CRYPTANALYSIS OF RSA THROUGH FORMAL VERIFICATION TOOLS

[1]SACHIN UPADHYAY , [2]YASPAL SINGH

[1]Department of Mathematical Sciences and Computer Applications, Bundelkhand University,
Department of Computer Science & Engineering, Bundelkhand Institute of Engineering & Technology,
Jhansi, India
[2]Department of Computer Science & Engineering, Bundelkhand Institute of Engineering & Technology,
Jhansi, India

E-mail: [1]sachinupadhyay2010@yahoo.co.in, [2] yash_biet@yahoo.co.in

ABSTRACT

In this paper we are basically analyzing the results given by Wiener's, who says that if the private exponent d used in RSA cryptosystem is less than n^.292 than the system is insecure. We will focus on the result given by Weiner's and try to increase the range of private exponent d up to n^0.5. As n is the product of p & q (which are the relative prime numbers). So our emphasis is how we can make our system secure(based on the breaking of keys), One solution can be by choosing a big prime numbers as p & q further the efficient use of RSA algorithm will help to make system secure.

Keywords - *Cryptography, Cryptanalysis, Conjunctive Normal Form (CNF), Decryption, Encryption, RSA Algorithm, SAT Solver Tool.*

1. INTRODUCTION

Cryptology is the study of secret codes. In speaking of cryptology, we discuss two main branches: cryptography is concerned with the writing of messages in secret code and the creation of these methods, while cryptanalysis is concerned with reading encrypted messages by breaking secret codes. There are two types of cryptography: secret key and public key cryptography .In secret key same key is used for both encryption and decryption. In public key cryptography each user has a public key and a private key. Here we are working on RSA (Rivest, Shamir and Adleman) is the most widely used public key cryptosystem. It may be used to provide both secrecy and digital signatures and its security is based on the intractability of the integer factorization. We will convert the modular equations of RSA into CNF (conjunctive normal form) form to apply as an input on SAT Solver.

2. CRYPTOGRAPHY

It is the study of Secret (crypto)-Writing (graph).It is the science or art of encompassing the principles and methods of transforming an intelligible message into one that is intelligible and then transforming the message back to its original form.

Cryptographic systems are characterized along three independent dimensions.

A. *Operation used for Encryption*

All encryption algorithms are based on two general principles: substitution, in which each element in the plaintext is mapped into another element and transposition, in which elements in plaintext are rearranged.

B. *The number of keys used*

If both sender and receiver use the same key, the system is referred to as symmetric, single-key, secret-key, or conventional encryption. On other way the system is referred to as asymmetric, two-keys, or public-key encryption.

C. *Processing of plaintext*

A block cipher processes the input one block of elements at a time, producing an output block for each input block. A stream cipher processes the input elements continuously, producing output one elements at a time, as it goes along.

3. METHODOLOGY

Here we are basically working on the RSA Algorithm. We will convert the RSA problem in sat-satisfibility condition and solve them using sat solver. So far only sat condition are used to invert hash function in cryptography we are first time using in this approach to attack a different types of cryptosystem.

A. RSA Algorithm

The RSA scheme is a block cipher in which the plaintext & ciphertext are integers between 0 and n-1 for some n. A typical size for n is 1024 bits, i.e., n is less than 2^{1024}. Units

Select p, q p and q both prime, $p \neq q$
Calculate $n = p \times q$

Encryption
Plain text $M < n$
Cipher text $C = M^e \bmod n$ (1)

Decryption
Cipher text C
Plain text $M = C^d \bmod n$ (2)

Where e is public key
d is private key

So now we need to convert the RSA problem in sat satisfibility condition and then use it as an input for the sat solver.

B. Attacks on RSA

Timing attack: Timing attacks are applicable not just to RSA, but to other public-key cryptography systems. This attack is alarming for two reasons: It comes from a completely unexpected direction and it is a cipher text only attack.

Brute force: The attacker tries every possible key on a piece of cipher text until an intelligible translation into plaintext is obtained. This involves trying all possible private keys
.

C. Classical encryption techniques

The technique enables us to illustrate the basic approaches to conventional encryption today. The two basic components of classical ciphers are substitution and transposition. Then other systems described that combines both substitution and transposition are discussed below

1. Substitution techniques

In this technique letters of plaintext are replaced by numbers or by symbols. If plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with cipher text bit patterns.

a) Caesar Cipher

Caesar Cipher replaces each letter of the message by a fixed letter a fixed distance away e.g. uses the third letter on and repeatedly used by Julius Caesar.

For example:

Plaintext:
The quick brown fox jumps over the lazy dog

Ciphertext:
WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ

b) Monoalphabetic Ciphers

With only 25 possible keys, the Caesar cipher is far from secure. A dramatic increase in the key space can be achieved by allowing an arbitrary

Example:

Plain Text:	a	b	c	d ….
Cipher Text:	D	E	F	G ….

Here a single cipher alphabet is used per message. If, instead the cipher line can be any permutation of the 26 alphabetic characters, then there are 26! or greater than $4x10^{26}$ possible keys, this would seems to eliminate brute-force attack techniques for cryptanalysis.

