Feasibility of using deadlocks in cryptography to resist brute-force/key-guessing attacks.

Manoj kumar[ENGS7280]

Aryabhatta College, University of Delhi, New Delhi-110021

Prof. Rajesh Sundaresan

Robert Bosch Center for Cyber-Physical System, Indian Institute of Science, Bengaluru-560012

Mentored by:

Dr. Arun Babu

Robert Bosch Centre for Cyber-Physical Systems,Indian Institute of Sciences,Bengaluru-560012

ACKNOWLEDGEMENT

It’s a great pleasure to express my gratitude and words of appreciation to the people who have been in various ways,the source of help,inspiration and encouragement during my internship.

At the outset, I avail this opportunity to express my sincere thanks to **Indian Academy of Sciences,Bengaluru** for this fellowship and heartfelt gratitude to my guide **Professor Rajesh Sundaresan**, Robert Bosch Centre for Cyber-Physical Systems, Indian Institute of Science.

I would sincerely like to thank my mentor **Dr. Arun Babu** for mentoring me during this fellowship program and guiding me at every step, without his guidance this project would not have been possible. I would also like to thank the members of Robert Bosch Center for Cyber-Physical Systems for their kind help and support.

ABSTRACT

The report presents the tasks completed during summer internship at Robert Bosch Centre for Cyber-Physical Systems. In this report we discuss about the use of deadlocks in cryptography to resist brute-force/key guessing attack.

Cryptography is the science of sending and receiving information securely. It involves the process of converting ordinary plain text into incomprehensible text and vice versa. These days cryptography also involves confidentiality, Integrity, Authentication etc.

Deadlock(In operating system) is a state in which a process depends on other process for its execution which in turn depends on some another process. Existing research is focused on detecting/avoiding deadlocks in various applications/scenarios. Currently, almost all the research in the area of deadlocks is focused on detecting and preventing deadlocks. And apparently, there are no uses of deadlocks. This research problem deals with the feasibility of using deadlocks in cryptography to resist brute-force/key-guessing attacks. This involves designing an encryption/decryption scheme which is most likely to end in a deadlock while attempting to decrypt a message with an incorrect key. However, for the correct key, the algorithm will decrypt successfully.

One important factor here is the key-guessing penalty (KGP) which is defined as the ratio of the expected time taken by an attacker to attempt decryption with an incorrect key, and the expected time taken by a genuine user to decrypt a message. The main goal of the work is to achieve high KGP. The idea is to use a cryptographic puzzle in order to achieve our goal.

Keywords: Deadlocks,Cryptographic puzzle,encryption,decryption

ABBREVIATIONS

ABBREVATIONS

	A	B
1	ARX	Add Rotate XOR
2	ASK	Asymmetric Key Encryption
3	ECC	Elliptic Curve Cryptography
4	FIPS	Federal Information Processing Standard
5	KDF	Key Derivation/Stretching Function
6	KGP	Key Guessing penality
7	KSA	Key Stretching Algorithm
8	NIST	National Institute Of Standards And Technology
9	PGP	Pretty Good Privacy
10	PKC	Public Key Cryptography
11	Pow	Proof Of Work
12	SKE	Symmetric Key Encryption
13	SHA	Secure Hashing Algorithm

1 INTRODUCTION

Cryptography is the science of securing data. It involves the conversion of a plaintext into some unreadable text for an unauthorized user, which allows it to be transmitted without unauthorized entities decoding into a readable format. Thus stopping the compromise of the data. In other words, it is a method of storing and transmitting data in a particular form so that only intended can read and process it. Cryptography is mainly based on the mathematical theory and computer science.

In earlier days, Cryptography mean only encryption/decryption, however, modern day cryptography is also concerned with confidentiality, Integrity, Non-repudiation i.e sender cannot deny his/her intention in the transmission of the information at a later stage. Authentication i.e sender and receiver can confirm each other.

The history of cryptography dates back to 2000BC with the Egyptian practice of hieroglyphics however the first known use of a modern cipher was by Julius Caesar. Cryptography is derived from the Greek word, meaning hidden.

