SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

FACULTY OF ELECTRICAL ENGINEERING
AND INFORMATION TECHNOLOGY

Department of Applied Informatics and Information Technology

Study field: 11-14-9 Applied Mathematics

Michal Šrámka

CRYPTANALYSIS OF SELECTED PUBLIC-KEY AND PRIVATE-KEY CRYPTOSYSTEMS

Ph.D. Dissertation

Advisor: Prof. RNDr. Otokar Grošek, PhD.

February 2006

I hereby certify that I am the sole author of this dissertation, up to the joint work that is properly acknowledged. Also, I declare that I provided references and acknowledgements to my best knowledge.

Michal Šrámka

Acknowledgements

I would like to thank Professor Otokar Grošek for his valuable suggestions, guidance, directions, and for his endless advising.

I would also like to thank to my family, friends, and to all people who are too numerous to mention, who aided, guided, encouraged, or otherwise influenced me through my study.

Contents

List of Tables

List of Figures

1 Introduction

Cryptology. Formed from the Greek words *kryptós* (hidden) and *lógos* (word). In today's information society, cryptology as the science of hidden, disguised information became one of the main tools for secure communication, privacy, trust, access control, electronic payments, electronic voting, and for countless other fields.

Cryptology is concerned with three dominant areas. *Cryptography*, the science of designing secure schemes, *cryptanalysis*, the science of breaking them, and *steganography*, the science of hiding information.

The goal of cryptanalysis is to find some weaknesses or insecurity in a cryptographic scheme. Cryptanalysis might be undertaken by a hostile attacker, attempting to subvert a system, or simply by the designer wishing to evaluate whether a proposed cryptographic scheme is secure.

In the past, the use of cryptography was a privilege reserved for armies, governments, and highly skilled specialists. This is no longer true. Cryptographic schemes become available for everyone. As cryptography evolved over the years, so did cryptanalysis. However, unlike cryptography which is a clearly defined science, cryptanalysis is still as much an art as it is a science. Success in cryptanalyzing a cryptographic scheme is a flash of inspiration almost as often as it the result of using cryptanalysis techniques alone.

Classical cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination, and of course luck.

On the other hand, modern cryptanalysis is almost entirely mathematized. For example, public-key cryptography requires a fundamentally different type of cryptanalysis than is used for cryptanalysis of private-key encryption schemes. Because public-key cryptography relies on hard mathematical problems, its cryptanalysis is essentially research into solving the underlying mathematical problems. Cryptanalysis of public-key cryptosystems is therefore very similar, virtually indistinguishable from research in any other area of mathematics. In addition, various academic relaxations are taken into account while cryptanalyzing a given encryption or other cryptographic scheme. These are discussed later in this dissertation. Finally, the modern cryptanalysis is not only concerned with encryption schemes. It covers analysis, weakness finding, and breaking of vast other cryptographic schemes - such as signature schemes, cryptographic hash functions, cryptographic protocols, and many others. Furthermore, the modern cryptanalysis is also concerned with fault and side channel attacks.

1.1 Goals of the dissertation

The major goal of this dissertation is to extend the knowledge in cryptology, namely in cryptanalysis. My research interests are mainly in public-key cryptosystems and their underlying problems. However, this dissertation explore and investigate both public-key and private-key cryptosystems, in order to contemplate, conclude, and especially extend the current issues in cryptology.

The minor goal of this dissertation is to present my capacity for both independent, original self-directed work and teamwork, that I believe constitute the skills of a researcher at the doctoral level.

This dissertation presents four research papers that in one way or another contribute to the area of study of cryptanalysis. Although the four papers explore different cryptographic schemes with different underlying mathematical problems, the outcome of the research exhibits the wide impact a cryptanalysis may result in.

In the first paper, the cryptanalysis of a cryptosystem based on the word problem leads to the design of another, more secure one. This is an example of a straightforward application of cryptanalytic results in cryptography. The research contained in this paper was presented at the *WartaCrypt '04* conference in Bedlewo, Poland on July 1-3, 2004. Parts of the research were also presented at the *Southeastern Weekend Algebra Meeting* in Hammond, Louisiana on November 5-7, 2004, and at the *69th Florida Academy of Sciences Annual Meeting* in Tampa, Florida on March 18-19, 2005.

In the second paper, the symmetry of the NTRU-like lattice (birotation) leads to the design of a faster algorithm for such lattice basis reduction. Hence the careful analysis of the underlying problem allows for a speedup in a known attack. The paper was presented at the *Third Pythagorean Conference* in Rhodes, Greece on June 1-7, 2003.

The third paper, cryptanalysis of a block cipher based on the Hopfield neural network, results in a solution for an old matrix conjugacy problem with many applications. The paper was presented at the *MoraviaCrypt '05* conference in Brno, the Czech Republic on June 15-17, 2005.

Last but not least, the fourth paper of this dissertation deals with recently proposed, poorly designed symmetric ciphers that are susceptible to classical and trivial attacks. The paper was presented at the *TatraCrypt '03* conference in Bratislava, Slovakia on June 26-28, 2003.

Sections 3.1.7, 3.2.7, 4.1.5, and 4.2.6 summarize the studied problems and provide some open problems for the further research.

1.2 Outline of the dissertation

There is a wide variety of cryptanalytic attacks against encryption schemes, and they can be classified in several ways. One distinction concerns what an attacker knows and does in order to learn secret information. The details of this classification are in the next Section together with some basic definitions of the terms. An outline of many cryptanalytic techniques is also provided.

The Section 3 on page 9 contains cryptanalysis of two public-key cryptosystems. This comprise the main part of this dissertation. The analysis of the first cryptosystem based on the combinatorial group theory also led to the proposal of a new encryption scheme based on similar properties as those analyzed. The cryptanalysis of the second cryptosystem based on the hardness of the lattice problems contains a novel non-deterministic algorithm for lattice basis reduction. This algorithm in many cases shows more promise than the algorithms known before.

Finally, the Section 4 on page 33 consists of cryptanalysis of two private-key (symmetric) cryptosystems. Although the cryptanalytic results are not so significant from the cryptological point of view, they led to a interesting observation. Namely, there are still cryptosystems proposed today that can be attacked using classical methods. As a byproduct a solution to an old mathematical problem was determined while cryptanalyzing of one of these cryptosystems.

Note, that each research topic is summarized at the end of the particular section.

2 Cryptanalysis

Cryptanalysis is a part of cryptology. The purpose of modern cryptanalysis is to analyze existing cryptosystems in order to reveal their weaknesses. As a by-product, the gained knowledge from cryptanalysis is then applied in cryptography for the purpose of design of more secure cryptosystems.

2.1 A few definitions

A person who performs cryptanalysis is called a *cryptanalyst*. However, traditionally and for mostly historical reasons, this person is often referred to as an *attacker*, *adversary*, *opponent*, *eavesdropper*, or *codebreaker*.

The general assumption in cryptanalyzing any cryptosystem is the assumption that the attacker has a complete knowledge of the cryptosystem being analyzed. This is known as the *Kerckhoffs' principle*[29]. On the other hand, if an analyzed cryptosystem is being hidden from the attacker, this would of course made the attacker's task more difficult. Fortunately, it turns out that the security of a cryptosystem cannot be based on this fact, because a knowledge of the cryptosystem can be obtained, assuming enough resources (time, funds, and minds). Therefore, the first lesson for cryptographers coming from cryptanalysis is that the security of a cryptosystem must solely depend on a key, not on the obscurity of the cryptosystem.

From now on, assume that an analyzed cryptosystem is always known, therefore its security is not based on *security-through-obscurity* paradigm just described.

An *attacker's goal* is to break the given cryptosystem. That means either to obtain a decryption key (*complete break*) or the ability to decrypt previously unseen ciphertext(s) (*partial break*).

To differentiate among the information given to the attacker, the following basic *attack models* are helpful. Some information comes from [44].

Ciphertext-only attack: An attacker possesses one or more ciphertexts.

This is the weakest type of attack. The attacker does not require any additional information, just ciphertext(s). In reality, this is the most common scenario, since an adversary can easily obtain the ciphertexts. The goal of the attacker is either to decrypt the ciphertexts in possession or obtain the decryption key.

