KNAPSACK CIPHER AND CRYPTANALYST USING HEURISTIC METHODS

Radomil Matoušek

Institute of Automation and Computer Science
Brno University of Technology,

Institute of Scientific Instruments
Academy of Sciences of the CR

Abstract: This paper describes a method of a cryptanalyst for a knapsack cipher. The deciphering method is based on the application of a heuristic random search hill-climbing algorithm, together with a genetic algorithm. It is shown that such an algorithm, implemented in Matlab environment, could be used to break a knapsack cipher. Some other aspects of the problem are discussed, too.

Keywords: Knapsack Ciphers, Hill-Climbing Algorithm, Genetic Algorithm, and Cryptanalyst.

1. INTRODUCTION

Knapsack cryptosystems belong to few major categories of public key cryptosystems. Whitfield Diffie and Martin E. Hellman published their landmark paper "New Directions in Cryptography" [01] in 1976, invented the idea of public key cryptography and proposed the fundamental technique of key agreement using the discrete log problem. Although now proven (major kinds) to be insecure, the first generalised public-key algorithm was the knapsack algorithm developed by Merkle and Hellman. Makle and Helleman published this first knapsack cipher in 1978 [02]. The first fall of Knapsack cryptosystems [02] came in 1982 by Shamir [03, 04, and 05]. A story of the rise and fall of many Knapsack cryptosystems is now extensive, as one can see from many papers, e.g., [06, 07, 08, 09, 10, 11, 12, 13, 14, and 15].

The algorithm is worth examining because it demonstrates several important public-key encryption concepts.

The knapsack problem may be stated as follows:

Given a collection of items, each with a different weight, is it possible to collect together a subset of the items to add up to a given weight? In mathematical terms, given a set of values $M_1, M_2, \ldots, M_n$ and a sum S, find the values of b_i such that:

$$S = b_1M_1 + b_2M_2 + \ldots + b_nM_n$$

Each b_i can only take a value of either zero or one. Take for example a collection of items weighing 1, 7, 8, 15 and 20. To pack a knapsack weighing 30, use weights 7, 8 and 15. It would not be possible to pack a knapsack that weighs 17.
The difficulty in solving this problem appears to grow exponentially with the number of items.

As shown for example in [15, 16, and 17], the special heuristic methods (Hill Climbing or Genetic Algorithms) can be used for solving Knapsack problem and appears also useful in the cryptanalysts Knapsack Ciphers, as illustrated in the present paper.

2. KNAPSACK CIPHERS

A knapsack cipher algorithm is based on the NP-complete knapsack-packing problem. This cipher encodes a plain text message as a solution to a series of knapsack problems. A block of plain text equal in length to the number of items in the collection selects the items in the knapsack. The resulting sum is the cipher text. This is shown below.

Plain text:	1 0 0 1 1	1 1 0 1 0	0 1 0 1 1	0 0 0 0 0
Knapsack	1 7 8 15 20	1 7 8 15 20	1 7 8 15 20	1 7 8 15 20
Cipher text	1+15+20 = 36	1+7+15 = 23	7+15+20 = 42	0 = 0

The idea is to create two different knapsack problems. One is easy to solve (superincreasing), the other not (trapdoor). Using the easy knapsack, the hard knapsack is derived from it. The hard knapsack becomes the public key. The easy knapsack is the private key. The public key can be used to easily encrypt messages, but cannot be used to decrypt messages. The private key easily decrypts the messages.