In recent times, cryptography has gained much attention because of privacy and security concern. The ability to securely store and transfer sensitive information has proved a critical factor in success in war and business.

DeadLocks:

Non technically, deadlock is a typical situation in which no progress can be made. It is a synonym to a stalemate, stand-off etc. Technically, A deadlock is a situation in which two computer programs sharing the same resource are effectively preventing each other from accessing the resource, resulting in both programs ceasing to function.

Earlier computer operating system ran only one program at a time, all resources of the system were available to the process. Later, with the advancement in technology operating system are now capable of running multiple programs at a time facilitating the use of multicore of the system. Eventually, some operating systmem offers a dynamic allocation of resources. Programs could request further allocations of resources after they had begun running. This leads to the problem of deadlocks.

Example of Deadlock like situation:

Program 1 requests resource A and receives it.

Program 2 requests resource B and receives it.

Program 1 requests resource B and is queued up, pending the release of B.

Program 2 requests resource A and is queued up, pending the release of A.

Necessary and sufficient conditions for deadlocks:

1. Mutual exclusion: The resources involved must be unshareable; otherwise, the processes would not be prevented from using the resource when necessary.

2. Hold and wait or partial allocation: The processes must hold the resources they have already been allocated while waiting for other (requested) resources. If the process had to release its resources when a new resource or resources were requested, a deadlock could not occur because the process would not prevent others from using resources that it controlled.

3.No pre-emption: The processes must not have resources taken away while that resource is being used. Otherwise, deadlock could not occur since the operating system could simply take enough resources from running processes to enable any process to finish.

4.Resource waiting or circular wait: A circular chain of processes, with each process holding resources which are currently being requested by the next process in the chain, cannot exist. If it does, the cycle theorem (which states that "a cycle in the resource graph is necessary for a deadlock to occur") indicated that deadlock could occur.

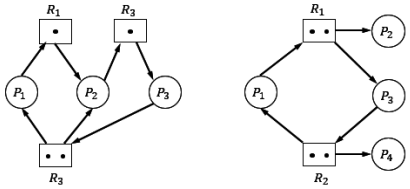

Fig 1 Deadlock

In the above figure, the left side process, there is a deadlock because process p1 and p2 are allocated instances of resources R3 and are part of the same cycle, p1-R1-p2-R3-p3-R3-p1

However right-hand side figure does not have deadlock because

1. Process p2 and p3 are allocated instances of R1 but they are not part of any cycle

2. Similarly, process p1 and p4 are allocated instances of R2, but they both are not part of any cycle.

2 Background/Rationale of the project

Currently, almost all the research in the area of deadlocks in various applications is focused on detecting and preventing deadlocks. And apparently, there are no uses of deadlocks.

The rationale behind this project is to use deadlocks to use deadlocks in cryptography to resist brute-force /key guessing attacks. This involves designing an encryption/decryption scheme which is most likely to end in deadlocks while attempting to decrypt the message using a wrong key. However, for the correct key, the algorithm will decrypt successfully.

One important factor is the key-guessing penality(KGP) which is defined as the ratio of the expected time taken by an attacker to attempt decryption with an incorrect key, and the expected time taken by a genuine user to decrypt a message. The main goal of the work is to achieve high KGP.

The importance of using deadlocks in this problem is that there is no any general solution to avoid deadlocks. However some probably deadlock handling includes:

1. Ignoring deadlocks

2. Detection of the deadlocks and doing process termination or resource preemption.

3. Prevention of the deadlocks.

However none of the methods guarantees that deadlocks will not occur. This very fact gives an advantage to our research problem.

3 Statement of the Problems

This research problem deals with the feasibility of using deadlocks in cryptography to resist brute-force/key-guessing attacks. This involves designing an encryption/decryption scheme which is most likely to end in deadlock while attempting to decrypt a message with an incorrect key. However, for the correct key, the algorithm will decrypt successfully.