From the security point of view, if a cryptosystem is susceptible to this attack, the cryptosystem is the least secure one (in comparison to the other attacks).

Known-plaintext attack: An attacker possesses one or more plaintexts and corresponding ciphertexts.

The attacker is given plaintext-ciphertext pair(s). The goal is to either determine the decryption key or to decrypt previously unseen ciphertext(s).

Chosen-plaintext attack: An attacker can choose plaintext messages and obtain the corresponding ciphertexts.

In this scenario, the attacker gains a black-box access to the encryption. That is, the attacker is able to encrypt any plaintext but does not know the secret key.

Mathematically speaking, the attacker has access to the oracle $\mathcal{O}$ that for any plaintext on input outputs the corresponding ciphertext. The goal of the attacker is again either to determine the decryption key or decrypt a previously unseen ciphertext(s).

Chosen-ciphertext attack: An attacker can choose ciphertext messages and obtain the corresponding plaintexts.

In this scenario, the attacker gains a temporary black-box access to the decryption. That is, the attacker is able to decrypt ciphertext but does not know the secret key.

Mathematically speaking, the attacker has access to the oracle $\mathcal{O}$ that for any ciphertext on input outputs the corresponding plaintext. If the goal of the attacker is to determine the decryption key, the attacker can feed any ciphertext as input. However, if the goal of attacker is to decrypt a ciphertext y, then the attacker can feed any ciphertext as input to the oracle except the ciphertext y. In other words, the goal of the attacker is to determine the decryption key or to read previously unseen ciphertexts.

2.2 Techniques of cryptanalysis

Almost all of the current techniques of cryptanalysis are based on mathematics. This is mainly because cryptography itself has been vastly mathematized. Those techniques that are not based on mathematics are usually techniques that are not used to attack the cryptosystem itself but its given implementation. Strictly speaking, these are the attacks afforded by incorrect security engineering.

One very important observation in cryptanalysis is that there may be *alternative keys* that perform the same decryption. This was first documented by Adi Shamir in 1982 while cryptanalyzing Merkle-Hellman knapsack cryptosystem[42], but such keys were also used in breaking ENIGMA, or they exist for RSA, too. The implication is that it may be possible to find an alternative key in a much easier way (and therefore faster way) than the original decryption key.

Mathematical techniques:

The areas of mathematics that are relevant to cryptanalysis are: probability and statistics, linear algebra, number theory, group/ring/field theory, complexity theory, combinatorics, graph theory, etc. Advanced mathematical, parallel, and distributed programming is often an essential requirement to the cryptanalysts.

Listing and describing all the cryptanalysis techniques available today is out of the scope of this dissertation, if not impossible. Therefore, very briefly, a few techniques with examples and references are provided. The relevant techniques are described in detail in the following sections.

A great source - a survey of techniques used in cryptanalyzing symmetric block cryptosystems is [37]. This source includes differential and linear cryptanalysis methods, too.

In addition to this two statistical attacks against block ciphers, probability and statistics provides many tools[44] for a cryptanalyst. Ranging from simple techniques as frequency analysis and counting, through various probabilities about a given language, to the often used Birthday paradox theorem.

It turns out that studying algebraic properties of a cryptosystem is often very helpful in analyzing its security. Some cryptosystems can be described in previously unconsidered algebraic ways. Similarly, in recent years, the lattice reduction related problems become an important tool not just in cryptography but also in cryptanalysis. Considering and using these new approaches (in comparison to the original underlying problems and their complexity) usually lead to a faster acquisition of decryption keys.

Non-mathematical techniques::

The currently known non-mathematic techniques are those that try to take advantage of flawed implementations of cryptographic schemes. These techniques are classified as *side-channel attacks* and *fault attacks*, and are afforded by careless security engineering. Further, these techniques rely on observing and exploiting of available physical information.

A typical example of a side-channel attacks is a reaction attack[21]. In such scenario, if the attacker is possible to observe reactions of a real person or machine while performing encryption/decryption, then the attacker is able to gain a significant knowledge either about the message or the decryption key itself.

Other well-known side-channel attack techniques are the simple and differential power analysis, time analysis, analysis of radiation and emission (electromagnetic, sound, ...), and analysis of patterns (memory access, hyperthreads, ...).

In fault attacks, the attacker is allowed to actively alter the running cryptosystem. Using physical means the attacker tries to induce faults in order to extract information about the message or the decryption key.

The fault attacks can use as subtle means as operating at different frequency, change of current, change of temperature. Or not so subtle, often extreme, means - freezing, x-ray, laser, ...

The lesson from these side-channel and fault attacks is that cryptosystems must be implemented in such a way that no reaction can be observed, and every value that can be measured should have no relevant significance to the message or the key currently processed by the cryptosystem. Fortunately, avoidance of and protection from these attacks is often straightforward.

2.3 Is cryptanalysis dead?

Herbert Yardley, a famous US codebreaker, in his 1931 book, *The American Black Chamber*, describes the one-time tape cipher machine (one-time pad). Of it he says,

> *Sooner or later all governments, all wireless companies, will adopt some such system. And when they do, cryptography [he meant cryptanalysis] as a profession, will die.*

In a way, the cryptanalysis has been dead for many years[28]: Edward H. Hebern's cipher machines and Arthur Scherbius' Enigma machines, used in World War II, could not be cryptanalyzed in those years by study of just the ciphertext, no matter how many were available. Nor could they be recovered by exhaustive search, since it was beyond the technology available that time (computers available today would be able to solve it). Therefore, *pure cryptanalysis* was already dead.

So, cryptanalysis was powerless against good cryptosystems as early as World War II. And it is still powerless against good cryptosystems today. Many cryptosystems known today cannot be broken by any known techniques of cryptanalysis. Indeed, in such systems even a chosen-plaintext attack cannot yield the decryption key. In a sense, then, *traditional cryptanalysis* is dead.

However, breaking a cryptosystem does not necessarily mean finding a practical way for an attacker to recover the plaintext from just the ciphertext. In *academic cryptanalysis*, the rules are relaxed considerably[37]. Breaking a cryptosystem simply means finding a weakness in the cryptosystem that can be exploited with a complexity less than brute-force. Never mind that brute-force might require 2^{128} encryptions; an attack requiring 2^{110} encryptions would be considered a break. Breaks might also require unrealistic amounts of known or chosen plaintext - 2^{56} blocks - or unrealistic amounts of storage - 2^{80}. Simply put, any weakness that proves that the cryptosystem is not as secure as advertised is considered a break. Another often encountered relaxation is the attack on a similar cryptosystem. This includes an attack against a block cipher with reduced rounds or an attack against a simplified variant of the original cryptosystem.

To conclude, cryptanalysis is not dead. The pure/traditional cryptanalysis just evolved into a *modern cryptanalysis* that using various (academic) relaxations looks for weaknesses in the cryptosystems. Although the cryptanalysis of the cryptosystems often does not lead to practical breaks of the systems, most of the knowledge gained from the weaknesses is then in turn applied back in in design of new, more secure cryptosystems.

3 Public-Key Cryptosystems

The first major part of this dissertation consists of cryptanalysis of two public-key cryptosystems.

A public-key cryptosystem based on combinatorial group theory is first described. The necessary combinatorial group theory definitions and results are mentioned. A new, previously unpublished, chosen-ciphertext attack of polynomial complexity is described. In addition to criticizing and attacking the Wagner-Magyarik public-key cryptosystem, a new public-key cryptosystem is proposed. This is an example where cryptanalysis of one cryptosystem led to design of another one.

The lattice basis reduction is a powerful technique in cryptanalysis. It can be used to break many different kind of problems. Here, it is shown how it can be used to find the private-key of the original NTRU public-key cryptosystem. A symmetry of the NTRU-lattice is described. A randomized hill-descending algorithm that uses this symmetry is proposed to reduce the NTRU-lattice, hence obtaining the private-key faster than traditional methods. Moreover, instances of the algorithm can run independently in parallel, thus providing linear speedup.

