**LAB 10**

**Object: Attack on RSA encryption with short RSA modulus**

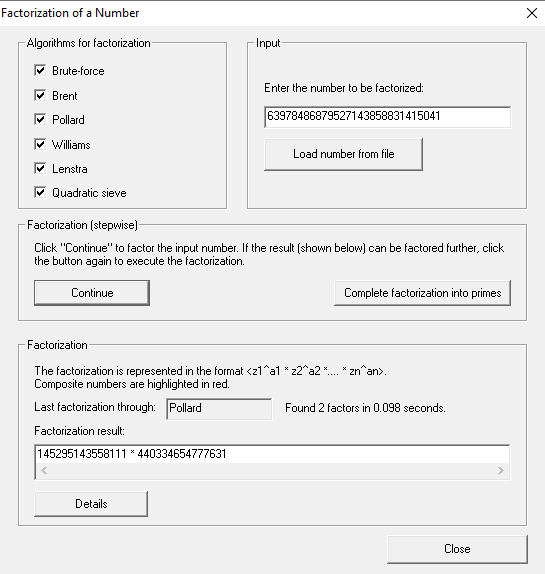
**Exercise)**

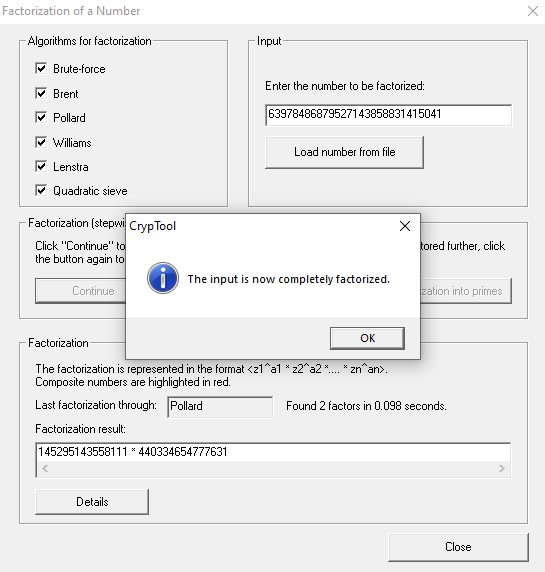
**In RSA, practical difficulty of factoring the product of two large prime numbers is known as the factoring problem. This is what RSA is based on. The prime factors must be kept secret. If the public key is large enough, only someone with knowledge of the prime factors can feasibly decode the message.**

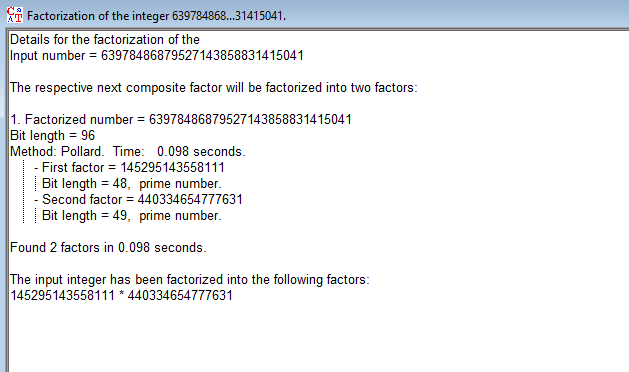
**If we know N = 63978486879527143858831415041 (95 bit, 29 decimal digits) and then try this number N = 351573870816322547022741576341143304183 (129 bit, 39 digit). Find the factors using cryptool by going in to Indiv. Procedures -> RSA Cryptosystem -> Factorization of a number. Then enter the number and find the factors. What does this tell you about the difficulty level of finding the factors in both cases? What are the factors in both cases? What algorithm was used last to factorize in both cases? Show practical demonstration.**