4 Objectives of the Research

This research is all about developing a deadlock based encryption/decryption algorithm which results in a severe deadlock if tried to decrypt with the wrong key however with the right key it decrypts easily. The research has been completed in the following steps:

1. Literature review on various necessary and sufficient conditions that causes deadlocks.

2. Literature review on various cryptographic puzzles.

3. Study of the feasibility of using deadlocks in ciphers.

4. Implementation /theoretical analysis of the technique(s).

4.1 Scope

The research aims to investigate the possibility of using deadlocks in cryptography to resist brute force attacks. Techniques such as using timmer to delay between incorrect key/password attempts, multi-factor authentication and CAPTCHA(Completely Automated Public Turing test to tell Computers and Humans Apart) are being used to resist brute-force attacks over the network. However, such techniques cannot be used if ciphertext is available with the adversary(offline brute-force attack).

Thus In this paper, we deal with the offline attack and our ideal case scenario is that ciphertext is available with the adversary. For instance, encrypted data gathered from a wireless channel, or lost/stolen encrypted files/disks. To resist offline brute-force attacks, generally key-stretching and slower algorithms are preferred. These techniques are better but for lower powered devices it is less effective. And also it takes same time in case of both real and fake users.

Thus using deadlocks in cryptography will add an advantage here in terms of time taken by real users are fake users. In case of the real user the algorithm will work normally but in case of fake users, the system will result in a deadlock. Thus, preventing the brute-force attack.

We limit our scope to offline attacks where the attacker has access to the ciphertext.

Basic Assumption:

1. The system uses a good hash function to generate the stretched key.

2. The proposed approach assumes the authenticity of the other cryptographic keys.

5 Literature review

In this chapter, a brief study is presented which has been done in order to have an idea of project elements. In the literature survey, we found that a lot of work has been done on deadlocks, ways of avoiding deadlocks, different ways of handling deadlocks etc. Similarly, cryptography is much-explored area. Different types of research have been done in this area concerning security and privacy. Different kinds of puzzles for different cases are available in the literature. However, the very less or no work is available in the literature on using deadlocks for any useful purpose. Thus taking help of the already available research on deadlocks and cryptographic puzzles. We combine these two concepts to develop a new encryption/decryption technique.

6 Concepts

Cryptographic puzzles:

Cryptographic puzzles are difficult problems that can be solved by investing non-trivial amounts of computation and /or memory[1].Rivest et. all put forward the use of cryptographic puzzle which is used to enable future decrytption[2]. Most of the puzzle-related literature concentrates on providing construction, often with additional, innovative properties. For example, puzzles that are non-parallelizable prevent an adversary from using distributed computations to solve them. Cryptographic puzzles also play a greater role in the verification and validation of data. Some of the cryptographic puzzles includes:

1. Time lock puzzle

2.Memory bound puzzle

3. CPU bound puzzle

4. Network Bound Puzzle

We use that aspect of cryptographic which upon solving gives the required output. Thus, anyone needs to solve this puzzle before progressing further to the solution.

Livelocks:

It is similar to a deadlock except that the states of the process in the livelock constantly changes with regard to one another none progressing. The main difference between deadlocks and livelock is that threads are not going to be blocked instead they will try to respond to each other continuously.

Proof of work:

It's an issue that requires some computational power to solve. The work must be moderately hard but feasibly on the on the requester side. This is also known as

CPU cost function. It is distinct from CAPTCHA, which is intended for a human to solve quickly, rather than a computer. Some of the Pow functions include:

1. Weaken Fiat-Shamir Signatures

2. Partial hash inversion

3 Hash sequences

4. Puzzles

Hashing:

It is a technique in which data is converted into a fixed length string which can never be recovered back.

Symmetric Key Algorithm:

It is an algorithm for cryptography that uses the same cryptographic keys for encryption of plain text and decryption of ciphertext.The keys may be identical or there may be a simple transformation to go between the two keys.

Asymmetric Key Cryptography:

It is also known as public key cryptography. In this scheme, two keys are used named public and private keys. The keys are simple large numbers that have been paired together but are not identical. Either of the keys can be used to encrypt a message; the opposite key from the one used to encrypt the message is used for decryption.

Kerckhoff's Principle[1883]

It states that a cryptosystem should be secure even if the attacker i.e eve knows all the details of the system with the exception of secret key.