3.1 Public-key Cryptosystems Based on Combinatorial Group Theory

In this section, we[6] analyze and critique the public-key cryptosystem, based on combinatorial group theory, that was proposed by Wagner and Magyarik in 1984. Their idea is actually not based on the word problem as they claim, but on another, generally easier, premise problem. Moreover, the idea of the Wagner-Magyarik system is vague, and it is difficult to find a secure realization of this idea. We propose a public-key cryptosystem inspired in part by the Wagner-Magyarik idea, but we also use group actions on words. The security evaluation of these schemes leads to interesting new complexity problems in combinatorial group theory.

3.1.1 Introduction

A number of public-key cryptosystems based on combinatorial group theory have been proposed since the early 1980s, the first of which was probably the outline of Wagner and Magyarik [45]. A good overview of various other group-based systems is given in the dissertation of M.I. Gonzalez-Vasco[16]; see also [17].

First, a critique of the Wagner-Magyarik cryptosystem is proposed, which is followed by a proposal of a public-key cryptosystem based on finitely presented groups with hard word problem, and which are also transformation groups.

3.1.2 Some combinatorial group theory

Some basic definitions from combinatorial group theory are provided. More details and rigor can be found in texts like [31] or [33].

Let G be a group, defined by a presentation (X, R), where $X = \{x_1, x_2, \ldots\}$ is a set of *generators* and $R = \{r_1, r_2, \ldots\}$ is a set of *relators.* When the sets X and R are both finite we say that the group G is *finitely-presented.* A *word* w over X is a finite sequence of elements of the set $X \cup X^{-1}$. The *empty word* is the empty sequence, of length 0. A word which defines the identity element in the group G is called a *relator.* We say that two words w and w' are *equivalent* for the presentation (X, R) if and only if the following operations, applied a finite number of times, transform w into w':

(T1) Insertion of one of the relators $r_1, r_1^{-1}, r_2, r_2^{-1}, \ldots \in R \cup R^{-1}$, or of a trivial relator (of the form $x_i x_i^{-1}$ or $x_i^{-1} x_i$ with $x_i \in X$) at the beginning of a word, at the end of a word, or between any two consecutive symbols of a word.

(T2) Deletion of one of the relators $r_1, r_1^{-1}, r_2, r_2^{-1}, \ldots$, or of a trivial relator, if it forms a block of consecutive symbols in a word.

An application of one transformation of the form (T1) or (T2) is called a *rewrite step.* Every element g of $G = (X, R)$ can be described by a word over $X \cup X^{-1}$, usually in many ways; the length of the shortest word that describes g is called the *word length of g*. For a word w over some fixed alphabet we denote the length of w by $|w|$; also, for $g \in G = (X, R)$ we denote the word length of g by $|g|$.

The *word problem* of a group with generating set X, as introduced by Max Dehn in 1911, is the following decision problem: For an arbitrary word w over $X \cup X^{-1}$, is w equivalent to the empty word?

In the 1950's, Novikov and Boone independently showed that there are finite group presentations whose word problem is undecidable. It is an important fact that the decidability and the complexity of the word problem of a finitely generated group depend only on the group, and not on the generators or the presentation chosen (provided that one sticks to finite generating sets). In other words, if G has decidable word problem for some finite generating set X then G has decidable word problem for every finite generating set. Concerning complexity, a change of the finite generating set changes the complexity only linearly (see [32]). Therefore, we are allowed to talk about "the word problem of a group G" without referring to a specific presentation.

It was proved more recently that there are finitely presented groups whose word problem is NP-complete [36], [7], or whose word problem is coNP-complete [4].

By a group with *easy* word problem we will understand a group whose word problem is decidable in deterministic polynomial time. The other groups are said to have a *hard* word problem.

We will also deal with the following variant of the word problem, which we call *the word choice problem.* Let us fix a group G with a finite generating set X, and let us fix two words w_0 and w_1 over $X \cup X^{-1}$.
Input: A word w over $X \cup X^{-1}$.
Premise: w is either equivalent to w_0 or to w_1.
Question: Is w equivalent to w_0 ?

Note that this is a "premise problem", i.e., a problem with restrictions (pre-condition) on the input; an algorithm for solving a premise problem can assume that the pre-condition holds, and is not required to give correct answers (or any answer at all) on inputs that violate the pre-condition.

The word choice problem is rather different from the word problem. E.g., for a finitely presented group, the word choice problem is always decidable; and for a group with word problem in NP or in coNP, the word choice problem is in NP $\cap$ coNP. One sees from these examples that the word choice problem can be much easier than the word problem.

3.1.3 The Wagner-Magyarik cryptosystem

In 1984 Wagner and Magyarik [45] proposed a public-key cryptosystem "based on the word problem". The general scheme follows.

Setup: Let X be a finite set of generators, and let R and S be finite sets of relators such that the group $G = (X, R)$ has a hard word problem, and the group $G' = (X, R \cup S)$ has an easy word problem. Choose two words w_0 and w_1 which are not equivalent in G' (and hence not equivalent in G either).

Public key: The presentation (X, R) and the words w_0 and w_1.

Encryption: To encrypt a single bit $i \in \{0, 1\}$, pick w_i and transform it into a ciphertext word w by repeatedly and randomly applying the transformations (T1) and (T2) from page 10 for the presentation (X, R).

Decryption: To decrypt a word w, decide in the group G' which of ww_0^{-1} and ww_1^{-1} is equivalent to the empty word for the presentation $(X, R \cup S)$.

The *private key* is the set S. Actually, this is not sufficient (and [45] is not very precise at this point): the public key should be a deterministic polynomial-time algorithm for the word problem of $G' = (X, R \cup S)$; indeed, just knowing S does not automatically and explicitly give us an efficient algorithm (even if such an algorithm exists).

To make their system concrete, Wagner and Magyarik introduce the following collection of finitely-presented groups: The set of generators is $X = \{x_1, x_2, \dots, x_m\}$ and the set of relators R is any set of words of the following three types:

(R1) $y_i y_j y_k y_\ell y_i^{-1} y_k^{-1} y_j^{-1} y_\ell^{-1}$

(R2) $y_i y_j y_k y_i^{-1} y_j^{-1} y_k^{-1}$

(R3) $y_i y_j y_k y_i^{-1} y_k^{-1} y_j^{-1}$

where y_i, y_j, y_k, and y_ℓ stand for generators or inverses of generators, not necessarily distinct. We will call such presentations *Wagner-Magyarik presentations.*

For the private key S they propose any set of words of the following three types:

(S1) x_i (elimination of a generator)

(S2) $x_i x_j^{-1}$ (collapse of two generators to one)

(S3) $x_i x_j x_i^{-1} x_j^{-1}$ (commutator of two generators)

where x_i and x_j are any generators. A requirement on S is that it should contain enough relators so that the group $G' = (X, R \cup S)$ is isomorphic to a "partially commutative free group", i.e., a group generated by a subset of X and presented by a few commutation relations between generators. This will guarantee that the word problem of G' can be decided in polynomial time [46]. The words w_0, w_1 need to be chosen so that they are not equivalent in G'.

3.1.4 Critique of the Wagner-Magyarik cryptosystem

1. *Vagueness of the general scheme:* In its general form the Wagner-Magyarik cryptosystem is far too vague. To turn their idea into an actual cryptosystem, design questions would need to be answered:

(D1) How do we find appropriate presentations (X, R) and $(X, R \cup S)$, as well as a polynomial-time algorithm for the word problem of $(X, R \cup S)$?

(D2) How do we find appropriate words w_0 and w_1?

(D3) How is the random application of the transformations (T1) and (T2) carried out, and when does it stop?

(D4) Finally, once all these design choices have been specified, how secure is this cryptosystem?

2. *Vagueness and insecurity of the concrete specification:* In their specific example, Wagner and Magyarik give an answer to design question (D1), albeit an unsatisfactory one. Design questions (D2), (D3) and (D4) are left open. Concerning (D1), it is an open problem whether the word problem of the Wagner-Magyarik presentations is hard. It is certainly not hard for every choice of (R1), (R2), (R3); e.g., some of the choices lead to commutative groups. This means that in the Wagner-Magyarik system, key generation is problematic: making sure that the chosen R makes the word problem of (X, R) hard is itself apparently a hard problem. Concerning (D4), a reaction attack[18] and a chosen-ciphertext attacks are possible, both of complexity $O(m^2)$.

