Hybrid Cryptosystem Based on 2-SAT & 3-SAT and the Implications of Polynomial Solvability of 3-SAT

Jaya Thomas
Department Of Computer Science & Engg.
Indian Institute of Technology,Indore

Narendra S. Chaudhari
Department Of Computer Science & Engg.
Indian Institute of Technology,Indore

ABSTRACT

In this paper, we elaborate the security threats that exist on hybrid cryptosystem based on satisfiability problem. In such system the encryption is carried out by generating 3-SAT clauses by random insertion of literal in a given 2-SAT clause instance. The solution of 2-SAT clause instance gives the secret key and the placement of literal for conversion to 3-SAT gives the position vector. Two crucial parameter for encryption. Thus, the system seems to be robust. However, the security of such system is at stake, when we apply the polynomial solvability formulation of 3-SAT[2]. Here, we propose a chosen plain text attack on such system using polynomial solvability of 3-SAT as reported in[3]. We observe that the complexity of the attack is O(3^n), where n is the number of clauses.

Keywords

Public Key, Satisfiability,2-SAT, 3-SAT, NP-complete, Secret key.

1. INTRODUCTION

Security is a major concern among researchers, especially regarding security of electronic transactions on internet. With the evolving needs and advancement in technology, have called researchers, for development of fairly secure cryptosystem. The security threats to the existing cryptosystem in quantum computer environment, has lead researchers to explore other cryptosystem based on some Nondeterministic Polynomial Complete(NP Complete) problem[9][10][11].

The most secured encryption scheme like RSA, based on factorization and discrete logarithms are no more, a secure system as their strength is compromised in quantum computers[8]. Thus, Knapsack Cryptosystem[5][7], Hybrid Cryptosystem[1], Cryptosystem based on Matrix Cover[6] are some of the newly proposed cryptosystem. Here, the question arises that cryptosystem which are build using these NP-complete problem are really secure. In this paper, we basically emphasize this aspect.\

Using the results reported in [2] we show that, how the security of such cryptosystem is at stake. To discuss this we have chosen Hybrid cryptosystem, which is based on satisfiability problem.

The cryptosystem for Alice and Bob consists of the following steps: Key Generation, encryption process at Alice end and decryption process at Bobs.

1.1 Key Generation in Hybrid Cryptosystem

The key generation process of this system uses both 2SAT and 3SAT. Thus, the two phase of transformation are involved. The steps for the key generation are summarized below:

1. Generate the random 2SAT clauses using C-2-SAT[1].
2. Use BinSat[1] algorithm to find the values of the literals such that the clauses are satisfiable i.e. *S*.
3. The obtained solution will constitute the secret key.
4. After creating the 2 − *SAT*, Alice camouflages it as an 3 − *SAT* problem, which is difficult. For each clause she proceeds, in this way:
   4.1 Add to each clause one literal at the appropriate position,
   4.2 Save the position of the added literal in a vector *V* and creates the integer $a = v_1v_2...v_m$.
5. Alice publishes finally the obtained 3 – *SAT* problem using C-3-SAT[1] which forms its *public key*.

1.2 Encryption in Hybrid Cryptosystem

When Alice wants to send a binary message $M = m_1,m_2....m_l$ to Bob, she reads for example the value *n* and *e*, Bob's RSA public key(*n*,*e*), calculates

$a' = a^e(\text{mod } n)$ and transmits (C, a'), where $C = M \oplus S$

1.3 Decryption in Hybrid Cryptosystem

To decrypt the message $C = c_1c_2...c_l$, Bob using its RSA secret value *d*, calculates

$$a = a'd(\text{mod } n),$$

and then deduces the vector V . He removes the variables added to the published 3 − SAT of Alice, and then obtains the corresponding 2 − SAT, thereafter its solution $S = (s_1, s_2, ..., s_l)$ by using the BinSat* algorithm [ ] and deduces the message of Alice by: $S \oplus C = M$.

The paper is organized as follows, Section II we will discuss the proposed methodology. In section III, we further explain the methodology using some examples. Section IV, we give the concluding remarks.

2. METHODOLOGY

In Hybrid cryptosystem, the public key–private key pair is generated using 2-SAT and 3-SAT instances respectively. It has been shown that both these problems are polynomial solvable[2][3]. Using these results, we here proposed an approach which shows that there exists a security threat on hybrid cryptosystem.