Salt vs key stretching/Strengthening

Salt is a noise that is added to make hash more secure The key is equivalent to the password.

Result = salt + encrypt(salt + encrypt(salt + encrypt(message)))

SHA

The secure hashing algorithms are a family of cryptograph i.e hash function published by National Institute of Standard and Technology(NIST) as a U.S Federal Information Processing Standard(FIPS). SHA2 consists of 6 hashing algorithm and is considered strongest. SHA 256 produces 23-byte hash value.

Encryption Vs Hashing

Hashing is a string or a number generated from a string of text. The resulting string or a number is a fixed length and will widely vary with small variations in input. While encryption turns data into a series of unreadable characters that aren't of a fixed length. Encrypted strings can be reversed back into original form if we have the right key. However, it is not possible with the hashing.

Cryptanalysis

It is the science of analyzing and breaking secure information/ communication. It uses mathematical formulas to search for the algorithm vulnerabilities and break into cryptography or information security system. Cryptanalysis attack types include:

1. Known Plaintext Analysis(KPA): Attacker decrypt ciphertexts with known partial plaintext.

2. Chosen-plaintext Analysis(CPA): Attacker uses ciphertext that matches arbitrary selected plaintext via the same algorithm technique.

3. Ciphertext-Only Analysis(COA): Attacker uses known ciphertext collections.

6.1 INFORMATION

Hashing and Encryption are the two buzzwords in cryptography. Hashing refers to a mapping between an arbitrarily length input, and a fixed length output. It can be anything from a simple crc32 to a full-blown cryptographic hash function such as MD5 or SHA1/2/256 etc. A good hash has a property that it can't be reversed. The reason why it is so is that most cryptographic hash functions iterate over the input set many a time to produce the output.

However, Encryption function provides a 1:1 mapping between an arbitrary length and output. And they are always reversible using some methods. It is always 1:1 for a given key. A well encrypted data is indistinguishable from random noise. This is different from a good hash output which is always of a consistent format.

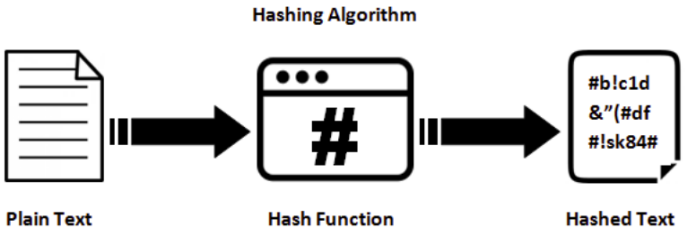

Fig 2 Hashing Function

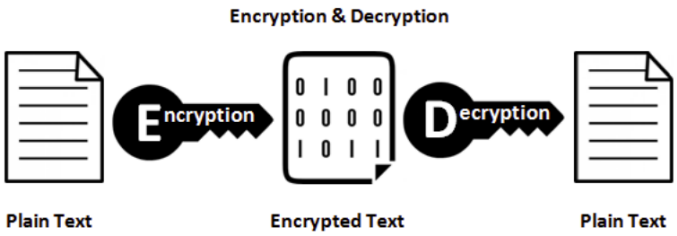

Fig 3 Encryption/Decryption Function

Computers intended for the time-sharing and /or real-time markets are often equipped with hardware lock(hard lock) which guarantees exclusive access to process force serialization. Brute force attacks mean trying too many passwords or passphrases with the hope of eventually guessing correctly. However, the strength of a password is a function of length, complexity, and unpredictability.

7 Methodology

Cryptography deals with the numbers and strings. Basically, every digital thing in the entire universe are numbers. Thus all the calculations are performed on the numbers.

The stated scheme involves three parties:

Alice, the sender of the information. Bob, receiver of the information and Eve, who is an eavesdropper.