3. *Chosen-ciphertext attack:* In addition to the published reaction attack[18], it is possible to obtain the private key through a chosen-ciphertext attack: For any relator s of type (S1), (S2), and (S3), the attacker considers the word $w_0 s$, encrypts it (by applying the transformations (T1) and (T2) several times) and then observes the decryption, say w. If w is equivalent to w_0 in G' the attacker learns that s belongs to S (or is implied by relators in S, which means that one might as well assume that it is in S). The complexity of this attack is $O(m^2)$.

4. *Alternative keys:* Another problem (already mentioned in [45]) is the existence of alternative keys. More precisely, in order to decrypt one does not explicitly need the presentation $(X, R \cup S)$. Any homomorphic image of G with easy word problem will decrypt, as long as it separates w_0 and w_1. So, even if S might be hard to find, one also has to prove that any homomorphic image of G with easy word problem, is hard to find; this adds to the difficulty of proving the security of any concrete cryptosystem that follows the Wagner-Magyarik approach.

5. *Word choice problem:* An analytical flaw in the Wagner-Magyarik paper (and subsequent papers that comment on their paper) is the claim that the system is based on the word problem. In reality, it is based on the *word choice problem*, that we introduced earlier. We pointed out already that the word choice problem can be much easier than the word problem. In particular, it seems unlikely that this system could ever lead to NP-completeness. Instead, (NP $\cap$ coNP)-completeness is more likely to be the highest difficulty that we can hope for, regarding robustness to attack. It is generally believed that NP $\cap$ coNP is a strict subclass of NP. Although no (NP $\cap$ coNP)-complete decision problem is known (see e.g., [13], page 116), it is not hard to see that for every NP-complete decision problem one can construct a (NP $\cap$ coNP)-complete *premise* problem. See the next section for details.

6. *In summary:* The Wagner-Magyarik cryptosystem is not a cryptosystem, but an approach towards finding new public-key cryptosystems. As a research approach it is worthwhile, however, leading to interesting (yet unsolved) problems.

3.1.5 A PKC based on finitely presented transformation groups

We describe a public-key cryptosystem that has some similarity with the Wagner-Magyarik system, as far as the encryption is concerned. However, we use a group G whose word problem is known to be coNP-complete. The main difference is that for decryption we use the action of the group on words (instead of Wagner and Magyarik's homomorphic image G').

Our contribution is that (referring to point 1 in our critique of the Wagner-Magyarik system) we answer the design questions (D1) and (D2). Design question (D3) is addressed, but our method needs further study, and probably further improvements. Regarding question (D4), the security of our scheme is much better motivated than the security of the original Wagner-Magyarik system, but it is necessarily limited (due to the multitude of hard open problems in complexity, combinatorial group theory, and cryptography).

We pick a finitely presented group $G = (X, R)$ together with a faithful transitive action of G on $\{0, 1, 2\}^*$ (the set of all strings over the alphabet $\{0, 1, 2\}$). We can assume that the word problem of G is coNP-complete. We conjecture that the word choice problem of G is (NP $\cap$ coNP)-complete. The next section deals with a semigroup version of this question.

An example of such a group is constructed in [4], where it is called $G = \langle G_{3,1}^{\text{mod}\,3}(0, 1; \#) \cup \{\kappa_{321}\}\rangle$; it is closely related to the Higman-Thompson group $G_{3,1}$ (generalizing Richard Thompson's infinite finitely presented simple group $G_{2,1}$). This group has the property that if two elements $g_0, g_1 \in G$ of word length $\le n$ are different then there exists a word $z \in \{0, 1, 2\}^*$ of length $O(n)$ on which g_0 and g_1 act differently. Moreover, given a word $z \in \{0, 1, 2\}^*$ and a word w over a finite generating set of G, the word $(z)w \in \{0, 1, 2\}^*$ (resulting from the action of w on z) can be computed in deterministic time $O(|z| + |w|)$. For a definition of the Higman-Thompson groups, see also [5], [39] and [22].

Key generation: We first pick a word $x \in \{0, 1, 2\}^*$. For encrypting and decrypting 0 we choose a word $z \in \{0, 1, 2\}^*$ and, similarly, for 1 we choose a word $u \in \{0, 1, 2\}^*$; the three words x, z, u should be long enough so that it is impossible to guess them. For 0, we also choose $m - 1$ "intermediary words" $z_i \in \{0, 1, 2\}^*$ (with $i = 1, \ldots, m - 1$); similarly, for 1 we choose $m-1$ "intermediary words" $u_i \in \{0, 1, 2\}^*$ (with $i = 1, \ldots, m-1$). Here, m is a security parameter chosen so that 2^m or 4^m is very large; e.g., we could have $m = 100$ or $m = 200$. The two sets $\{z\} \cup \{z_i : i = 1, \ldots, m - 1\}$ and $\{u\} \cup \{u_i : i = 1, \ldots, m - 1\}$ are required to be disjoint.

Next, we choose a "system of words" over $X \cup X^{-1}$ for encrypting a bit 0, and a system of words over $X \cup X^{-1}$ for encrypting a bit 1. A system of words (say for encrypting 0) is a sequence of m finite sets $(Z_1, \ldots, Z_m)$. Each set Z_j is a small set of words over $X \cup X^{-1}$ (with e.g., 4 elements). Each element $w \in Z_j$ has the property that $(z_{j-1})w = z_j$, for $j = 2, \ldots, m - 1$; also, for each element $w \in Z_1$, $(x)w = z_1$, and for each element $w \in Z_m$, $(z_{m-1})w = z$. For 1, a similar system $(U_1, \ldots, U_m)$ of sets of words is chosen, with similar properties regarding x, u_j ($j = 1, \ldots, m - 1$), and u. The action diagram below shows the role of the intermediate words $z_i \in \{0, 1, 2\}^*$ and the action of the words in Z_j on the intermediate words:

$$x \xrightarrow{Z_1} z_1 \xrightarrow{Z_2} z_2 \xrightarrow{Z_3} \quad \ldots \quad \xrightarrow{Z_{i-1}} z_{i-1} \xrightarrow{Z_i} z_i \xrightarrow{Z_{i+1}} \quad \ldots \quad \xrightarrow{Z_{m-1}} z_{m-1} \xrightarrow{Z_m} z$$

The **private key** is (x, z, u). (The words z_i and u_i are required to remain secret but are not needed after key selection, i.e., they are not used in encryption or decryption.)

The **public key** consists of the presentation (X, R), as well as the two set systems $(Z_1, \ldots, Z_m)$ (for 0), and $(U_1, \ldots, U_m)$ (for 1).

Encryption: To encrypt a bit 0, randomly choose an element w_j in each set Z_j $(j = 1, \ldots, m)$, and concatenate these elements to form the word $w_1 w_2 \ldots w_m$. Next, as in the Wagner-Magyarik system, we rewrite $w_1 w_2 \ldots w_m$ by applying the relators of $G = (X, R)$ (as well as the trivial relators) randomly a "sufficiently large" number of times; see the discussion below concerning this rewriting. This yields some word W_0, encrypting 0. To encrypt a bit 1, the procedure is similar, but now the set system $(U_1, \ldots, U_m)$ is used.

Decryption: With a ciphertext w, compute $(x)w$. If $(x)w = z$, decrypt as a 0; if $(x)w = u$, decrypt as a 1.

Some design issues:

1. The words $x, z, u \in \{0, 1, 2\}^*$ are selected uniformly at random among words of length between n and $2n$. Here n is a security parameter; e.g., $n = 100$ or $n = 200$. Similarly, the intermediary words are selected uniformly at random among words of length between $n/2$ and $4n$.

   Another security parameter is m; e.g., $m = 100$ or $m = 200$.

