Data protection and privacy in cost and production functions

Łukasz Matysiak
Kazimierz Wielki University
Bydgoszcz, Poland
lukmat@ukw.edu.pl

July 3, 2023

Abstract

In this paper, we consider an ongoing open problem to any enterprise, which is how to protect data and privacy in cost and production functions. We present simple and effective methods of data encryption and cost and production functions.

1 Introduction

There is a common problem of data protection and privacy issue, such as costs or factors of production, in enterprises. Of course, the data can be calculated and encrypted, but if a stranger learns the encrypted data and the given cost and production functions, it is only a matter of the adversary recreating the real data. The aim of this article is to secure the data, but also to secure the cost and production function.

The production function and the cost function are closely related, because the cost of producing a given quantity of a product depends on inputs of factors of production and their prices. The cost function TC shows what is the minimum cost of producing a given quantity of product at given input prices. The cost function can be expressed as the sum of fixed costs F and variable costs CV, which depend on the amount of production Q:

Keywords: cost function, data, encryption, privacy, production function
2010 Mathematics Subject Classification: Primary 91B24, 91B44 Secondary 94A60.

$$TC = F + CV(Q).$$

The cost function can also be determined on the basis of the production function if we know the prices of the factors of production: labor w and capital r. Then the cost function has the following form:

$$TC = wL + rK,$$

where L and K are the quantities of labor and capital needed to produce Q units of output. To find these quantities, one has to solve the cost minimization problem at a given quantity of production:

$$\min(wL + rK)$$

with the condition $Q = f(L, K)$, where $f(L, K)$ is the production function.

It is worth noting that if the fixed cost will be of the form

$$TC(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0,$$

then the average cost AC is of the form

$$AC(X) = \frac{TC(x)}{x} = a_n x^{n-1} + a_{n-1}x^{n-2} + \cdots + a_1 + \frac{a_0}{x}.$$

We can calculate the marginal cost MC as follows:

$$MC(x) = TC'(x) = na_n x^{n-1} + \cdots + 2a_2 x + a_1.$$

For simplicity, we will denote the cost and production functions as a simple mathematical function f dependent on certain variables.

Thus, the cost function will be denoted by $f(x) = ax + b$, where b is a fixed cost, a is a variable cost depending of the factor x. But nowadays such a function may not be enough, because we can change the prices in certain time periods 0, t_1, t_2, $\ldots$, t_n. Therefore, in the section 2, we will consider the general cost function:

$$f(x) = a_0 + \sum_{i=1}^{n} a_i \max(0, \min(t_i - t_{i-1}, x - t_{i-1})),$$

where the general cost function is a linear piecewise function.

In the section 3, we present the three most common types of production functions and how to encrypt them. However, if a given company used other production functions, it can be inferred from this section analogously as to encrypt a different type of function. We consider the traditional type of production function:

$$Q(x_1, \dots, x_n) = a_1x_1 + \dots a_nx_n.$$

The Cobb-Douglas production function is common:

$$Q(x_1, \dots, x_n) = a_0x_1^{a_1}x_2^{a_2} \dots x_n^{a_n}.$$

And also the previously mentioned minimum function:

$$Q(x_1, \dots, x_n) = \min(a_1x_1, \dots, a_nx_n).$$

2 Cost functions

In this section, we will discuss cost function data encryption. Encryption of cost function data is a process of transforming data containing information on costs incurred by the company in the process of production or provision of services using a secret key, so that they are incomprehensible to unauthorized persons. Encryption of cost function data is intended to ensure the confidentiality, integrity and availability of this data, and to prevent manipulation and abuse by competitors, customers, suppliers or others. Encrypting cost function data can also improve business efficiency and competitiveness by optimizing decision-making processes and resource allocation. In this section, we will present a certain encryption method for various cases, which is easy to use, and may give unauthorized people the illusion of the truth of the data, because the following method does not change the form of the cost function.

Theorem 2.1. *Let $f(x) = ax+b$ be a cost function, $a \neq 0$. Let $g(x) = cx+d$ be a linear and encryption function and let s be a square-free number relatively prime of a, c, with $a, b < s$. Then $(g \circ f) \pmod{s}$ is an encrypted function of cost and linear type.*

Proof. Let $f(x) = ax + b$, $g(x) = cx + d$ and s be defined as above. Then

$$(g \circ f)(x) = g(f(x)) \equiv g(ax + b) \equiv (ca)x + (cb + d) \pmod{s}.$$

As we can see, $(g \circ f)$ is a linear function and looks like a cost function with coded coefficients.