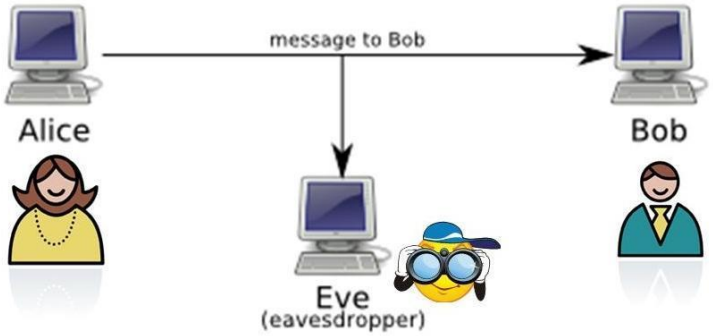

Fig 4 Information transmission

For encryption,

Alice enters his password which consists of a mixture of numbers, strings, and special characters. Let's call this password as K. This password is now stretched to 256 bits using KSA(Key Stretching Algorithm).

So, K ->K1 Here K1 is stretched key.

with the combination of K and K1 we generate another key K2, which is used for encryption of the plain text.

Thus, K+ K1 -> K2 Here K2 is key used for encrypting the plain text.

Now using K1, which is a stretched key. A puzzle is generated which needs to be solved for decryption of the ciphertext.

K1 -> Cryptographic Puzzle

This Cryptographic puzzle is sent to the person who wants to decrypt the ciphertext. In addition to it, Real User already knows the password i.e K.

For Decryption,

Bob, the person who wants to decrypt the ciphertext using right key already know the K(password which was used to encrypt the palintext).

Now Using K again a 256 bits hash is generated, which will be same as the K1.

Thus again, K -> K1

This K1 is used to solve the puzzle. If the puzzle is solved then again K and K1 are combined to generate a new key which is same as K2. Thus K2 is used to decrypt the ciphertext. This approach is similar to symmetric key cryptography(SKC).

7.1 Generation of Cryptographic Puzzle

There are several types of cryptographic puzzles available in the literature. We used a specific version of the cryptographic puzzle. Our problem required such puzzles which upon giving a wrong key ends in a deadlock.

For Cryptographic puzzle we propose the following approach:

7.2 Blocks Based approach

Puzzle creation:

In this approach, we divide the stretched key k1 into blocks i.e we have 256 bits key.

Thus forming each block of 16 bits, we will have 16 blocks in total.

Now picking any two random blocks, one bit in each block is swapped with one another.

We store the swap sequence in an array. Thus after random swapping, a hash is generated which is again of 256 bits. In addition to this, a hash of array, which stores the sequence is also generated.

Both of these things is sent to the person who wants to decrypt the ciphertext.

At decryption end:

The user enters the key K which is stretched to K1. Now, again the 256 bits generated key is divided into 16 blocks containing 16 bits each. With the help of sequence array, the bits or the thread is swapped. In the meantime it matches with the available hash value for the random bits if the value matches then the process proceeds otherwise it performs a more complicated task, thus wasting time and resources of the system. Thus for the right user bits will keep on matching with the available hash value. While for the person with the wrong key. It will take more, thus applying key guessing penalty. It uses the mutex to synchronize the thread. Thus causing the right swap.

Thus, the time taken by person with wrong key/ time taken by person with the wright key >1

Limitations with this approach:

If the system matches the bits at the beginning or upto a certain length. Then attacker will get a chance to try next key. If it repeats for a larger number of time, Then it will be easier for the attacker to try a large number of key. In that situation, this approach will fail to do the required.

7.3 Graph Based approach

For the puzzle creation,

This approach uses the graph to generate the puzzle. In this approach, We use some random function.

From 256 bits stretched key. Some function is added and is completely randomized. From this completely randomized set of bits. A random graph is constructed which contains all 256 vertices containing weight as 0 and 1.

Now remove all the link containing 0. Keeping in mind that it does not form a loop anywhere in the graph.

We send hash of this, graph as puzzle to those who want to decrypt the ciphertext.

At Decryption End:

While solving the puzzle, the password K is stretched to K1 and again the bits in the stretched key is shuffled. Now a random graph is generated and all the edges containing 0 is removed. If the graph is acyclic then it is hashed and it its value is matched with the puzzle hash value. Otherwise again the random graph is generated and 0 is removed and checked.

Limitations of this approach:

It may be possible that if hacker enters the wrong key and if it contains the same number of 0 and 1 as the real one. Then it will take the same amount of time for both real and fake user to decrypt the ciphertext.