2. How is the "system of words" $(Z_1, \ldots, Z_m)$ (and similarly $(U_1, \ldots, U_m)$) determined? For each pair of intermediary words (z_j, z_{j+1}) (for 0) we design a boolean circuit that maps z_j to z_{j+1}; similarly, we design a boolean circuit that maps u_j to u_{j+1}. These two circuits should be as similar as possible (in fact, when $z_j \neq u_j$, the same circuit could be used for both; we then can make them different in random details). If we want Z_{j+1} (and U_{j+1}) to have 4 elements we repeat this four times. Next, we use the correspondence between circuits and elements of the Higman-Thompson group $G_{3,1}$ (see [4]) to construct elements of G that simulate these circuits.

3. *Random rewriting:* The rewriting of an element from $Z_1 \times \ldots \times Z_m$ (respectively from $U_1 \times \ldots \times U_m$) could be done as follows. First enlarge the presentation $G = (X, R)$, by including R^{-1} (the set of inverses of the words in R) into the set of relators, and adding all cyclic permutations of words in $R \cup R^{-1}$ as well; this gives us the "symmetrized presentation" (X, R_s) of G. Next, we turn (X, R_s) into a string rewriting system by taking all rules of the form $u \to v$ for any (possibly empty) strings u, v over $X \cup X^{-1}$ such that $u^{-1}v$ is a relator in R_s. We also add the rules $1 \to a^{-1}a$ and $a^{-1}a \to 1$ for any $a \in X \cup X^{-1}$; here, 1 is the empty string. For rewriting a word w of length n we do the following:

PROCEDURE A: 1. choose a position in the word obtained so far; 2. choose a rule, and apply it at the chosen position (if the rule doesn't apply at this position, go back to step 1.).

After n repetitions of procedure A, we check whether every letter of w has been rewritten (this assumes that we marked the original letters of w); if not all letters have been rewritten, run procedure A another n times; keep repeating n runs of procedure A until all letters of w have been rewritten. At this point, most positions of w will have been rewritten many times.

Now we could mark all the letters in the word w' obtained so far, and start over with the rewriting until all positions in w' have been rewritten. All this could be repeated a few more times.

The encryption of 0 first chooses one out of 4^m elements from $Z_1 \times \ldots \times Z_m$ (respectively from $U_1 \times \ldots \times U_m$ for 1). The rewriting process then makes it hard to recognize what system of sets the chosen element $w_1 \ldots w_m$ originally came from. The rewrite rules are applied everywhere in the word, so that no local pattern from a set Z_j or U_j ($j = 1, \ldots, m$) remains. Because of the exponential number of choices for $w_1 \ldots w_m$, the role of the rewriting is less important than in the original Wagner-Magyarik idea. The role of the systems of words $(Z_1, \ldots, Z_m)$ and $(U_1, \ldots, U_m)$ is precisely to (exponentially) strengthen the confusion caused by rewriting, and this is one of the contributions of our paper. But the rewriting is nevertheless important, and research is needed to determine how (and how much of) the random rewriting should be done.

4. *Security, open problems:* A *alternative key* is any triple (x', z', u') of words over the alphabet $\{0, 1, 2\}$, with the properties that $(x')v = z'$ for any word v that encrypts 0, and $(x')w = u'$ for any word w that encrypts 1. For a known-plaintext or a chosen-ciphertext attack, suppose the attacker has a collection of plaintext-ciphertext pairs $(0, v_i)$, $(1, w_j)$ for $i = 1, \ldots, m$, and $n = 1, \ldots, n$. Finding (alternative) keys is the search version of the *common action problem* of groups elements, which we conjecture to be NP-hard; see the next section.

   Our complexity analysis in this paper refers to worst case complexity. For security, almost-all case complexity, or at least average case complexity is needed. Unfortunately, almost-all case and average case complexity are still relatively poorly explored, and still have definitional problems.

   Other open problems:

   - Is the word choice problem of the group $G = \langle G_{3,1}^{\mathrm{mod}\,3}(0, 1; \#) \cup \{\kappa_{321}\}\rangle$ an

(NP $\cap$ coNP)-complete premise problem? (The next section gives a result for semigroups.)

- Is the common action problem of the group $G = \langle G_{3,1}^{\text{mod}\,3}(0,1;\#) \cup \{\kappa_{321}\}\rangle$ NP-complete? (The next section gives a result for circuits and a connection with G.)

5. *Other groups that could be used in our public-key cryptosystem:*

The Higman-Thompson group $G_{3,1}$ with infinite generating set $\Delta_{3,1} \cup \{\tau_{0,i} : i > 0\}$, as studied in [4], could be used. This group has a finite presentation, and over this finite presentation the word problem is easy. However, over the infinite generating set $\Delta_{3,1} \cup \{\tau_{0,i} : i > 0\}$ the word problem of $G_{3,1}$ is coNP-hard. This group can be used directly to simulate circuits.

The finite symmetric group S_N could be used; here $N = 2^n$, and n is a security parameter, e.g., $n = 100$. Although this group is finite, its size is exponential in the security parameter. It is an open problem whether S_N has presentations of size linear in n. We think of S_N as acting on bit-strings of length n, hence it is natural to use elements of S_N for representing circuits.

3.1.6 Additional observations and proofs

(NP $\cap$ coNP)-complete premise problems:

We obtain an (NP $\cap$ coNP)-complete word choice problem for a finitely presented *semigroup*. For groups it is an open problem whether there are (NP $\cap$ coNP)-complete word choice problems.

Let $S_{np} = (X, R)$ be a finitely presented *semigroup* with NP-complete word problem, as constructed in [3]; this presentation was derived from any nondeterministic polynomial-time Turing machine that recognizes an NP-complete language.

Theorem 3.1.1. *The word choice problem of the finitely presented semigroup S_{np} above is an (NP $\cap$ coNP)-complete premise problem.*

Proof. Let L be any problem in NP $\cap$ coNP. Consider a nondeterministic polynomial-time Turing machine that recognizes L and consider also a nondeterministic polynomial-time Turing machine that recognizes the complement $\overline{L}$. Without loss of generality we can assume that these two Turing machines are actually the same Turing machine (let's call it M), except for the accept states: L is accepted by M using accept state q_1, and $\overline{L}$ is accepted by M using accept state q_2. In [3] the acceptance problem "does M accept a word w using accept state q_i?" (for $i = 1, 2$) is reduced to the word problem "$F(q_0 w) =_{S_{np}} F(q_i)$?"; here, q_0 is the start state of M, and F is a linear-time computable function from the words over the symbol set of M to the words over X; F is the function that reduces the decision problem of M to the word problem of S_{np}. Observe that the same word $F(q_0 w)$ is used for both L and $\overline{L}$. Therefore, $w \in L$ iff $F(q_0 w) =_{S_{np}} F(q_1)$, and $w \notin L$ iff $F(q_0 w) =_{S_{np}} F(q_2)$; hence also, $F(q_1) \neq_{S_{np}} F(q_2)$. So, F reduces the language L to the word choice problem of the semigroup S_{np}, relative to the two words $F(q_1)$ and $F(q_2)$. □

The common action problem:

Let G be a group generated by a finite set $X \subset G$ and acting faithfully (by total or partial injective or bijective maps) on the set A^* (the set of all words over a finite alphabet A). The *common action problem* problem of G (with generating set X, acting on A^*) is specified as follows:
INPUT: words $w_1, \ldots, w_n$ over $X \cup X^{-1}$;
QUESTION: does there exist $(x, y) \in A^* \times A^*$ such that for each $i = 1, \ldots, n$: $(x)w_i = y$?
The *search* version of this problem consists of outputting any such pair (x, y), rather than just finding out whether there is one.

The *circuit common action problem* is specified as follows:
INPUT: combinational circuits C_i (with I/O function $f_i : \{0,1\}^n \to \{0,1\}^n$), for $i = 1, \ldots, k$;
QUESTION: is there $(x, y) \in \{0,1\}^n \times \{0,1\}^n$ such that for each $i = 1, \ldots, k$: $f_i(x) = y$?