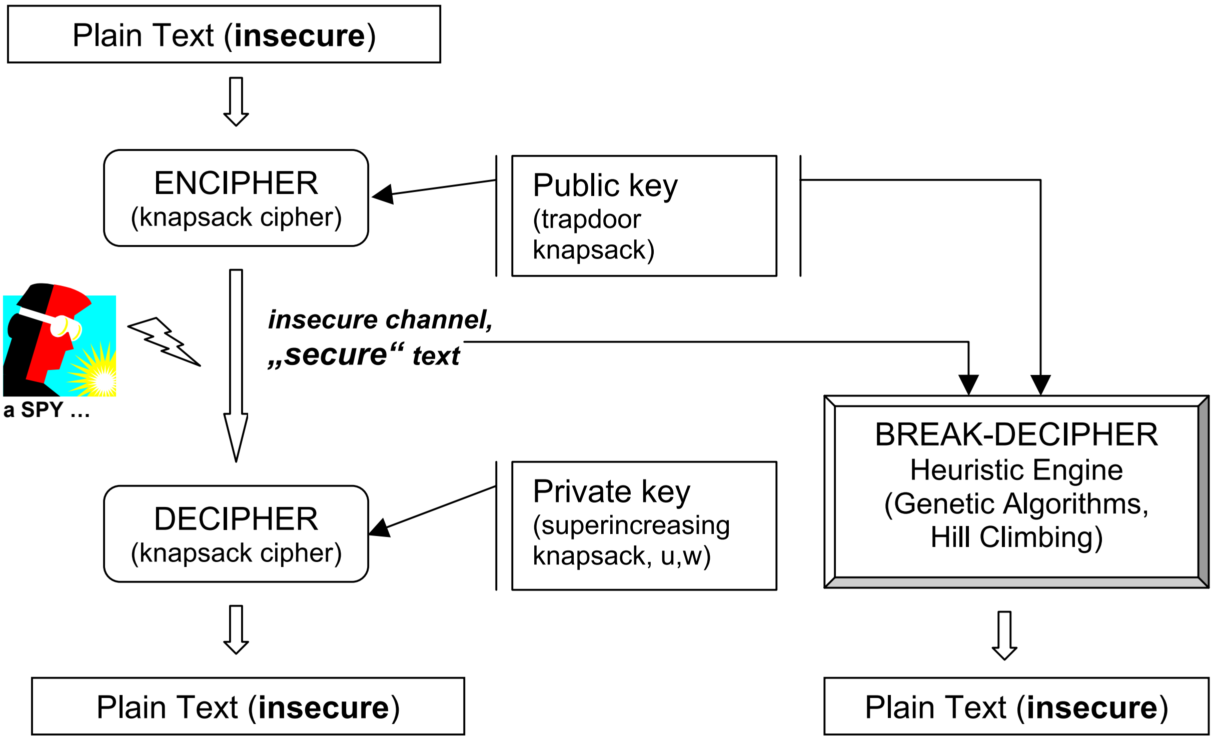

Fig. 1.: Merkle-Hellman cryptosystem and cryptanalysts by means Heuristic methods.

Merkle-Hellman Cryptosystem

Merkle and Hellman [02] describe basic method to construct a knapsack cryptosystem:

The easy knapsack is an additive knapsack whose weights form a superincreasing sequence. This easy knapsack is transformed into a hard (trapdoor) knapsack whose weights are the results of the modular multiplication of the easy knapsack weights. Using the same example as in [02], the easy knapsack vector is <171, 196, 457, 1191, 2410>, the message is <1, 1, 0, 1, 0>, the transformation is

$$a_i = a_i' \cdot w \bmod u\,,$$

where a_i and a_i' are the weights in the hard and easy knapsack cargo vectors respectively, w=2550 and u=8443 are integers such that gcd(w,u) = 1 and u > $\Sigma a_i'$, and the hard knapsack cargo vector is <5457, 1663, 216, 6013, 7439>. (The reason for ensuring gcd(w,m) = 1 is that it guarantees w^{-1} exists. The reason for ensuring u>$\Sigma a_i'$ is that we need a unique solution in x.) The subset sum S of the hard additive knapsack problem is 5457*1 + 1663*1 + 216*0 + 6013*1 + 7439*0 = 15115. Using the inverse transformation,

$$S' = S \cdot w^{-1} \bmod u\,,$$

where w^{-1} = 3950, we get S' = 3797, where S' is the subset sum of the easy additive knapsack problem. Then we proceed to use a polynomial-time algorithm to solve the easy knapsack problem <171, 196, 457, 1191, 2410> * x = 3797, to recover the message vector x = <1, 1, 0, 1, 0>. If a_i's are large numbers, solving $\Sigma a_i * x_i$ = S is hard but solving $\Sigma a_i' * x_i$ = S' is easy.