7.4 Thread- Resource based approach.

Puzzle creation

In this approach, there are threads and some resources. Resources are some functions. The threads acquire any random resources and are executed. Mutex determine of order of the thread executed. The function stores a random value for each thread. Now after the execution of each thread. A hash is generated. Thus here the hash and the function is sent to the user who wants to decrypt the ciphertext.

At the decryption end.

It follows the same order, when any thread enters the resources it performs the simple calculation if the stored value matches with the produced value otherwise it performs the complex calculation. This is done for every thread and at the end. A hash is generated and is matched with the sent hash.

Limitation of this approach:

Sometimes the right key also leads to heavy calculation, if the produced value does not match with the stored value. Thus some time it may take equal time for both the real user and the fake user.

8 Results and discussion

From the above discussion, we can draw an inference that although it is possible to use deadlocks in cryptography but problem lies with offline attack which becomes the weakest point in the system. However, methods suggested above works only in limited environment conditions which may not work in all cases which were the need of this research. Tightening some parameter leads to incohesive deadlocks which violets the basic principle of the deadlocks. Thus finding the most efficient way which satisfies all the conditions of the problem remains the thing to explore.

9 Conclusion

With the advancement in technology security and privacy has a greater role to play. Using deadlocks in cryptography can be a new area which may improve the security of the system at a greater level.

In this paper, we have tried to develop a strong relationship between deadlocks and cryptography and using deadlocks in cryptography to create encryption and decryption technique which could resist brute-force attack.

We have introduced the concept of key-guessing penality (KGP) which is helpful in measuring the time taken by the real user in comparison to time taken by a fake user while decryption of the ciphertext.

However, there is further scope for research on the other possible and simpler ways to achieve higher KGP, and study the properties of such ciphers.

References:

[1] Groza, B. & Warinschi, B. Des. Codes Cryptogr. (2014) 73: 177. https://doi.org/10.1007/s10623-013-9816-5

[2] Sruthy, R. S. (2014). A Secure Cryptographic Puzzle based Approach Ensuring Total Security for transmitted Information with IP Tracing. *IOSR. J. Comput. Eng*, *6*, 27-32.Sruthy, R. S. (2014). A Secure Cryptographic Puzzle based Approach Ensuring Total Security for transmitted Information with IP Tracing. *IOSR. J. Comput. Eng*, *6*, 27-32.

3.Operating system(deadlocks)

https://en.wikibooks.org/wiki/Operating_System_Design/Concurrency/Deadlock

4. Password strength

https://en.wikipedia.org/wiki/Password_strength

5. Password cracking

https://en.wikipedia.org/wiki/Password_cracking

6. Various test to crack password

https://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher

7. Hashing vs encryption

https://stackoverflow.com/questions/4948322/fundamental-difference-between-hashing-and-encryption-algorithms

8. Cryptographic puzzles

https://security.stackexchange.com/questions/18466/what-is-a-cryptographic-puzzle

9. Proof-of-work(Cryptographic puzzles)

https://en.wikipedia.org/wiki/Proof-of-work_system

10. Key Derivation/Stretching Function(KDF)

https://en.wikipedia.org/wiki/Key_derivation_function

11. Dining philosophers problem

https://en.wikipedia.org/wiki/Dining_philosophers_problem

12. Linear blocks code

https://en.wikipedia.org/wiki/Linear_code

13. Cryptographic puzzles i.e Cryptographic puzzles and Dos resilience, revisited

https://link.springer.com/content/pdf/10.1007%2Fs10623-013-9816-5.pdf

14.Elliptic Curve cryptography

https://en.wikipedia.org/wiki/Elliptic-curve_cryptography

15. Babu, P. Arun, and Jithin Jose Thomas. "Freestyle, a randomized version of ChaCha for resisting offline brute-force and dictionary attacks." arXiv preprint arXiv:1802.03201 (2018).

FURTHER READING:

1. Introduction to Cryptography

https://www.akadia.com/download/documents/intro_to_crypto.pdf

2. Crypto 101 (lvh)

https://www.crypto101.io/