Theorem 3.1.2. *The common action problem for combinational circuits is NP-complete.*

Proof. We will reduce the circuit satisfiability problem (which is NP-compete) to the circuit common action problem. In the circuit satisfiability problem the input is a combinational circuit and the question is whether there is a circuit input $x \in \{0,1\}^n$ for which the circuit produces the all 1s output 1^n. A circuit C has an input x that produces the output 1^n iff the following two circuits C'_1, C'_2 have a common action pair: C'_1 on input x first uses C and then checks whether the output of C (on input x) is 1^n; if is, C'_1 outputs 1^n, otherwise C'_1 outputs 0^n. The circuit C'_2 always outputs 1^n. So, x is a satisfying input of C iff $(x, 1^n)$ is a common action pair of C'_1 and C'_2, which is iff C'_1 and C'_2 have a common action pair at all. □

We would like to reduce the common action problem of circuits to the common action problem of the group $G = \langle G_{3,1}^{\mathrm{mod}\,3}(0, 1; \#) \cup \{\kappa_{321}\}\rangle$ by using methods similar to those of [4].

However, those methods only show that the common action problem of G is NP-complete when we restrict the question to pairs (x, y) with $x \in 0\{0,1\}^* \cup 0\{0,1\}^*2$. It seems likely that the common action problem of G is NP-complete, but this remains a conjecture.

3.1.7 Summary

The general idea for a public-key cryptosystem proposed by Wagner and Magyarik in 1984, is an interesting subject for research. The original idea is too vague to be called a cryptosystem, and it is an interesting challenge to make the idea precise in such a way as to obtain a secure system. Also, the idea needs a better analysis; in particular, it is not based on the word problem (as has been claimed so far) but on the word choice problem, which is a less difficult problem and which is related to (NP $\cap$ coNP)-completeness of premise problems. It seems possible to construct public-key cryptosystems based on a combination of finite presentations and transformation groups. Such a system is described - it is based on groups related to the Higman-Thompson groups. The security evaluation of these schemes leads to interesting new complexity problems in combinatorial group theory.

3.2 NTRU and Non-Deterministic Lattice Reduction

The most efficient passive attack on the original NTRU public-key cryptosystem, which was proposed by D. Coppersmith and A. Shamir [10], is based on finding a short enough vector in an integral lattice. We[41] show that NTRU lattice possesses a cyclic automorphism group whose symmetry may be exploited. We present a method for reducing bases of NTRU integral lattices based on this symmetry. In addition to these methods, we use hill-descending techniques to combine new and proposed lattice-reduction algorithms. This approach includes deterministic and non-deterministic components which may be efficiently parallelized.

3.2.1 NTRU background

The NTRU cryptosystem was originally proposed by J. Hoffstein, J. Pipher, and J. Silverman [23]. The system was considered to be a public-key cryptosystem by its proposers even though correctly encrypted messages occasionally failed to decrypt. The point of this distinction is that security claims for NTRU have been made based on the assumption that it is indeed a public key system. Some of these claims have been questioned (see [35]).

Since its origin, NTRU has undergone several security and performance improvements. Due to its fast performance and relatively small memory requirements, NTRU is suitable for applications such as smartcards, mobile devices, and embedded technologies. As such, NTRU has been accepted to IEEE P1363 family of standards and is currently being considered for standard by the Consortium for Efficient Embedded Security (CEES).

The security of NTRU is not necessarily based on the difficulty of reducing the NTRU lattice, but lattice reduction is currently one of the best known practical attacks. It should be noted here that there is a more efficient attack introduced by Proos [35] based on exploiting NTRU decryption failures. However, such an attack requires oracle calls (for a complete description please refer to [35] and [25]).

Even though many changes to the NTRU encryption scheme have been made, the attacks described here are attacks based on the scheme introduced in [23]. There were at least two modifications proposed to enhance implementation efficiency but they may affect security, too.

3.2.2 Notation and preliminaries

For a given positive integer m, let $\mathbb{Z}_m$ denote the ring of integers modulo m. Let $\mathcal{R}$ denote the quotient ring of polynomials with integer coefficients modulo the ideal $(X^N - 1)$; i.e., $\mathcal{R} = \mathbb{Z}[X]/(X^N - 1)$, and similarly, $\mathcal{R}_m = \mathbb{Z}_m[X]/(X^N - 1)$. For $j, k \in \mathbb{Z}$, let $\mathcal{R}\{j, k\}$ denote the set of all polynomials in $\mathcal{R}$ with j coefficients equal to 1, k coefficients equal to -1, and all other coefficients equal to 0. Also the following notation will be useful:
If $x, y \in \mathbb{Z}$, then "x symod y" denotes the residue of x modulo y in the interval

$$[-\lceil \frac{y}{2} - 1\rceil, \lfloor \frac{y}{2} \rfloor].$$

A polynomial in $\mathcal{R}$ can be represented as a vector whose coordinates correspond to the polynomial coefficients. If $v = (v_0, \ldots, v_{N-1})$ and $w = (w_0, \ldots, w_{N-1})$, then denote the concatenation of v and w by (v, w).

Let $v = (v_0, \ldots, v_{N-1}), w = (w_0, \ldots, w_{N-1})$ and $c = (c_0, \ldots, c_{N-1})$ be polynomials in $\mathcal{R}$ (or $\mathcal{R}_m$) such that $c = v \cdot w$. Then, the product can be expressed as a simple matrix product $C = V \cdot W$ as follows

$$\begin{bmatrix} c_0 \\ \vdots \\ c_{N-2} \\ c_{N-1} \end{bmatrix} = \begin{bmatrix} v_0 & v_{N-1} & \cdots & v_2 & v_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ v_{N-2} & v_{N-3} & \cdots & v_0 & v_{N-1} \\ v_{N-1} & v_{N-2} & \cdots & v_1 & v_0 \end{bmatrix} \begin{bmatrix} w_0 \\ \vdots \\ w_{N-2} \\ w_{N-1} \end{bmatrix},$$

where, $C = (c_0, \ldots, c_{N-1})^T$, $W = (w_0, \ldots, w_{N-1})^T$, and V is a circulant matrix whose bottom row is $v = (v_0, \ldots, v_{N-1})$ in reverse order and each row of V is a left cyclic shift of the row below. Let V to be $V = \mathrm{cir}(v)$.

An *integral lattice* $\mathcal{L}$ is a discrete subgroup of the additive group of Euclidean space $\mathbb{R}^n$. In particular the lattice under consideration is $n-$dimensional in $\mathbb{R}^n$. Its elements are all possible integral linear combinations of vectors $\{v_1, \ldots, v_n\} \subset \mathbb{R}^n$, such that the v_i are linearly independent over $\mathbb{Z}$.

For $n > 1$, a lattice $\mathcal{L}$ of dimension n has infinitely many bases. In fact, if β is a basis of a lattice $\mathcal{L}$ given by the columns of an $n \times n$ matrix B, then the columns of B' form another basis of $\mathcal{L}$ if and only if $B' = TB$ for some unimodular integral matrix T. A unimodular matrix is a matrix of determinant ± 1.

If β and β' are two bases of the same lattice whose vectors form columns of B and B' respectively, then $|\det B| = |\det B'|$. Hence, the quantity $|\det B|$ is invariant for a lattice $\mathcal{L}$, and therefore independent of the choice of basis β. The $|\det B|$ is called the *volume* of $\mathcal{L}$, and denoted $\mathrm{vol}(\mathcal{L})$. If $\mathcal{S}$ is a sublattice of $\mathcal{L}$, it easily follows that $\mathrm{vol}(\mathcal{S}) = k \cdot \mathrm{vol}(\mathcal{L})$, for some $k \in \mathbb{Z}$.

The standard (Euclidean) norm of a vector v is denoted by $\|v\|$. Assuming $\beta = \{b_1, \ldots, b_n\}$ is a basis of a lattice $\mathcal{L}$, define the weight of β by:

$$\mathrm{wt}(\beta) = \|b_1\| \cdot \|b_2\| \cdot \ldots \cdot \|b_n\|.$$

If β and β' are bases of a lattice $\mathcal{L}$ satisfying $\mathrm{wt}(\beta') < \mathrm{wt}(\beta)$, then call β' *reduced relative to* β.

3.2.3 An outline of the NTRU algorithm

An instance of the NTRU public-key encryption scheme [23] is specified by the integer parameters (N, p, q, d_f, d_g, d_r). In [23] the proposers suggest $p = 3$, $q = 2^k$, and $N \in \{137, 251, 347, 503\}$. The parameters d_f, d_g, and d_r determine the structure of the scheme. The message space is $\mathcal{R}$ symod p.