3. HEURISTIC METHODS

A Matlab® environment for implementation of these algorithms was used.

- HC algorithm (Hill Climbing with h_1-transformation)
- GA algorithm (Genetic Algorithm and GA-FIS)
- GA-HC algorithm (hybrid of GA and HC algorithms)

We must note that the heuristics are algorithms for which we are not able to guarantee the computation of a correct result in a reasonable time. Their basic advantages are simplicity and robustness. Genetic Algorithms and Hill Climbing are well-known representatives of heuristics.

Only for interest is make table 1, which shown the complexity of knapsack decryption:

n	***2^n×n (array)***	***Memory [Bytes]*** [**]		***Time [s]*** [*]
1	2×1	16 B ≡	16 B	0
10	1024×10	81920 B ≡	80 kB	0
20	1048576×20	167772160 B ≡	160 MB	3.5
21	2097152×21	352321536 B ≡	336 MB	7.4
22	4194304×22	738197504 B ≡	704 MB	15.2
23	8388608×23	1543503872 B ≡	1472 MB	31.9

[*] Depends on algorithm and computer hardware (Athlon®XP1800+, 768MB, KT266).
[**] Environment: Matlab® R12

Table 1: The Complexity of the KS problem brute force solutions in Matlab® environment.

4. IMPLEMENTATION

The heuristic algorithms were implemented as m-function and tested. In all algorithms a binary string of the length *n* represents a solution **x** to the problem S: the *i*-th item is selected for the knapsack iff *x*(*i*) = 1. The fitness of each string is determined as:

$$fitnes_{\min} = abs\left(\sum_{i=1}^{n} a_i x_i - S\right),$$

where fitness function is zero for feasible solution **x**, i.e., solutions such that $\sum_{i=1}^{n} a_i x_i = S$, and is greater than zero otherwise.

Function for generating of the public and private key is follow:

```
% Markle & Halleman knapsack cipher
% ---------------------------------------------
% GENERATOR FOR KNAPSACK:
%
% [superincreasing_knapsack,... --> s
%  trapdoor_knapsack,...         --> t
%  u,w,...                       --> t = w*s mod u,
%                                --> s = w^(-1)*t mod u
%  ]=MHgenerKS(length_of_knapsack)
%
%  ex.: [s,t,u,w]=MHgenerKS(5)

function [s,t,u,w]=MHgenerKS(lenKS);

% superincreasing_knapsack
a=round(rand(1,lenKS)*10)+1;
s(1)=a(1);
for i=2:lenKS,
s(i)=sum(s(1:i-1))+a(i);
end

% trapdoor_knapsack
u=2*s(lenKS)+round(rand(1)*10)+1;
w=13;
while gcd(u,w)~=1;
w=w+1;
end
t=mod(w.*s,u);
```

5. RESULTS

In our test cases of knapsack cipher we used methodology very similar as for general knapsack problem. A result is not restricted on unreal cases number of items in knapsack (many authors used vary short knapsack) for example in Chapter 2 is fifth items.

The U_1 and U_2 (theoretical) characteristics were designed [18] to compare results of the heuristics:

$$U_1 = \frac{\text{number of runs with optimal heuristic search}}{\text{number of all runs}} \times 100\ [\%]$$

$$U_2 = \frac{\text{sum of items of all solutions found by heuristic}}{\text{sum of items of all optimal solutions}} \times 100\ [\%]$$

Simple and short knapsacks like the one presented are of no use in real implementations. They are too easy to break. Practical implementations should contain at least 200 terms (items). Each term in the super-increasing knapsack should be 200 bits long. Knapsacks of this size are infeasible to solve by brute force. Presented results show the performance of used methods for KS cipher of a given size (up to 50 items). These results are only approximation of real hard knapsack cipher. Because, the results can by understand as good, but generalised next carefully.