In this paper, we discuss the security attack on hybrid cryptosystem which is a slight variation from our previous work[4]. In the variation of the previous work now we have focussed on the results obtained after solving the public key instance of 3SAT by applying[2][3].Here we analyse that the major concern in this cryptosystem is the public key, private key generated using 3SAT and 2SAT resp. Thus, the attacker having access to Cipher text and public key could develop an algorithm for searching the private key used for encryption. Here, we are going to discuss one such approach for breaking the cryptosystem.

The approach is as follows, the cryptanalyst having access to the public key in 3SAT, would generate all possible solution value for the literals in the clause usi*ng Update Pairs() and Truth_Aalysis_3SAT()* [2]. After obtaining the values corresponding to the literals, the next step is to find the literals whose satisfying value would satisfy maximum number of the clauses. It is observed that these literals are the literals that are being inserted to convert 2SAT to 3SAT using C − *3* – SAT[1]. After finding the literals that got inserted, next is to find the position at which the literals where inserted. Analyze each clause to find the position of these literal in different clauses. Using the permutation of the obtained position values we can determine vector *a*. Since, this is an attack on the system, many possible values of a would be generated. Taking each possible value of *a* one at a time, we remove the literals at these positions, to reduce the given 3 SAT instance to 2SAT.

Thus, here a chosen-plaintext attack (CPA) attack model is studied. An attack model for cryptanalysis which presumes that the attacker has the capability to choose arbitrary plaintext to be encrypted and obtain the corresponding cipher text. The goal of the attack is to gain some further information which reduces the security of the encryption scheme. In the worst case, a chosen-plaintext attack could reveal the encryption scheme's secret key. The obtained *a* vector is tested for the chosen cipher text. To perform this task, according to the value of vector *a*, the 3SAT clause is converted back to 2SAT by removing the literals mentioned at the position by *a*. Once the 2SAT clauses are obtained, they are solved using BinSAT* to obtain *S*. The XOR of CipherText(*C)* and Secret Key(*S)* would give us back the plain text(*M*). The discussed method is briefed in the given algorithm.

Algorithm: Public key Reduction to 2SAT

```
Input: The public key (3SAT clauses)
Output: Position vector($a$) of randomly inserted literal
1.  Find the literal values in clauses using UpdatePair() and
    Truth analysis[3].
2.  For each literals count the number of times a literal
    occur in the second section of the generated ordered
    pair.
3.  Store the number of occurrence of literals in second
    part of ordered pairs[3].
                        //randomly inserted literals
4.  Find literal and its negation that occurs in the second
    part and sums up to the maximum count equivalent to
    highest individual occurrence of any literal.
5.  Maintain a position set for the randomly inserted literals
    by considering the position of above obtained literal in
    the public key.
6.  Perform permutation of the values to obtain the set of
    position vector.
7.  For each set obtained in step 6.
    7.1  Initialize $a$ as the set value.
    7.2  Remove the literals from 3SAT to get 2SAT
    7.3  Solve 2SAT using BinSAT* to get $S$.
    7.4  Obtain M by using $S \oplus C$
    7.5  If M is desired text then exit
         Else go to step 7.
```

3. EXPERIMENTATION

In this section we discuss two aspects one on the sender end and other with respect to the attacker. The proposed approach is purely based on the observation of the given system and the computational formulation proposed [2] showing the polynomial solvability of 3SAT. The sender is concern with the key generation and encryption process. The attacker on the other has access to the generated public key and the cipher text. We, elaborate the methodology using the following examples.

3.1 Example One: Generation of Keys

Alice chooses the message to be transmitted as *SS* = 01101, here let the total number of clause be *m* (*i.e.* 3). Then she executes C-2-SAT:

k = 1 : $i = 1, j = 2, b = 0.22, SS[1] = 0 \Rightarrow C_1 = (\bar{x}_2 \vee \bar{x}_1)$.

k = 2 : $i = 3, j = 4, b = 0.55, SS[3] = 1 \Rightarrow C_2 = (x_4 \vee \bar{x}_3)$.

k = 3 : $i = 4, j = 5, b = 0.63, SS[4] = 0 \Rightarrow C_3 = (x_5 \vee \bar{x}_4)$.

The following clauses in 2 – *SAT are generated* :

$$(\bar{x}_1 \vee \bar{x}_2)\ (\bar{x}_3 \vee x_4)\ (\bar{x}_4 \vee x_5).$$

To find the solution of the above equation we ca use BinSat* algorithm. The same is illustrated in the fig 1.

