$. Since $d'|(bq + r)$ and $a = bq + r$, $d'|a$. So, $D' \subseteq D$.
Since $D' \subseteq D$, and $D \subseteq D'$, $D = D'$. Since the set of common divisors of (a, b) and (b, r) are the same, $gcd(a, b) = gcd(b, r)$■

The Euclidean algorithm for $gcd(a, b)$.
Let a and b be two positive integers. Using *Theorem 1*, we can say $a = b \cdot q + r,$ such that $0 \leq r < b$. Now, we can

actually form a chain of equations in the following way:

$$a = b \cdot q + r \text{ such that } 0 \leq r < b$$
$$b = r \cdot q_1 + r_1 \text{ such that } 0 \leq r_1 < r$$
$$r = r_1 \cdot q_2 + r_2 \text{ such that } 0 \leq r_2 < r_1$$
$$\vdots$$
$$r_{i-2} = r_{i-1} \cdot q_i + r_i \text{ such that } 0 \leq r_i < r_{i-1}$$
$$r_{i-1} = r_i \cdot q_{i+1} + 0$$

Using *Theorem 4*, we know that if $a = bq + r$ then $gcd(a, b) = gcd(b, r)$. Applying this to our chain of equations, we get:

$$gcd(a, b) = gcd(b, r)$$
$$gcd(b, r) = gcd(r, r_1)$$
$$gcd(r, r_1) = gcd(r_1, r_2)$$
$$\vdots$$
$$gcd(r_{i-2}, r_{i-1}) = gcd(r_{i-1}, r_i)$$
$$gcd(r_{i-1}, r_i) = gcd(r_i, 0) = r_i$$

Hence, $gcd(a, b) = r_i$.■

Bézout's identity.

Consider two integers $0 < b < a$. Consider the equations of the Euclidean algorithm that gave $gcd(a, b) = r_i$. Let us rewrite all of those equations by having the remainder in the left-hand side of the equation except for the last equation.

$$r = a - b \cdot q \tag{1}$$
$$r_1 = b - r \cdot q_1 \tag{2}$$
$$r_2 = r - r_1 \cdot q_2 \tag{3}$$
$$\vdots$$
$$r_{i-2} = r_{i-4} - r_{i-3} \cdot q_{i-2} \tag{4}$$
$$r_{i-1} = r_{i-3} - r_{i-2} \cdot q_{i-1} \tag{5}$$
$$r_i = r_{i-2} - r_{i-1} \cdot q_i \tag{6}$$

Now, consider (6) which tells us $r_i = r_{i-2} - r_{i-1} \cdot q_i$ where r_i is in terms of r_{i-1} and r_{i-2}. Now, if we substitute the equation for r_{i-1} from (5) into (6), we get r_i in terms of r_{i-2} and r_{i-3}. If we then substitute the equation of r_{i-2} from (4), we get r_i in terms of r_{i-3} and r_{i-4}. So, if we keep substituting backwards, we eventually get r_i in terms of a and b. Let the coefficients then be x and y for a and b respectively. So, $ax + by = r_i = gcd(a, b)$, where $x, y \in \mathbb{Z}$. We have shown that for any two integers, their *gcd* can always be expressed as a linear combination of those two numbers.

Theorem 5: b^{-1} mod a exists if a and b are relatively prime to each other.

If a and b are relatively prime to each other, we know that $gcd(a, b) = 1$. Using Bézout's identity, we can say $a \cdot x + b \cdot y = 1$. Taking mod a on both sides, we get

$$(a \cdot x + b \cdot y) \bmod a = 1 \bmod a$$
$$\text{Or, } (b \cdot y) \bmod a = 1 \bmod a \qquad \text{Theorem 1}$$

This shows that if a and b are relatively prime to each other, then $(b \cdot y) \bmod a = 1 \bmod a$. This means that the inverse of b is y since $(b \cdot y) \bmod a = 1 \bmod a$. We have shown that $b^{-1} \bmod a$ exists if a and b are relatively prime to each other.

3 THE MERKLE–HELLMAN KNAPSACK CRYPTOSYSTEM

Knapsack review.

Let us quickly review how knapsack works. The knapsack problem has a set of n weights $W_0, W_1, ..., W_{n-1}$, and a sum S such that $S = a_0 \cdot W_0 + a_1 \cdot W_1 + \ ... \ + a_{n-1} \cdot W_{n-1}$ where $a_i \in \{0, 1\}$. For a given sum S, we need find $a_0, a_1, .., a_{n-1}$ provided that is possible. We know that solving this problem is known to be NP-complete[1].

A superincreasing knapsack is a special case knapsack such that the weights are ordered from low to high, and that a given weight $W_i > \sum_{n=0}^{i-1} W_n$. In other words, each weight is greater than sum of all previous weights. Solving this knapsack takes only $O(n)$ time complexity since for a given S, we can just start with the largest weight and move towards the smaller weights.

Now, Let us construct a knapsack cryptosystem.