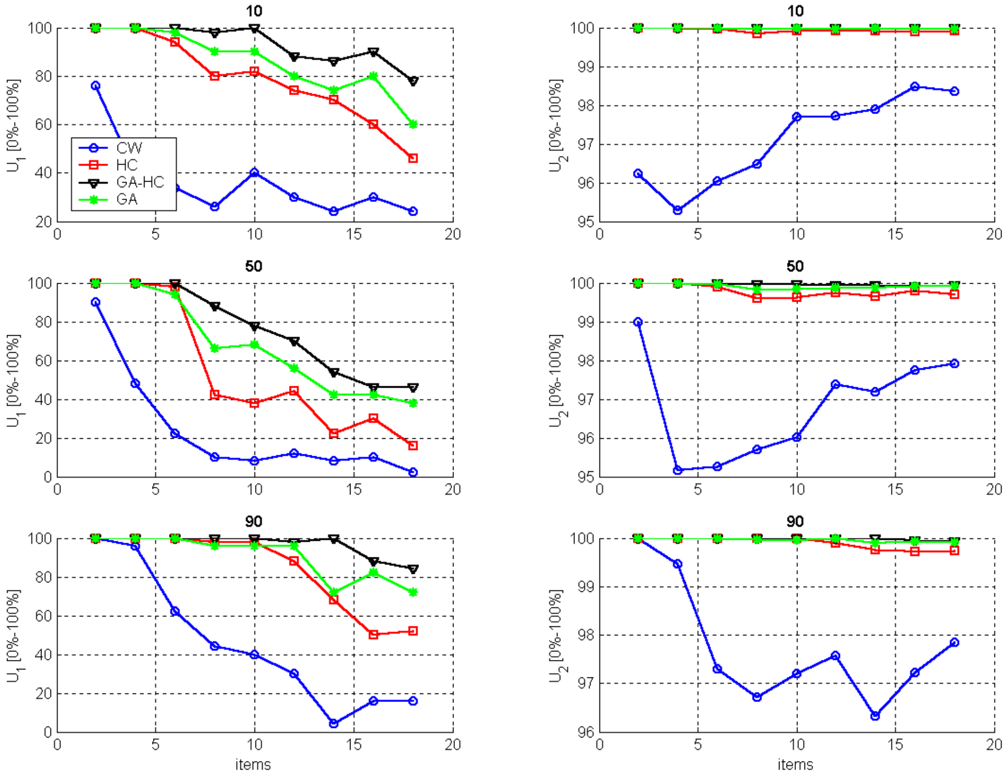

Fig. 2: Comparison of HC, GA and GA-HC methods for different KS ciphers.

Acknowledgements

The paper was supported by the projects CEZ: J22/98: 261100009 “Non-traditional methods for investigating complex and vague systems”, MŠMT ME 526, and MŠMT No.: 260000013.

REFERENCES

[01] Whitfield Diffie, Martin E. Hellman. New Directions in Cryptography. IEEE Transactions on Information Theory, vol. IT-22, no. 6, November 1976, pp. 644-654.

[02] Ralph C. Merkle, Martin E. Hellman. Hiding Information and Signatures in Trapdoor Knapsacks. IEEE Transactions on Information Theory, vol. IT-24, 1978, pp. 525-530.

[03] Adi Shamir. A Polynomial-time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem. Proceedings of the IEEE Symposium on Foundations of Computer Science. IEEE, New York, 1982, pp. 145-152.

[04] Adi Shamir. A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem. In David Chaum, Ronald L. Rivest, Alan T. Sherman. editors, Advances in Cryptology – CRYPTO ’82. Plenum, New York, 1983.