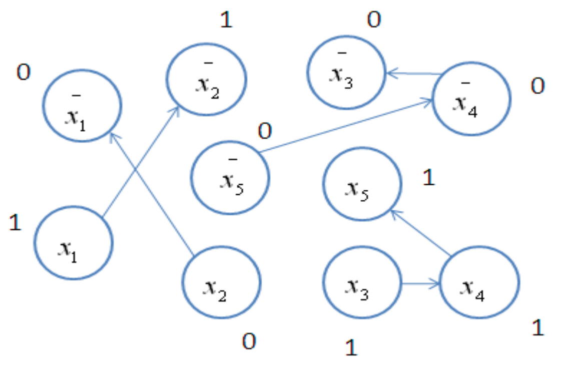

Fig 1: Illustrating BinSat* algorithm

Thus, Alice gets $S = 10111$. Then she uses $C - 3 - SAT$ to generate the public key for the cryptosystem.

k = 1 : $p = 4$, $b = 0.27 \Rightarrow C_1 = (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_4)$, $V[1] = 3$.

k = 2 : $p = 2$, $b = 0.68 \Rightarrow C_2 = (\bar{x}_3 \vee x_2 \vee x_4)$, $V[2] = 2$.

k = 3 : $p = 1$, $b = 0.11 \Rightarrow C_3 = (\bar{x}_1 \vee \bar{x}_4 \vee x_5)$, $V[3] = 1$.

Hence, Alice's public key is the 3-SAT:

$$(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_4)(\bar{x}_3 \vee x_2 \vee x_4)(\bar{x}_1 \vee \bar{x}_4 \vee x_5).$$

The vector $V = (3, 2, 1)$, then $a = 321$.

3.1.1 Encryption

Let $(n, e) = (36581, 5)$ be Bob's RSA public key. Alice ciphers a, i.e.,

$$a' = ae \ (mod\ n)$$
$$= 3215 \ (mod\ 36581)$$
$$= 2677$$

and its message $M = 0110001101$, using S, i.e.,$C = M \oplus S$.

Alice sends Bob: $(C, a') = (1101111010, 2677)$.

3.1.2 Attacker Approach

The attacker will proceed by solving the public key 3SAT clause instance using Truth_Analysis( ) & UpdatePairs( ) modules discussed in[2][3]. This would result in final set of ordered pairs generated with *first* and their respective *second*. Here we will slightly modify our approach[4] discussed. Rather, then finding the position vector a in order to convert 3SAT back to 2SAT, to find the private key S.

We will proceed by solving the ordered pair generated by the 3SAT clauses. The attacker would basically analyze the *first* and *second* part of the ordered pairs generated. The second of the ordered pairs consist of those literals which are crucial and dependent on the value of first. If *first* literals of any ordered pair are selected to be satisfied, then the corresponding second have to assigned values. It is found by continuous analysis that those literals, which are deliberately inserted to convert the given 2SAT into 3SAT would occur usually with less frequency in the *second* section of the ordered pairs.

Another point of observation is that as algorithm C-3-SAT would insert the literals from the specified range specified by the user; Like in the given example the range specified is between 1 to 5. Thus literal or its negation inserted to convert 2 SAT to 3SAT would be from within this range. Thus the occurrence of inserted literals would be more as compared to other.

In order to attack such system we need to keep track on the frequency of the literals particularly in the *second* section of the generated ordered pairs. The observed frequency for the inserted literals will be more, and can be easily identified.

Suppose the attacker get access to the public key

$$(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_4)(\bar{x}_3 \vee x_2 \vee x_4)(\bar{x}_1 \vee \bar{x}_4 \vee x_5).$$

Now, consider the result after applying the algorithm discussed [2][3].For the given example the result of ordered pair generated are recorded in the below given table:

Table 1:Literals and number of occurrence in second

First Pair	Second Pair
$\{x_1, x_2\}$	$\bar{x}_4$
$\{x_1, x_4\}$	$\{\bar{x}_2, x_5\}$
$\{x_2, x_4\}$	$\bar{x}_1$
$\{\bar{x}_3, x_2\}$	x_4
$\{\bar{x}_3, x_4\}$	x_2
$\{\bar{x}_2, x_4\}$	$\bar{x}_3$

$\{ x_1 , \bar{x}_5 \}$	$\bar{x}_4$
$\{ x_4 , \bar{x}_5 \}$	$\bar{x}_1$

3.1.3 Observation

After construction of the table check for the literals that occur frequently in *second* of the ordered pair . In the above given example, we notice that setting $\bar{x}_1 , \bar{x}_4$, occur in 2 pair each. As, per the observation we can conclude that these literals where inserted, randomly to form the 3SAT. It is further observed that there may exist some other literals also, which were inserted randomly. To check for them, evaluate again the table to find any such literal and its negation, such that the total number of clauses satisfied by them jointly is equal to the maximum value. If found, it indicates that the clause was also inserted randomly.