**Factorization of a Number:**

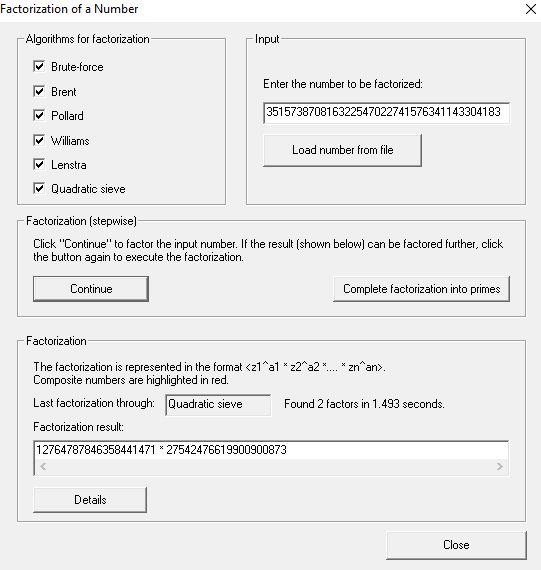
First we take n = 63978486879527143858831415041

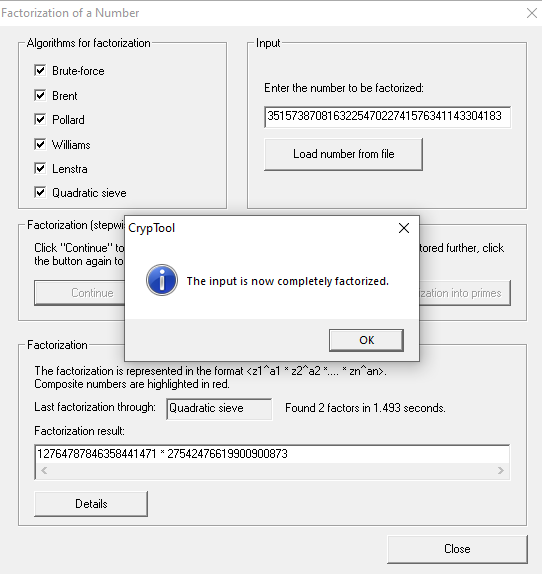
****

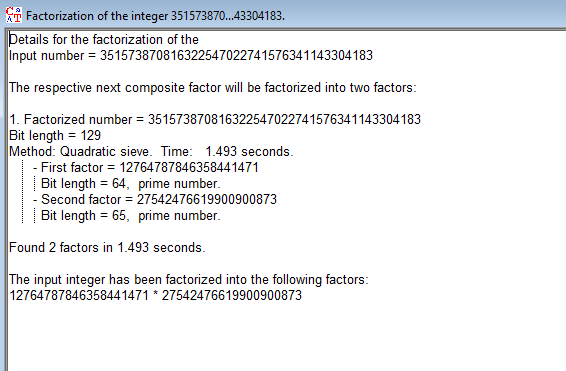
****

****

Now, we take n = 351573870816322547022741576341143304183

****

****

****

**Result of Factorization:**

**(Case 1)**

**Value of n:** 63978486879527143858831415041 (95-bit)

**Factorization Time:** 0.098 seconds

**Factors:** 145295143558111 and 440334654777631

**Last Algorithm Used to Factorize:** Pollard

**(Case 1)**

**Value of n:** 351573870816322547022741576341143304183 (129-bit)

**Factorization Time:** 1.493 seconds

**Factors:** 12764787846358441471 and 27542476619900900873

**Last Algorithm Used to Factorize:** Quadratic sieve

**Analysis:**

**Factorization Time:** The factorization time for Case 2 **(129-bit)** is notably longer **1.493** seconds compared to Case 1 (**95-bit**) which took only **0.098** seconds. This indicates an increase in the difficulty level of factorization as the size of **n** grows, consistent with the exponential nature of RSA key strength.

**Algorithm Selection:** The choice of factorization algorithm is crucial. Case 1 utilized the Pollard algorithm, which was effective for the smaller **n**. However, Case 2 required the more sophisticated Quadratic Sieve due to the increased size of **n**. The complexity of factorization algorithms is directly related to the size of the number being factored.

**Key Size and Security:** The longer factorization time and the need for a more advanced algorithm in Case 2 highlight the enhanced security provided by larger key sizes. This aligns with the standard practice of recommending larger key sizes to resist modern computational attacks.

**Practical Implications:** While the factorization times in this analysis are relatively short, in practical scenarios, these times can become significant for large key sizes. The use of Quadratic Sieve in Case 2 signifies the application of more resource-intensive methods as key sizes increase.

In conclusion, the analysis underscores the importance of considering key size in RSA for ensuring robust security. Larger key sizes contribute to increased difficulty in factorization, a critical factor in the cryptographic strength of RSA.