1. Generate a superincreasing knapsack of a weights: $\{W_0, W_1, ..., W_{a-1}\}$
2. Pick two numbers m and n such that m and n are relatively prime to each other. From *Theorem 5*, we know that m^{-1} exists such that $(m \cdot m^{-1}) \bmod n = 1 \bmod n$.
3. Calculate m^{-1} using the euclidean algorithm.
4. Make sure that n is greater than the sum of all the weights in our super-increasing knapsack.
5. Convert this super-increasing knapsack to a general knapsack by computing $(W_i \cdot m) \bmod n$. So, our general knapsack should look like :

$$\{(W_0 \cdot m) \bmod n, (W_1 \cdot m) \bmod n, ..., (W_{a-1} \cdot m) \bmod n\}$$

6. The general knapsack becomes our public key while the super increasing knapsack along with m^{-1} becomes our private key.

Encryption.

Now, let us quickly see how encryption is done. To encrypt an integer A, we take the binary representation of A extended/shortened to a bits. Then the 1 bits are used select the elements of the general knapsack that are summed to give the sum S which is our ciphertext. For example, a cipher text can be of the kind: $S = (W_0 \cdot m) \bmod n + (W_2 \cdot m) \bmod n + (W_{a-1} \cdot m) \bmod n$. This means our message's binary representation is : $101 \ldots 001$ where only three bits are 1s.

Decryption.

To decrypt cipher text S, we would find $(S \cdot m^{-1}) \bmod n$ and solve for the private super-increasing knapsack. If we let $S = (W_0 \cdot m) \bmod n + (W_2 \cdot m) \bmod n + (W_{a-1} \cdot m) \bmod n$:

$$\begin{aligned}
(S \cdot m^{-1}) \bmod n &= ((W_0 \cdot m) \bmod n + (W_2 \cdot m) \bmod n + (W_{a-1} \cdot m) \bmod n) \cdot m^{-1}) \bmod n \\
&= ((W_0 \cdot m \cdot m^{-1}) \bmod n + (W_2 \cdot m \cdot m^{-1}) \bmod n + (W_{a-1} \cdot m \cdot m^{-1}) \bmod n)) \bmod n \quad \textit{Theorem 3} \\
&= (W_0 \bmod n + W_2 \bmod n + W_{a-1} \bmod n) \bmod n \\
&= (W_0 + W_2 + W_{a-1}) \bmod n \\
&= (W_0 + W_2 + W_{a-1}) \bmod n \\
&= W_0 + W_2 + W_{a-1}
\end{aligned}$$

$(W_0 + W_2 + W_{a-1}) \bmod n = W_0 + W_2 + W_{a-1}$ holds because, during the construction of the knapsack cryptosystem, it was required that n be greater than the sum of all the weights in the super-increasing sequence. As a result, $W_0 + W_2 + W_{a-1} < n$. So, $W_0 + W_2 + W_{a-1} \bmod n = W_0 + W_2 + W_{a-1}$.

So, this implies that, when solving the super-increasing knapsack, the bits selected during encryption using the public knapsack will correspond exactly to those identified during decryption with the private knapsack. As a result, the original message, such as $101 \ldots 001$, is successfully recovered. Therefore, decryption works as intended.

4 INSECURITY OF THE MERKLE–HELLMAN KNAPSACK CRYPTOSYSTEM

Although the Merkle–Hellman knapsack cryptosystem was an early and creative public-key cryptosystem, it was eventually shown to be insecure. Although the knapsack cryptosystem is based on an NP-complete problem, the specific way it is implemented in the Merkle–Hellman scheme allows for efficient attacks. These weaknesses have led to the cryptosystem being considered insecure for practical use.

Shamir's Attack.

Adi Shamir published a polynomial-time attack that effectively broke the original Merkle–Hellman knapsack cryptosystem[2]. The key insight is that the modular transformation does not blur the structure of the original super-increasing knapsack. So, the general knapsack that we construct from the superincreasing knapsack is actually a well-organized instance of the knapsack.[3] As a result, Shamir's attack takes advantage of the well-organized general knapsack to reconstruct an equivalent private key. This attack does not require brute-force guessing of the super-increasing sequence but instead uses the weakness in how the public key is generated. For a detailed explanation of this attack, see Shamir's original paper[2].

5 CONCLUSION

In this paper, we examined the mathematical foundations that support the Merkle–Hellman knapsack cryptosystem, including the Euclidean algorithm and Bézout's identity, which together explain why modular inverses exist when two numbers are relatively prime. These concepts are essential for understanding how the private and public keys are constructed and how decryption becomes possible. However, despite the creative mathematical foundations, this knapsack cryptosystem system has been proven insecure, most notably through Adi Shamir's polynomial-time attack in 1982.

REFERENCES

[1] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP–Completeness, W. H. Freeman & Company, 1979.
[2] A. Shamir, A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem, IEEE Transactions on Information Theory, vol. IT-30, no. 5, pp. 699–704, September 1984.
[3] M. Stamp, Information Security: Principles and Practice, Wiley- Interscience, 2005