[05] Adi Shamir. A Polynomial-time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem. IEEE Transactions on Information Theory, vol. IT-30, no. 5, September 1984, pp. 699-704.

[06] Andrew M. Odlyzko. The Rise and Fall of Knapsack Cryptosystems. In Carl Pomerance, editor, Cryptology and Computational Number Theory, Proceedings of Symposia in Applied Mathematics, vol. 42. American Mathematics Society, Providence, RI, 1990, pp. 75-88. Available at http://www.research.att.com/~amo/doc/arch/knapsack.survey.ps

[07] Valtteri Niemi. A New Trapdoor in Knapsacks. In Ivan Bjerre Damgård, editor, Advances in Cryptology - EUROCRYPT '90, Lecture Notes in Computer Science, vol. 473. Springer, Berlin, 1991, pp. 405-411.

[08] Andrew M. Odlyzko. The Rise and Fall of Knapsack Cryptosystems. In Carl Pomerance, editor, Cryptology and Computational Number Theory, Proceedings of Symposia in Applied Mathematics, vol. 42. American Mathematics Society, Providence, RI, 1990, pp. 75-88. Available at http://www.research.att.com/~amo/doc/arch/knapsack.survey.ps

[09] Andrew M. Odlyzko. Cryptanalytic Attacks on the Multiplicative Knapsack Cryptosystem and on Shamir's Fast Signature Scheme. IEEE Transactions on Information Theory, IT-30, 1984, pp. 594-601. Available at http://www.research.att.com/~amo/doc/arch/knapsack.attack.ps

[10] Glenn Orton. A Multiple-iterated Trapdoor for Dense Compact Knapsacks. In A. De Santis, editor, Advances in Cryptology – EUROCRYPT '94, Lecture Notes in Computer Science, vol. 950. Springer, Berlin, 1995, pp. 112-130.

[11] Minghua Qu, Scott A. Vanstone. The Knapsack Problem in Cryptography. In Gary L. Mullen, Peter Jau-Shyong Shiue, editors, Finite Fields: Theory, Applications, and Algorithms. Contemporary Mathematics, vol. 168. American Mathematics Society, 1994, pp. 291-308.

[12] Harald H. Ritter. Breaking Knapsack Cryptosystems by l_∞-Norm Enumeration. 1st International Conference of the Theory and Applications of Cryptology - Pragocrypt '96, 1996, pp. 480-492. Available at http://www.mi.informatik.uni-frankfurt.de/research/papers/ritter.knapsack_cryptosystems.1996.ps

[13] Frank Rubin. Comments on "Cryptanalysis of Knapsack Cipher Using Genetic Algorithm". Cryptologia, vol. XVIII, no. 2, April 1994, pp. 153-154.

[14] Adi Shamir, Richard E. Zippel. On the Security of the Merkle-Hellman Cryptographic Scheme. IEEE Transactions on Information Theory, vol. IT-26, no. 3, May 1980, pp. 339-40.

[15] Richard Spillman. Cryptanalysis of Knapsack Ciphers using Genetic Algorithm. Cryptologia, vol. XVII, no. 4, October 1993, pp. 367-377.

[16] Matoušek, R. Hill Climbing and 0/1 Knapsack Problem, In Proceeding of Mendel 2002 Conference, Brno, July 2002

[17] Matthews, R. 1993 The use genetic algorithms in cryptanalists, Cryptologia, vol. XVII, no. 2, October 1993, pp. 187-201

[18] Matoušek, R., 2002, GA-HC: A Hybrid Genetic Algorithm, Fuzzy Colloquium, Zittau 2002

Address:
Institute of Automation and Computer Science
Brno University of Technology
Technická 2, 616 69 Brno, Czech Republic
Tel.:+420541143334

Institute of Scientific Instruments
Academy of Sciences of the CR
Královopolská 147, 612 64 Brno, Czech Republic

matousek@uai.fme.vutbr.cz