To get the function f back, we need to find the function $g^{-1} \pmod{s}$. In our situation it is

$$g^{-1}(x) \pmod{s} \equiv c^{-1}x - c^{-1}d \pmod{s}.$$

Calculate:

$$g^{-1}(g(f(x))) \equiv g^{-1}(cax + cb + d) \equiv c^{-1}(cax + cb + d) - c^{-1}d \equiv$$
$$\equiv ax + b \equiv f(x) \pmod{s}.$$

The calculations show that the function f can be encrypted using the function g. The new function is in the form of a cost function.

To show that the encryption is unambiguous, let's look at what f^{-1} and g^{-1} are. Of course, f is invertible if and only if $\mathrm{GCD}(a, s) = 1$ and g is invertible if and only if $\mathrm{GCD}(c, s) = 1$, which follows from assumptions. Let $y = g(f(x))$. Then $f(x) = g^{-1}(y)$, and next $x = f^{-1}(g^{-1}(y))$. That is, for every y there is exactly one x such that $g(f(x)) = y$ and can be reconstructed by applying the inverses of g and f. □

In a similar way, we can encrypt when we convert the square-free number s to the square-free polynomial $s(x)$.

Proposition 2.2. *Let $f(x) = ax+b$ be a linear cost function. Let $g(x) = cx+d$ be the encryption function and let $s(x)$ be a square-free function relatively prime of $f(x)$ i $g(x)$. Then $g(f(x)) \pmod{s(x)}$ is an encrypted function of cost and linear type.*

Proof. The proof is analogous to Theorem 2.1. □

Of course, in general, the above linear form cost function does not occur in enterprises. The cost includes many things and many time periods. In the proofs, calculations take into account the operation modulo s, where s is a square-free number.

Theorem 2.3. *Let $f(x) = b + a_1 \min(t_1, x) + a_2 max(0, x - t_1)$ be a linear piecewise cost function. Let $g(x) = c + d \min(s_1, x) + e \max(0, x - s_1)$ be an encryption function and let s be a square-free number relatively prime of a_1, a_2, d, e. Then $(g \circ f) \pmod{s}$ is an encrypted cost function and a linear piecewise function.*

Proof. Let $f(x)$, $g(x)$ and s be defined as above. Let's encrypt the function f with g modulo s:

$$(g \circ f)(x) = g(f(x)) \equiv c + d \min(s_1, b + a_1 \min(t_1, x) + a_2 \max(0, x - t_1))+$$
$$+ e \max(0, b + a_1 \min(t_1, x) + a_2 \max(0, x - t_1) - s_1).$$

As we can see the function $g \circ f$ is a linear piecewise function and looks like a cost function with encrypted coefficients.

To get the function f back, we need to find the function $g^{-1} \pmod{s}$. In our situation it is

$$g^{-1}(x) = \begin{cases} \dfrac{x-c}{d} & \text{for} \quad x - c < ds_1 \\ s_1 + \dfrac{x-c-ds_1}{e} & \text{for} \quad ds_1 < x - c \end{cases}$$

When we calculate $g^{-1} \circ (g \circ f)(x) = g^{-1}(g(f(x))) \pmod{s}$ we get the original function $f(x)$.

The proof of uniqueness is analogous to the Theorem 2.1. □

Theorem 2.4. *Let $f(x) = a_0 + \sum_{i=1}^{n} a_i \max(0, \min(t_i - t_{i-1}, x - t_{i-1}))$ be a lineat piecewise cost function. Let $g(x) = b_0 + \sum_{i=1}^{m} b_i \max(0, \min(w_i - w_{i-1}, x - w_{i-1}))$, be an encryption function and let s be a square-number, relatively prime of a_i and b_j for $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$. Then $(g \circ f) \pmod{s}$ is an encrypted cos function and linear piecewise function.*

Proof. Let $f(x)$, $g(x)$ and s be defined as above. Let's encrypt f with g modulo s:

$$(g \circ f)(x) = g(f(x)) \equiv b_0 + \sum_{i=1}^{m} b_i \max(0, \min(w_i - w_{i-1}, f(x) - w_{i-1}))$$

As we can see the function $g \circ f$ is a linear piecewise function and looks like a cost function with encrypted coefficients.