Key generation:

Given $\mathcal{R}$, Bob chooses $f \in \mathcal{R}\{d_f, d_f - 1\}$ and $g \in \mathcal{R}\{d_g, d_g\}$. He then computes the multiplicative inverse f_p^{-1} of f in $\mathcal{R}_p$, and similarly, the inverse f_q^{-1} of f in $\mathcal{R}_q$. Finally, Bob computes the polynomial

$$h \equiv p \cdot f_q^{-1} \cdot g \pmod q \tag{1}$$

Bob's public key information is now

$$\{N, p, q, d_f, d_g, d_r, h\}$$

and his private key is f.

Encryption:

Alice selects a random polynomial $r \in \mathcal{R}\{d_r, d_r\}$ and encrypts a message m by

$$e \equiv r \cdot h + m \pmod q$$

Decryption:

In order to decrypt the received ciphertext e, Bob first computes

$$a \equiv f \cdot e \quad (\text{symod } q)$$

and then determines

$$b \equiv a \quad (\text{symod } p)$$

Finally, Bob computes

$$c \equiv f_p^{-1} \cdot b \pmod p$$

Now c should be Alice's original message m. In the rare case when $m \neq c$, a decryption failure has occurred, and therefore a notion of incorrect encryption/decryption must be introduced.

3.2.4 A lattice reduction attack on NTRU

The following attack was introduced by D. Coppersmith and A. Shamir [10]. Let B be the following $2N \times 2N$ matrix

$$B = \left[\begin{array}{c|c} I & O \\ \hline \text{cir}(h) & qI \end{array}\right] \tag{2}$$

where I is the $N \times N$ identity matrix, O the $N \times N$ zero matrix, and

$$\mathrm{cir}(h) = \begin{bmatrix} h_0 & h_{N-1} & \dots & h_2 & h_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h_{N-2} & h_{N-3} & \dots & h_0 & h_{N-1} \\ h_{N-1} & h_{N-2} & \dots & h_1 & h_0 \end{bmatrix}$$

is the circulant matrix corresponding to the public polynomial $h = h_0 + h_1 x + \cdots + h_{N-1}x^{N-1}$, defined in the equation (1).

The columns of B span a particular integral lattice $\mathcal{L}$. By the definition of B, it is easy to show that the concatenated vector $(f, pg) \in \mathcal{L}$. From the definitions, it is easily seen that $\|(f, pg)\|$ is the publicly known value $\sqrt{2(d_f + d_g p^2) - 1}$. The following classical theorem of Minkowski [19], applied to the Euclidean norm, gives an upper bound to the shortest vector in a lattice.

Theorem 3.2.1 (Minkowski's Upper Bound). *In a lattice $\mathcal{L}$ of dimension n the shortest non-zero vector v satisfies:*

$$\|v\| < c\sqrt{n}\,\sqrt[n]{\mathrm{vol}(\mathcal{L})}$$

where c is a constant.

So far, for large enough n, the best known approximation of c is 0.3196.

Since the standard choice for parameter p is 3, and since both d_f and d_g are (by definition) less than $\frac{1}{2}N$, one immediately obtains that $\|(f, pg)\| < \sqrt{10N - 1}$. By applying the Minkowski's upper bound (Theorem 3.2.1) to the NTRU lattice $\mathcal{L}$, it is clear that the shortest vector in $\mathcal{L}$ has a norm less than $c\sqrt{2N}\,\sqrt[2N]{\mathrm{vol}(\mathcal{L})} = c\sqrt{2N}\,\sqrt[2N]{q^N} = c\sqrt{2Nq} = 0.3196\sqrt{2Nq}$. Since $\|(f, pg)\|$ is normally smaller than the Minkowski's upper bound, the vector (f, pg) is likely to be a very short vector in $\mathcal{L}$.

The "Gaussian" heuristic[23] provides a method for predicting the norm of the shortest vector in a random lattice $\mathcal{L}$ of large dimension n. Specifically, the shortest vector of $\mathcal{L}$ is approximately of norm

$$\sigma = \sqrt{\frac{n}{2\pi e}}\,V^{\frac{1}{n}},$$

where $n = \dim(\mathcal{L})$ and $V = \mathrm{vol}(\mathcal{L})$. In particular, for the NTRU lattice $\mathcal{L}$ whose basis is given by the equation (2), $\dim(\mathcal{L}) = 2N$ and $\mathrm{vol}(\mathcal{L}) = q^N$ imply that

$$\sigma = \sqrt{\frac{Nq}{\pi e}}.$$

It is easily seen that $\sqrt{10N-1}$ is usually slightly less than σ, therefore the vector (f, pg) is a short vector in NTRU lattice $\mathcal{L}$.

Experimental results confirm that whenever a vector $\tau \in \mathcal{L}$ such that $\|\tau\| = \|(f, pg)\|$ was found, f could be always determined, up to rotational transformation which is discussed in the next paragraphs. Call the vector τ a *target vector.* Therefore, by reducing the basis of $\mathcal{L}$ sufficiently, an attacker may obtain the secret polynomial f if a vector of norm $\|\tau\|$ can be found in $\mathcal{L}$.

There exists yet another short vector in lattice $\mathcal{L}$ that is known in advance. For the standard parameter choices, such a vector will have smaller norm than a target vector τ. Unfortunately, this is of no advantage in finding a target vector τ.

Theorem 3.2.2. *Let* $i = (\underbrace{1, \ldots, 1}_{N}, \underbrace{0, \ldots, 0}_{N})$. *If* $\mathcal{L}$ *is an NTRU lattice of dimension* $2N$, *then* $i \in \mathcal{L}$. *Moreover, the norm of vectors* i, *and* $-i$ *is* $\sqrt{N}$.

Proof. Let B be a matrix whose columns are the basis for an NTRU lattice $\mathcal{L}$. Adding the first N columns of matrix B would produce vector i if the coefficients of h add up to 0, that is if $h(1) = 0$. However, $g \in \mathcal{R}(d_g, d_g)$ implies that $g(1) = 0$, which implies that $h(1) = pf_q^{-1}(1)g(1) = 0$. Thus, i can be expressed as an integral linear combination of the columns of B, so, $i \in \mathcal{L}$. □

Refer to vectors $\{i, -i\}$ as *trivial short vectors.* A trivial short vector is most likely the shortest non-zero vector in $\mathcal{L}$, clearly of smaller norm than Minkowski's upper bound. Hence, τ is most likely the shortest non-trivial, non-zero vector in an NTRU lattice.

A basis that contains a target vector is referred to as a *resolution basis.* The well-known LLL algorithm[30] and its various improvements due to C. P. Schnorr and others, is currently the fastest method for general basis reduction. C. P. Schnorr also introduced a BKZ method [38] which, while possibly sacrificing polynomial runtime, produces a further reduced basis. In this approach, a larger set of vectors is processed simultaneously. The cardinality of this set of vectors is called the *block size.*

3.2.5 Symmetry of the NTRU lattice

The NTRU lattice possesses a non-trivial cyclic automorphism group. This symmetry leads to a new cryptanalytic approach.

Let vector $v = (v_1, \ldots, v_{2N}) = (u, w)$, where $u = (v_1, \ldots, v_N)$ and $w = (v_{N+1}, \ldots, v_{2N})$.

Denote the cyclic right shift of a vector x by r positions with $\mathrm{rotate}_r(x)$. Then, define *birotation* of v by k positions as

$$\mathrm{birotate}_k(v) = (\mathrm{rotate}_k(u), \mathrm{rotate}_k(w)).$$

The following Theorem shows some of the nice properties of such birotations.