Here, literal ($x_2 , \bar{x}_2$) are the required pair. In this case both literal and its negation, position are included for permutation to find the vector *a*.

For, the given example we notice that $\bar{x}_1$ occur at position 1 in both first and third clauses. $\bar{x}_4$ occur at position 3 and 2 in clauses first and third resp. The third literal inserted was either x_2 or $\bar{x}_2$. Here, in this example the position of x_2 and $\bar{x}_2$ is 2. So, the permutation is to be carried by varying the obtained set {1}{3,2}and {2}. The possible values of position vector will be as discussed below the list may be varying.

{132},{122},{321}{221}{312}{231}{212}{213}{123}.

Now, as per the obtained position vector set removing the literals from 3SAT to change it to 2SAT. Finally, solving the 2SAT using BinSAT* to obtain secret key (*S*). Then applying M = S $\oplus$ C , we can obtain the required message. Note that the above mentioned test should be carried out first on known plain text, cipher text pair, to deduce the secret key. Once, the secret key is obtained, it can be used to decipher the further communication.
Since, we note the {321} is the required position vector. Hence, the attack is feasible.

3.2 Example Two

In this example we are going to analyze public key with 6 clause and 5 literals .Applying the algorithm[1], let us consider the 2-SAT clauses generated.

$$(x_1 \vee x_2) (x_1 \vee x_4) (x_4 \vee \bar{x}_2)(x_1 \vee \bar{x}_2)(x_2 \vee \bar{x}_1)(x_2 \vee \bar{x}_3)$$

Thus, Alice gets here secret key S = 11011 by applying BinSAT*. Then she uses $C - 3 - SAT$ to generate the public key for the cryptosystem.

k = 1 : $p = 3, b = 0.27 \Rightarrow C_1 = (\bar{x}_3 \vee x_1 \vee x_2), \ V[1] = 1.$

k = 2 : $p = 3, b = 0.68 \Rightarrow C_2 = (x_1 \vee x_3 \vee x_4), V[2] = 2.$

k = 3 : $p = 3, b = 0.11 \Rightarrow C_3 = (x_4 \vee \bar{x}_2 \vee x_3), V[3] = 3.$

k = 4 : $p = 4, b = 0.27 \Rightarrow C_4 = (x_1 \vee \bar{x}_2 \vee \bar{x}_4), \ V[4] = 3.$

k = 5 : $p = 3, b = 0.68 \Rightarrow C_5 = (x_2 \vee \bar{x}_3 \vee \bar{x}_1), V[5] = 2.$

k = 6 : $p = 4, b = 0.11 \Rightarrow C_6 = (x_2 \vee \bar{x}_4 \vee \bar{x}_3), V[6] = 2.$

Thus the public key generated is:

$$(\bar{x}_3 \vee x_1 \vee x_2)(x_1 \vee x_3 \vee x_4)(x_4 \vee \bar{x}_2 \vee x_3) (x_1 \vee \bar{x}_2 \vee \bar{x}_4)(x_2 \vee \bar{x}_3 \vee \bar{x}_1)(x_2 \vee \bar{x}_4 \vee \bar{x}_3)$$

To the above obtained public key applying the algorithm[2][3], we have the following results.

Table 2: Literals and number of occurrence in second

First Pair	Second Pair
$\{ x_3 , \bar{x}_1 \}$	$\{ \bar{x}_4 , x_2 \}$
$\{ \bar{x}_1 , \bar{x}_2 \}$	$\{ \bar{x}_3 , x_4 \}$
$\{ \bar{x}_1 , \bar{x}_3 \}$	$\{ \bar{x}_2 , x_4 \}$
$\{ \bar{x}_1 , \bar{x}_4 \}$	$\{ x_2 , x_3 \}$
$\{ \bar{x}_3 , \bar{x}_4 \}$	$\{ \bar{x}_2 , x_1 \}$
$\{ \bar{x}_4 , x_2 \}$	$\{ x_3 \}$
$\{ x_2 , \bar{x}_3 \}$	$\{ x_1 , x_4 \}$

$\{ \bar{x}_1 , x_2 \}$	$\{ \bar{x}_4 , x_3 \}$
$\{ \bar{x}_1 , x_4 \}$	$\{ \bar{x}_3 , \bar{x}_2 \}$
$\{ x_2 , x_4 \}$	$\{ x_1 \}$
$\{ \bar{x}_2 , x_1 \}$	$\{ \bar{x}_3 \}$
$\{ x_3 , x_1 \}$	$\{ x_2 \}$
$\{ \bar{x}_2 , x_4 \}$	$\{ \bar{x}_3 \}$
$\{ x_4 , x_3 \}$	$\{ x_1 , x_2 \}$
$\{ \bar{x}_2 \}$	$\{ \bar{x}_3 \}$
$\{ x_3 \}$	$\{ x_2 \}$