To get the function f back, we need to find the function $g^{-1} \pmod{s}$. In our situation it is:

$$g^{-1}(x) = \begin{cases} \dfrac{x-b_0}{b_1} & \text{dla} \quad x-b_0 < b_1 s_1 \\[2ex] \dfrac{x-b_0-b_1s_1}{b_2+s_1} & \text{dla} \quad b_1s_1 \leqslant x-b_0 < b_1s_1+b_2(s_2-s_1) \\[2ex] \dfrac{x-b_0-b_1s_1-b_2(s_2-s_1)}{b_3+s_2} & \text{dla} \quad b_1s_1+b_2(s_2-s_1) \leqslant x-b_0 < \\ & \quad < b_1s_1+b_2(s_2-s_1)+b_3(s_3-s_2) \\ \dots\dots\dots\dots\dots\dots\dots & \\[2ex] \dfrac{x-b_0-b_1s_1-\sum\limits_{i=2}^{n-1} b_i(s_i-s_{i-1})}{b_n+s_{n-1}} & \text{dla} \quad b_1s_1+\sum\limits_{i=2}^{n-1} b_i(s_i-s_{i-1}) \leqslant x-b_0 \end{cases}$$

When we calculate $g^{-1}(g(f(x))) \pmod{s(x)}$ we get the original function $f(x)$.

The calculations show that the f can be encrypted using the function g. The new function is in the form of a cost function. Encryption is specified correctly.

The proof of uniqueness is analogous to Theorem 2.1. □

3 Production functions

Encryption of the production function consists in securing information on how the company uses inputs of production factors: labor and capital, to achieve a certain production volume. Encryption can be applied both to the general production function and to the production function of individual products or services. Encryption is designed to protect data from unauthorized access, theft, manipulation or disclosure. In this section, we will cover a similar encryption method to the 2 section. The proofs include operations modulo s, where s is a square-free number.

Theorem 3.1. *Let $Q(x_1, \dots, x_n) = a_1x_1 + \dots a_nx_n$ be the production function. Let s be a square-free number and let $g(x) = b_1x + \dots b_mx$ be an encryption function such that $\mathrm{GCD}(b_1 + \dots + b_m, s) = 1$. Then $g \circ Q$ is an encrypted production function of the same type as Q.*

Proof. Let $Q(x_1, \ldots, x_n)$, $g(x)$, s be defined as above. Then

$$(g \circ Q)(x_1, \ldots, x_n) \equiv g(Q(x_1, \ldots, x_n)) \equiv x_1 \sum_{i=1}^{m} a_1 b_i + \cdots + x_n \sum_{i=1}^{m} a_n b_i \pmod{s}.$$

As we can see, composition is a function of the same type as Q and looks like a production function with coded coefficients.

To get the Q function back, we need to find the g^{-1} function. In our situation it is:

$$g^{-1}(x) \equiv (b_1 + \cdots + b_m)^{-1} x \pmod{s}.$$

Let's check:

$$g^{-1}(g(Q(x_1, \ldots, x_n))) \equiv g^{-1}(x_1 \sum_{i=1}^{m} a_1 b_i + \cdots + x_n \sum_{i=1}^{m} a_n b_i) \equiv$$
$$\equiv (b_1 + \cdots + b_m)^{-1}(x_1 \sum_{i=1}^{m} a_1 b_i + \cdots + x_n \sum_{i=1}^{m} a_n b_i) \equiv Q(x_1, \ldots, x_n)$$

The calculations show that the function Q can be encrypted using the function g. The new function is in the form of the production function. Encryption is specified correctly.

The uniqueness proof is analogous to the Theorem 2.1. □

The next production function is a Cobb-Douglas function.

Theorem 3.2. *Let $Q(x_1, \ldots, x_n) = a_0 x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}$ be the production function. Let $g(x) = b_0 x^{b_1} x^{b_2} \ldots x^{b_m} = b_0 x^{b_1 + \cdots + b_m}$ be the encryption function and let s be a square-free number relatively prime of a_0 and b_0. Then $g \circ Q$ is an encrypted production function of the same type as Q.*

Proof. Let $Q(x)$, $g(x)$ and s be defined as above. Then

$$g \circ Q \equiv g(Q(x_1, \ldots, x_n)) \equiv (b_0 a_0^{b_1 + \cdots + b_m})(x_1^{a_1} \ldots x_n^{a_n})^{b_1 + \cdots + b_m}.$$

As we can see, composition is a function of the same type as Q and looks like a production function with coded coefficients.

To get the function Q back, we need to find the function g^{-1}. In our situation it is:

$$g^{-1}(x) \equiv \exp\left(\log\left(\left(\frac{x}{b_0}\right)^{\frac{1}{b_1 + \cdots + b_m}}\right)\right).$$

Let's check:

$$g^{-1}(g(Q(x))) \equiv g^{-1}(b_0 a_0^{b_1+\cdots+b_m})(x_1^{a_1}\dots x_n^{a_n})^{b_1+\cdots+b_m}) \equiv Q(x)$$

The calculations show that the function Q can be encrypted using the function g. The new function is in the form of the production function. Encryption is specified correctly. The uniqueness of the encryption is proved in the same way as in Theorem 2.1. □

The next production function is also considered in enterprises.