2. Transposition Techniques

In this technique a different kind of mapping is achieved by performing some sort of permutation on the plaintext letters. The simplest use of transposition is rail fence technique in which the plaintext is written down as a sequence of diagonals and then read off as a sequence of rows.

Cryptanalysis the objective of attacking an encryption system is to recover the key in use rather than simply to recover the plaintext of a single cipher text. There are two general approaches to attacking a conventional encryption scheme:

1. Cryptanalysis

Cryptanalytic attacks rely on the nature of the algorithms plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext-ciphertext pairs. This type of attack exploits the characteristics of the algorithms to attempt to deduce a specific plaintext or to deduce the key being used.

2. *Brute-force attack*

The attacker tries every possible key on a piece of cipher text until an intelligle translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success.

4. SAT SOLVER TOOLS

Mostly SAT solvers are based on the Davis-Putnam-Logemann-Loveland (DPLL) algorithm and require the input formula to be in Conjunctive Normal Form (CNF). However, typical formulas that arise in practice are non-clausal, that is, not in CNF. Converting a general formula to CNF introduces overhead in the form of new variables and may destroy the structure of the initial formula, which can be useful to check satisfiability efficiently.

Boolean satisfiability (SAT) solvers are used heavily in verification tools as decision procedures for propositional logic.

In complexity theory, the satisfiability problem (SAT) is a decision problem, whose instance is a Boolean expression written using only AND, OR, NOT, variables, and parentheses. A formula of propositional logic is said to be satisfeable if logical values can be assigned to its variables in a way that makes the formula true.

5. CONJUNCTIVE NORMAL FORM

In Boolean logic, a formula is in conjunctive normal form (CNF) if it is a conjunction of clauses, where a clause is a disjunction of literals, where a literal and its complement cannot appear in the same clause. As a normal form, it is useful in automated theorem proving. It is similar to the product of sums form used in circuit theory.

All conjunctions of literals and all disjunctions of literals are in CNF, as they can be seen as conjunctions of one-literal clauses and conjunctions of a single clause, respectively. As in the disjunctive normal form (DNF), the only propositional connectives a formula in CNF can contain are and, or, and not. The not operator can only be used as part of a literal, which means that it can only precede a propositional variable.

6. CONVERSION INTO CNF

Every propositional formula can be converted into an equivalent formula that is in CNF. This transformation is based on rules about logical equivalences: the double negative law, De Morgan's laws, and the distributive law.

The generated formula is:

$$(X_1 \vee \cdots \vee X_{n-1} \vee X_n) \wedge (X_1 \vee \cdots \vee X_{n-1} \vee Y_n) \wedge \cdots \wedge (Y_1 \vee \cdots \vee Y_{n-1} \vee Y_n).$$

This formula contains 2^n clauses; each clause contains either X_i or Y_i for each i.

These transformations are guaranteed to only linearly increase the size of the formula, but introduce new variables. For example, the above formula can be transformed into CNF by adding variables

$Z_1, \ldots, Z_n$ as follows:

$$(Z_1 \vee \cdots \vee Z_n) \wedge (\neg Z_1 \vee X_1) \wedge (\neg Z_1 \vee Y_1) \wedge \cdots \wedge (\neg Z_n \vee X_n) \wedge (\neg Z_n \vee Y_n).$$

An interpretation satisfies this formula only if at least one of the new variables is true. If this variable is Z_i, then both X_i and Y_i are true as well. This means that every model that satisfies this formula also satisfies the original one.

7. PROBLEM STATEMENT

The aim of the paper is that is to eventually increase the range of private exponent d used in RSA cryptosystem up to N^0.5, if d < N^0.5 the system is insecure. Basically we are challenging the work done by DAN BONEH & GLENN DURFEE they proved that if d is less than N^.292 that the system is insecure, this is the first improvement over an old result of Wiener showing that when d is less than N^.25 the RSA system is insecure.

8. PUBLIC-KEY ALGORITHMS

This type of algorithms are symmetric, that is to say the key that is used to encrypt the message is different from the key used to decrypt the message.

The encryption key, known as the Public key is used to encrypt a message, but the message can only be decoded by the person that has the decryption key, known as the private key. This type of encryption has a number of advantages over traditional symmetric Ciphers. It means that the recipient can make their public key widely available anyone wanting to send them a message uses the algorithm and the recipient's public key to do so, only the recipient, with the private key can decrypt the message.

RSA Cryptosystem

The RSA scheme makes use of an expression with exponentials. Plaintext is encrypted in blocks, with each block having a binary value less than some number n. That is, the block size must be less than or equal to log2 (n); in practice the block size is I bits, Encryption and decryption are of the following form, for some plaintext block M and cipher text block C:

$$C = M^e \bmod n \qquad (3)$$

$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n \qquad (4)$$

The private key consists of {d , n} and the public key consists of {e , n} . Suppose that user A has published its public key and that user B wishes to send the message M to A. then B calculates C = M^e (mod n) and transmits C.
On receipt of this cipher text, user A decrypts by calculating M = C^d (mod n)

Example of attack on RSA

Here we will group the characters into blocks of three and compute a message representative integer for each block.

