Key Exchange for Unified Classical Cryptographic Methods using Unsorting Techniques

A.M. Ponraj

Abstract—It is well known that a sorting algorithm is an algorithm that puts elements of a list in a certain order. We introduce a new terminology called ‘unsorting’. It is a reverse process. This process will undo all data swaps in the reverse order to get the original input list. It is interesting to note that the complicity is same as $N^2$ or $N \ln(N)$ depending on the sorting algorithm where $N$ is the number of input element. We have done unsorting using a Boolean matrix $B$ of size, say, $N^2$ which was created by answering the question of data movement as ‘YES’ or ‘NO’ while sorting by varying the $(I,j)$ in the for loop. In the sorting algorithm, instead of numbers we give character as input to the algorithm. We select $B$ as a very large matrix generated by the sorting algorithm and use a part of $B$ as the secret key for encryption and decryption. In order to achieve the variable Caesar encryption method, we shift the character in the ASCII code as a function of $(I,j)$ in the for loop. We have given the source code written in C for bubble sort, shell sort, insertion sort and selection sort. Heap sort requires additional information as parent or child warranting more than double the storage space.

Keywords—Methods using Unsorting Techniques.

I. INTRODUCTION

A CIPHER is a means of concealing a message, where letters of the message are substituted or transposed for other letters, letter pairs, and sometimes for many letters. In cryptography, a classical cipher (Ref 4-7) is a type of cipher that was used historically but now has fallen, for the most part, into disuse. In general, classical ciphers operate on an alphabet of letters (such as "A-Z"), and are implemented by hand or with simple mechanical devices. They are probably the most basic types of ciphers, which made them not very reliable, especially after new technology was developed. Modern schemes use computers or other digital technology, and operate on bits and bytes. Many classical ciphers were used by well-respected people, such as Julius Caesar and Napoleon, who created their own ciphers which were then popularly used. Many ciphers had their origins in the military and were used for transporting secret messages among people on the same side. Classical schemes are often susceptible to ciphertext-only attacks, sometimes even without knowledge of the system itself, using tools such as frequency analysis. Sometimes grouped with classical ciphers are more advanced mechanical or electro-mechanical cipher machines, such as the Enigma machine.

II. SORTING ALGORITHMS

A sorting algorithm is an algorithm that puts elements of a list in a certain order. The most-used orders are numerical order and lexicographical order (Ref 1-3). Efficient sorting is important for optimizing the use of other algorithms (such as search and merge algorithms) that require sorted lists to work correctly; it is also often useful for canonicalizing data and for producing human-readable output. More formally, the output must satisfy two conditions:

The output is in non decreasing order (each element is no smaller than the previous element according to the desired total order);

The output is a permutation, or reordering, of the input. Since the dawn of computing, the sorting problem has attracted a great deal of research, perhaps due to the complexity of solving it efficiently despite its simple, familiar statement.

Fig.1 The Design of Unification Systems using sorting algorithms

A.M. Ponraj, Department Of Computer Science Madurai Kamaraj University, Madurai – 625 021, India, Email: ponrajcomputer@gmail.com.
For example, bubble sort was analyzed as early as 1956. Bubble sort, also known as sinking sort, is a simple sorting algorithm that works by repeatedly stepping through the list to be sorted, comparing each pair of adjacent items and swapping them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted. The algorithm gets its name from the way smaller elements "bubble" to the top of the list. Because it only uses comparisons to operate on elements, it is a comparison sort. Although the algorithm is simple, it is not efficient for sorting large lists; other algorithms are better.

A. Shell Sort

Shell sort is a sorting algorithm, devised by Donald Shell in 1959, that is a generalization of insertion sort, which exploits the fact that insertion sort works efficiently on input that is already almost sorted. It improves on insertion sort by allowing the comparison and exchange of elements that are far apart. The last step of Shell sort is a plain insertion sort, but by then, the array of data is guaranteed to be almost sorted. The algorithm is an example of an algorithm that is simple to code but difficult to analyze theoretically.
III. RESULT AND DISCUSSION

It is observed that all sorting algorithms have a learning part and a decision making part. In the learning part they take an element in some ith position and compare with another element at position j in the given list Linput. The selections of position (I,J) are dictated by the given algorithm. In the decision making part they make a decision to swap the (I,J) position where I≠J. Thus this set of decision can be recorded as a sparse Boolean matrix, B(nxn). In the worst case the size of this boolean matrix is O(n^2). In this paper, for encryption, we take Linput as the message of Character and swapping as encryption of the message. Similarly, for decryption we undo (i.e unswap) the array, as a reversible process. In other words, introduce sorting process as a encryption and reverseing the sorting process as decryption. For Caesar Cipher shifting of the character, Cs, we take the shift as a function of swapped position.

Thus the Caesar Cipher shift Cs = f(I,J)mod26 for all I,J if B(I,J) = 1 where f(I,J) is based on simple arithmetic operations such as I*J, I-J, int(I/J), I+J, I*I+J etc.

We design the key B(I,J) as follows:
1. B(I,J) is strictly upper diagonal matrix.
2. The lower diagonal matrix is selected with following condition
   BB^{-1} = I
3. The sender encrypt the message of length m < n using B and f(I,J)
4. The receiver decrypt the message using B^{-1} and f(I,J)

IV. SUMMARY AND CONCLUSION

We have demonstrated the usage of sorting algorithms for encryption and decryption of a message as a unified approach for classical cryptography. Although our discussion is based on bubble sort and shell sort, one can easily extend this idea for other sorting algorithms. It is observed that heap sort needs extra information for unswap as the swaps varies from child to parent while unswapping. It is also observed that cascading of the algorithm will strengthen the keys.

REFERENCES

http://dx.doi.org/10.1145/321119.321126
http://dx.doi.org/10.1145/368370.368387
Diffie–Hellman key exchange is a method of securely exchanging cryptographic keys over a public channel and was one of the first public-key protocols as conceived by Ralph Merkle and named after Whitfield Diffie and Martin Hellman. DH is one of the earliest practical examples of public key exchange implemented within the field of cryptography. Published in 1976 by Diffie and Hellman, this is the earliest publicly known work that proposed the idea of a private key and a corresponding public key.

**Classical Key Exchange.**

- **Bootstrap problem:** how do Alice, Bob begin? Alice can’t send it to Bob in the clear! Assume trusted third party, Cathy. Alice and Cathy share secret key $k_A$; Bob and Cathy share secret key $k_B$. Use this to exchange shared key $k_s$. November 1, 2004.

---

**Simple Protocol.**

```
Alice \{ request for session key to Bob \} k_A
Cathy.

\{ ks \} k_A || \{ ks \} k_B.
```

**Contributions.**

Classical Authenticated Key Exchange. Quantum Cryptography. Classical Authenticated Key Exchange. Background. Algebraic and number theoretic background. Using cryptographic protocols, we can defend users’ passwords from phishing attacks that aim to trick the user into revealing their password to an attacker. Quantum techniques can also be applied to other cryptographic tasks, and one such task is digital cash: the no-cloning theorem which prevents quantum states from being copied naturally leads one to ask if one could create quantum money which could not be counterfeited because of the laws of nature. 1.2 Contributions.