Theorem 3.3. *Let $Q(x_1,\dots,x_n) \equiv \min(a_1x_1,\dots,a_nx_n)$ be the production function. Let $g(x) \equiv \min(b_1x,\dots,b_mx)$ be the encryption function. Then $g \circ Q$ is an encrypted production function of the same type as Q.*

Proof. Let $Q(x_1,\dots,x_n)$ and $g(x)$ be defined as above. Then

$$g(Q(x_1,\dots,x_n)) \equiv \min(b_1a_ix_i,\dots,b_ma_ix_i),$$

where a_ix_i is the actual value of the function Q. As we can see, composition is a function of the same type as Q and looks like a production function with coded coefficients.

To get the function Q back, we need to find the function g^{-1}. In our situation it is:

$$g^{-1}(x) \equiv \frac{x}{b_k},$$

for some $k \in \{1,\dots,m\}$ such that $x \leqslant \min(b_1,\dots,b_m)$.

Let's check:

$$g^{-1}(g(Q(x))) \equiv g^{-1}(\min(b_1a_ix_i,\dots,b_ma_ix_i)) \equiv \frac{\min(b_1a_ix_i,\dots,b_ma_ix_i)}{b_k} \equiv$$

$$\equiv \frac{b_ka_ix_i}{b_k} \equiv a_ix_i$$

The calculations show that the function Q can be encrypted using the function g. The new function is in the form of the production function. Encryption is specified correctly. The uniqueness of the encryption is proved in the same way as in Theorem 2.1. □

4 Some additional considerations

In all encryptions, we can use a prime number instead of a square-free number, but this increases the risk of reading our data. Often, cracking a cipher is based on factoring the key into prime factors, and such factoring is always unambiguous. However, the described square-free factorizations in [4] and [5] are usually not unambiguous, which increases the security of our data.

In the paper [3] we have an abstract description of converting a number-theoretic cipher into an algebraic one by replacing the coefficients with ideals generated by a given element. The great advantage is artificial getting rid of the finiteness of the alphabet in favor of an infinite set of alphabets. Unfortunately, the disadvantage is the very high computational complexity.

Traditional encryption uses the $\mathbb{Z}$ integer ring, which is a special case of the Dedekind ring.

Lemma 4.1. *For an integral domain R that is not a field, all of the following conditions are equivalent:*

1. *Every nonzero proper ideal factors into primes.*
2. *R is Noetherian, and the localization at each maximal ideal is a discrete valuation ring.*
3. *Every nonzero fractional ideal of R is invertible.*
4. *R is an integrally closed, Noetherian domain with Krull dimension one (that is, every nonzero prime ideal is maximal).*
5. *For any two ideals I and J in R, I is contained in J if and only if J divides I as ideals. That is, there exists an ideal H such that $I = HJ$. A commutative ring (not necessarily a domain) with unity satisfying this condition is called a containment-division ring.*

Thus a Dedekind domain is a domain that either is a field, or satisfies any one, and hence all five, of 1. through 5. Which of these conditions one takes as the definition is therefore merely a matter of taste. In practice, it is often easiest to verify 4. In [2], two cryptosystems are considered that use the fractional ideal property, where in the first system the ideal is a key, and in the second a second alphabet.

The article [1] presents a method using Galois groups. One could think of applying such encryption to our cost and production functions.

5 Declarations

This paper was not funded.

Author declares that there is no conflict of interest.

Data available within the article or its supplementary materials.

References

[1] Chrzaniuk, M., Duda, M., Hanc, M., Kowalski, S., Matysiak, Ł., Skotnicka, Z., Waldoch, M., *Certain cryptographic systems based on an algebraic structure*, East Asian Journal on Applied Mathematics, Vol. 13, No. 1, pp. 177-193.

[2] Jankowska, M., Matysiak, Ł., *A structure of Dedekind in the cryptosystem*, SCIREA Journal of Mathematics. Vol. 7, No. 1, 2022, 1-8, (2022).

[3] Matysiak, Ł., *Generalized RSA cipher and Diffie-Hellman protocol, Journal of applied mathematics & informatics*, 39 (1-2), 93-103, (2021).

[4] Matysiak, Ł., *On square-free and radical factorizations and existence of some divisors*, https://lukmat.ukw.edu.pl/files/On-square-free-and-radical-factorizations -and-existence-of-some-divisors.pdf, 2021.

[5] Matysiak, Ł., *On square-free and radical factorizations and relationships with the Jacobian conjecture*, accepted in The Asian Journal of Mathematics, (2022).