# Parallel Encryption Method for Big Data

Sena Efsun Cebeci and Enver Ozdemir
***Istanbul Technical University***

## Introduction

In the past few decades, information technology revolution affected almost all aspects of human life and emerged a growing demand of secure systems.

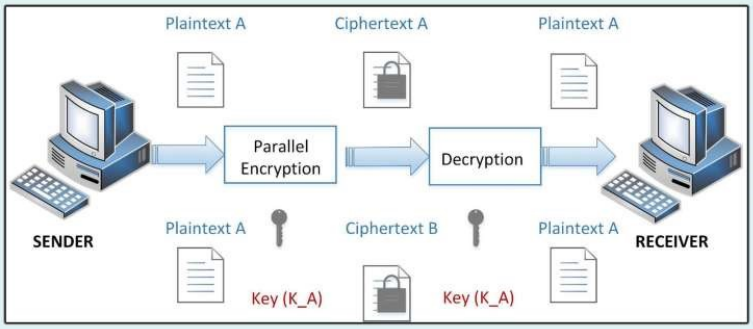
For information technology systems, providing **security via encryption** has become significant in this respect.

Due to its applicability in Big Data, in this work we mainly focus on **parallel encryption problem**.

## Motivation

- Growing demand of constructing secure systems resistant to possible attacks.
- Important data should be securely encrypted and stored.
- Parallel encryption is suitable for many applications such as Big Data and provides efficiency for the usage of multi-core processors.
- Existing encryption techniques have drawbacks:
  **Lack of parallelization**
  **Lack of robustness**

## Parallel Encryption Method




In this method, we simply encrypt the plaintext with the same key.

In each time the same letter in the plaintext corresponds to a different ciphertext.

The idea behind the encryption relies on the mathematical ground.

For example, we make use of **Mumford Representation** and **Cantor's Algorithm** and **Singular Curves** while introducing our method.

Singular curves provide benefits for parallelism and randomness.

We also use **Shamir's Secret Sharing Approach** in our encryption method.

Shamir's secret sharing approach provides high security and proven that it is information theoretically secure.

### Mumford's Representation

$H: y^2 = f(x)$ is a curve defined over a field $F$ and its degree is $2g + 1$. $Jac(H)$ is a group of $F$ and $D \in Jac(H)$ represented by unique pair of polynomials$(u(x), v(x))$, following conditions must to be satisfied:

- $u(x)$ is a monic polynomial in $F_q[x]$.
- $\deg(v(x)) < \deg(u(x)) \leq g$.
- $v(x)^2 - f(x)$ is divisible by $u(x)$.

### Cantor's Algorithm

$D_1$ and $D_2$ are two reduced divisors, $D_1 = (u_1, v_1)$ and $D_2 = (u_2, v_2)$ curve $C: y^2 + h(x)$ over the field $K$. The algorithm works as follows:

*i.* $h = \gcd(u_1, u_2, v_1 + v_2)$ with polynomials $h_1, h_2, h_3$. $h = h_1 u_1 + h_2 u_2 + h_3(v_1 + v_2)$

*ii.* $u = \frac{u_1 u_2}{h^2}$ and $v \equiv \frac{h_1 u_1 v_2 + h_2 u_2 v_1 + h_3(v_1 v_2 + f)}{h} \pmod{u}$ repeat:

*iii.* $u' = \frac{v^2 - f}{u}$ and $v' \equiv v \ (mod\ u')$

*iv.* $u = u'$ and $v = -v'$ until $\deg(u) \leq g$

### Singular Curves

$F_q$ is a finite field with characteristic $p$ where $q = p^n$.

$N$ is a singular curve defined by $y^2 = xf(x)^2$ where $f(x)$ is an irreducible polynomial of degree $d$ in $F_q[x]$.

$N$ is called Jacobian group, $Jac(N)$. An element $D \in Jac(N)$ is uniquely represented by a monic polynomial $g(x)$ of degree less than $d$.

## Parallel Encryption Algorithm

$N$ is a singular curve represented by $N: y^2 = xf(x)^2$ over $F_P$ and the key is $Key(N, F_P, m, n)$. Security parameters are m and n values.
Steps of the algorithm is as follows:

1. "a" is a given plaintext where $a \in F_P$.
2. Choose a random polynomial its degree less than $f(x)$ and embed "a" to $D = g_1(x) = x^2 + bx + a$.
3. Encrypt the plaintext multiplying the polynomial by n such that $D_2 = g_2(x) = ng_1(x)$. Assume the order of $Jac(N) = P^d - 1 = nm + (n-1)$.

## Conclusions

Propose a secure parallel encryption method for Big Data and for various different applications.

We plan to implement our method into a hardware system and explore how it affects the system in terms of security as future work.

## References

1. D. D. Mumford, Tata Lectures on Theta II, Birkhauser, 1982.
2. D. G. Cantor, Computing in the Jacobian of a Hyperelliptic Curve, Math. Comp. 48, (1987) 95-101.
3. E. Ozdemir, Curves and Their Applications to Factoring Polynomials, PhD thesis, 2009.
4. A. Shamir, How to Share a Secret. Communications of the ACM, 22(11),(1979), 612-613.