3.2.1 Observation

After construction of the table check for the literals that occur frequently in *second* of the ordered pair . In the above given example, we notice that occurrence of literal $\bar{x}_3 , \bar{x}_4$, is prominent. Thus, as, per the observation we can conclude that these literals where inserted, randomly to form the 3SAT. It is further observed that the literals $x_2 , \bar{x}_2$ are equally occurring. Thus the position vectors corresponding to all these literal must be considered for the permutation to find the position vector.

Looking at the position of $\bar{x}_3$ which occurs at position {1,2}, literal position for $\bar{x}_4$ position are {2,3}. Another literal and negation pair to be considered would be of $x_2 , \bar{x}_2$ at positions {1,2}. Now as the 6 clause generated we need to find the permutation combination for {1,2} ,{2,3} and {1,2}.

On applying all possible permutation of these value we will get the desired combination pair of {1,2,3,3,2,2}. Although the approach would be hit and trial, by applying all possible permuted value and removing the literals from the position, to reduce it to 2SAT instance. Thereafter applying the BinSAT* to find the literals value and thus, the desired secret key.

Then applying M = S $\oplus$ C , we can obtain the required message. Note that the above mentioned test should be carried out first on known plain text, cipher text pair, to deduce the secret key. Once, the secret key is obtained, it can be used to decipher the further communication.

4. CONCLUSION

In this paper we have, illustrated the analysis of attack on hybrid cryptosystem. We have used the results[2][3] to indicate that some kind of attack is possible on such cryptosystem by determining the position vectors. Although the approach is hit and trial but the approach is feasible and is tested for 10 literals. The complexity of the attack O(3^n), where n is the number of clauses. Thus, the complexity increases with the number of clauses.

5. REFERENCES

[1] L Rezkallah, S. Bouroubi An new hybrid cryptosystem based on the satisfiability Problem , downloaded from the site www.laid3.usthb.dz/road/horizontal/ road3909.pdf

[2] Narendra .S. Chaudhari, ,Feb 2011 Polynomial Solvability of 3-SAT -Part III: Polynomial algorithm for 3-SAT ,NHSS,Udaipur,India, ISBN : 978-81-7906-266-1 pp-71-76.

[3] Narendra .S. Chaudhari, Feb 2011 Polynomial Solvability of 3-SAT - Part II: Algorithmic formulations for 2-SAT, NHSS, Udaipur, India, ISBN :978-81-7906-266-1 pp-59-64 .

[4] Jaya Thomas, Narendra .S. Chaudhari , Apr 2011,Polynomial Solvability of Satisfiability and its Implication to Hybrid Cryptosystem Security International Conference on Em--erging Trends in Networks and Computer Communication (ETNCC), 2011.Udaipur,pp:521-54.

[5] Kobayashi, K; Tadaki, K; Kasahara, M; Tsuiji, S; A Knapsack cryptosystem based on multiple knapsacks ,ISITA 2010, pp428 – 432.

[6] K.B. Lakshmanan and Ravi Janardan , A Public-Key Cryptosystem based on the Matrix Cover {NP} - Complete Problem. Advances in Cryptology: Proceedings of Crypto 82, R. L. Rivest and A. Sherman and D. Chaum,editors. , volume 0, Plenum Press, New York, 1983. Pages 21-37.

[7] A. Shamir, A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem, *IEEE Trans. Inform.Theory*, vol. IT-30, 1984, pp. 699-704.

[8] Peter W. Shor, 1997 Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer Siam Journal on Computing - SIAMCOMP , vol. 26, no. 5, pp. 1484-1509.

[9**]** Massimo Caboara , Fabrizio Caruso' and Carlo Traverso October 2010, Lattice Polly Cracker cryptosystems Journal of Symbolic Computation Volume 46, Issue 5, May 2011, pp. 534-549.

[10] Rainer Steinwandt, Willi Geiselmann , Regine Endsuleit 2002, Attacking a polynomial-based cryptosystem: Polly Cracker International Journal of Information Securi--ty Volume 1, Number 3, pp143-148.

[11] R. C. Merkle and M. E. Hellman, "Hiding Information and Signatures in Trapdoor Knapsacks," *IEEE Trans. Inform. Theory*, vol. 24, 1978, pp. 525-530.