ATTACK*AT*SEVEN= ATT ACK *AT
*SEVEN

In the same way that a decimal number can be represented as the sum of powers of ten,
e.g. $135 = 1 \times 10^2 + 3 \times 10^1 + 5$

We could represent our blocks of three characters in base 26 using A=0, B=1, C=2, ..., Z=25

$$ATT = 0 \times 26^2 + 19 \times 26^1 + 19 = 513$$
$$ACK = 0 \times 26^2 + 2 \times 26^1 + 10 = 62$$
$$XAT = 23 \times 26^2 + 0 \times 26^1 + 19 = 15567$$
$$XSE = 23 \times 26^2 + 18 \times 26^1 + 4 = 16020$$
$$VEN = 21 \times 26^2 + 4 \times 26^1 + 13 = 14313$$

For this example, to keep things simple, we'll not worry about numbers and punctuation characters, or what happens with groups AAA or AAB.

In this system of encoding, the maximum value of a group (ZZZ) would be 26^3-1 = 17575, so we require a modulus n greater than this value.

1. We "generate" primes p=137 and q=131
2. n = p.q = 137.131 = 17947
   phi = (p-1)(q-1) = 136.130 = 17680
3. Select e = 3
   check gcd(e, p-1) = gcd(3, 136) = 1, and
   check gcd(e, q-1) = gcd(3, 130) = 1
4. Compute d = e^{-1} mod phi = 3^{-1} mod 17680
   = 11787.
5. Hence Public Key, (n, e) = (17947, 3)
   Private Key (n, d) = (17947, 11787)

To encrypt the first integer that represents "ATT", we have c = m^e mod n = 513^3 mod 17947 = 8363.

We can verify that our private key is valid by decrypting
m' = c^d mod n = 8363^{11787} mod 17947 = 513.

Overall, our plaintext is represented by the set of integer m

(513, 62, 15567, 16020, 14313)

We compute corresponding cipher text integers c = m^e mod n,

(8363, 5017, 11884, 9546, 13366)

The security of RSA algorithm depends on the ability of the hacker to factorize numbers. New, faster and better methods for factoring numbers are constantly being devised. The Trent best for long numbers is the Number Field Sieve. Prime Numbers of a length that was unimaginable a mere decade ago are now factored easily. Obviously the longer a number is, the harder is to factor, and so the better the security of RSA.

9. RESULTS

As we have already mentioned that we are increasing the range of private exponent, in this paper we give the substantial improvement to Wiener's result. Our results are based on the seminal work of Coppersmith Wiener describes a number of clever techniques for avoiding his attack while still providing fast RSA signature generation. One such suggestion is to use a large value of e. Indeed, Wiener's attack provides no information as soon as e > N^1.5. In contrast, our approach is effective as long as e < N^1.875. Consequently, larger values of e must be used to defeat the attack. Results are successfully tested for some of the sets of prime numbers like 5 & 7, 17 & 11, 137 & 131. etc. It’s been basically depends on the fact that you need to be very choosy in prime numbers sets.

REFERENCES

[1]. Eli Biham and Rafi Chen. Near-collisions of SHA-0. In Matthew K. Franklin, editor,Advances in Cryptology—CRYPTO 2004, volume 3152 of Lecture Notes in Computer Science, pages 290–305. Springer, 2004.

[2]. M. Ajtai and C. Dwork, A public-key cryptosystem with worst-case/average-case equivalence, *Proc. 29th ACM STOC* (1997), 284-297 [3].
Eli Biham, Rafi Chen, Antoine Joux, Patrick Carribault, Christophe Lemuet, and William Jalby. Collisions of SHA-0 and reduced SHA-1. In Cramer [Cra05], pages 36–57.

[4]. Gilles Brassard, editor. Advances in Cryptology—CRYPTO ’89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20–24, 1989,Proceedings, volume 435 of Lecture Notes in Computer Science. Springer, 1990.

[5]. L.M. Adleman, *Statement*, Cryptographer's Expert Panel, RSA Data Security Conference, San Francisco, CA, January 17, 1996.

[6]. Ronald Cramer, editor. Advances in Cryptology—EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22–26, 2005, Proceedings, volume 3494 of Lecture Notes in Computer Science. Springer, 2005.

[7]. Ivan Damg°ard. Collision free hash functions and public key signature schemes. In David Chaum and Wyn L. Price, editors, Advances in Cryptology—EUROCRYPT ’87, volume 304 of Lecture Notes in Computer Science, pages 203–216. Springer, 1988.

[8]. Ivan Damgoard. A design principle for hash functions. In Brassard [Bra90], pages 416–427.

[9]. Magnus Daum. Cryptanalysis of Hash Functions of the MD4-Famiy. PhD thesis, Ruhr-Universit¨at Bochum, 2005.