Theorem 3.2.3. *Let $\mathcal{L}$ be an NTRU lattice. Then $v \in \mathcal{L}$ if and only if* $\mathrm{birotate}_k(v) \in \mathcal{L}$ *for any integer k.*

Proof. Let P_k be the $N \times N$ permutation matrix that performs a cyclic shift by k positions, and let P be the following block matrix

$$P = \left[\begin{array}{c|c} P_k & O \\ \hline O & P_k \end{array}\right]$$

where O is the $N \times N$ zero matrix. It is easy to see that

$$\mathrm{birotate}_k(v) = Pv \tag{3}$$

Since $v \in \mathcal{L}$, it can be expressed as a linear combination of columns of B, where B is the $2N \times 2N$ matrix as defined in (2), that is, there exists a vector $x = (x_1, \ldots, x_{2N})$, with all integer coefficients, such that

$$v = Bx.$$

Multiplying both sides on the left by P yields

$$Pv = PBx = PBP^{-1}Px.$$

Use (3) to obtain

$$\mathrm{birotate}_k(v) = PBP^{-1} \cdot \mathrm{birotate}_k(x) \tag{4}$$

On the other hand, since

$$P_k \cdot \mathrm{cir}(h) = \mathrm{cir}(h) \cdot P_k$$

the following equality holds

$$
\begin{aligned}
PBP^{-1} &= \left[\begin{array}{c|c} P_k & O \\ \hline O & P_k \end{array}\right] \left[\begin{array}{c|c} I & O \\ \hline \operatorname{cir}(h) & qI \end{array}\right] \left[\begin{array}{c|c} P_k^{-1} & O \\ \hline O & P_k^{-1} \end{array}\right] = \\
&= \left[\begin{array}{c|c} P_k & O \\ \hline P_k \cdot \operatorname{cir}(h) & qP_k \end{array}\right] \left[\begin{array}{c|c} P_k^{-1} & O \\ \hline O & P_k^{-1} \end{array}\right] = \\
&= \left[\begin{array}{c|c} P_k P_k^{-1} & O \\ \hline P_k \cdot \operatorname{cir}(h) \cdot P_k^{-1} & qP_k P_k^{-1} \end{array}\right] = \left[\begin{array}{c|c} I & O \\ \hline \operatorname{cir}(h) & qI \end{array}\right] = B.
\end{aligned}
$$

Finally, equation (4) states that $\operatorname{birotate}_k(v) = B \cdot \operatorname{birotate}_k(x)$, that is, $\operatorname{birotate}_k(v) \in \mathcal{L}$, since it can be written as a linear combination of columns of B.

For the converse, assume $\operatorname{birotate}_k(v) \in \mathcal{L}$ and observe that

$$\operatorname{birotate}_{N-k}(\operatorname{birotate}_k(v)) = \operatorname{birotate}_N(v) = v,$$

and so by the direct statement of this proposition it follows that $v \in \mathcal{L}$. □

An elementary result is summarized in the following Theorem. The proof is trivial, following from knowledge of basic theory about lattices.

Theorem 3.2.4. *Let $\mathcal{L}$ be any integral lattice, and suppose that vectors $V = \{v_1, v_2, \ldots, v_n\}$ form a basis for $\mathcal{L}$. If $w \in \mathcal{L}$ then, for any i, $1 \leq i \leq n$, the set of vectors $V' = \{v_1, v_2, \ldots, v_{i-1}, w, v_{i+1}, \ldots, v_n\}$ forms a spanning set for a sublattice $\mathcal{L}'$ of $\mathcal{L}$. If vectors in V' are linearly dependent over $\mathbb{Z}$, then $\det V' = 0$, $\dim(\mathcal{L}') < \dim(\mathcal{L})$, and $\mathcal{L}'$ is a proper sublattice of $\mathcal{L}$. When the vectors in V' are linearly independent over $\mathbb{Z}$, then $\det V' \neq 0$, the sublattice $\mathcal{L}'$ is of the same dimension as $\mathcal{L}$, and V' forms a basis of $\mathcal{L}'$ which may or may not be a proper sublattice of $\mathcal{L}$. In fact, $\mathcal{L}' = \mathcal{L}$ if and only if $\operatorname{vol}(\mathcal{L}') = \operatorname{vol}(\mathcal{L})$.*

The implementation of the Theorem 3.2.3 is a core component of the proposed NTRU lattice reduction technique. That is, the vector of largest norm in the basis B can be replaced by a birotation of any vector of smaller norm. If this new set of vectors spans the same lattice, then this basis is reduced relative to B. The algorithm in Table 1 illustrates this process.

Input: $2N \times 2N$ matrix B (a basis of an NTRU lattice) with columns $b_1, \dots b_{2N}$ corresponding to the basis vectors, and $m, n \in \mathbb{Z}$, such that $\|b_n\| < \|b_m\|$

Output: $2N \times 2N$ matrix B' with weight less then the weight of B

(* BIROT)

```
$B'' \leftarrow B$
$i \leftarrow 1$
$a \leftarrow \det(B)$
START:
$n^{\text{th}}$ column of $B'' \leftarrow \text{birotate}_i(m^{\text{th}}$ column of $B'')$
$b \leftarrow \det(B'')$
if $\frac{b}{a} = \pm 1$ goto END
$i \leftarrow i + 1$
if $i < N$ goto START
$B' \leftarrow B$
TERMINATE
END:
$B' \leftarrow B''$
TERMINATE
```

Table 1: BIROT algorithm

3.2.6 A hill-descending approach

It is known that the LLL algorithm is sensitive to the order in which the basis vectors are presented. A systematic selection of permutations of the basis vectors, along with the birotation reduction are the building blocks of a hill-descending approach.

Let $\mathcal{L}$ be an integral lattice and $\mathcal{B}$ the collection of all bases for $\mathcal{L}$. Then, $\mathcal{B}$ is an orbit under the action of the integral unimodular group $SL(n, \mathbb{Z})$. That is, if B is a particular basis for $\mathcal{L}$, then $\mathcal{B} = \{BT \mid T \in SL(n, \mathbb{Z})\}$. Now, the symmetric group S_n is a subgroup of $SL(n, \mathbb{Z})$, where permutation $\pi \in S_n$ can be viewed as $n \times n$ permutation matrix.

Define an *objective function* $\varphi : \mathcal{B} \longrightarrow \mathbb{R}$ to be $\varphi(B) = \|b\| / \|\tau\|$ for all $B \in \mathcal{B}$, where b is a shortest non-trivial vector in B, τ a target vector.

The hill-descending approach implements a walk $B_0 \rightarrow B_1 \rightarrow \dots \rightarrow B_r$ in the space of bases of the lattice $\mathcal{L}$, where $\varphi(B_i) \leq \varphi(B_{i+1})$. Here, B_0 is the basis of $\mathcal{L}$ defined in (2), and success is achieved if B_r is a resolution basis; i.e., if $\varphi(B_r) = 1$.

Ordering of the basis vectors:

Let $B = \{b_1, \ldots, b_{2N}\}$ be a basis for NTRU lattice $\mathcal{L}$, and let $\pi_0 \in S_{2N}$ be the identity permutation on $2N$ letters. Define the *distance* between two permutations $\alpha, \beta \in S_{2N}$ to be $\mathrm{d}(\alpha, \beta) = k$, if the two permutations differ in exactly k positions. Let

$$B(\alpha, k) = \{\, \beta \in S_{2N} \,|\, \mathrm{d}(\alpha, \beta) = k \,\}$$

denote the family of all permutations of distance k from a fixed permutation α. It follows that

$$|B(\alpha, k)| = \binom{n}{k} D_k$$

where D_k is the number of derangements on k letters. It is easy to see that $B(\alpha, 0) = \{\alpha\}$ and $|B(\alpha, 1)| = 0$.

In the hill-descending algorithm, a random sample of permutations from $B(\pi_0, k)$, for a fixed value of k is selected. These permutations are then applied to the ordering of the basis vectors in B. An application of LLL for each permuted basis B' yields a value of the objective function $\varphi(B')$. In practice, a basis B is represented in matrix form, and permuting the basis vectors means permuting the columns of the matrix.

The algorithm:

Following is a description of a Las Vegas type method for resolving NTRU lattices. The algorithm is based on a combination of BKZ-LLL and BIROT primitives, and can be implemented in parallel.

The algorithm requires input parameters L and t, where L is an NTRU lattice basis and t is the norm of a target vector τ, i.e. $t = \sqrt{2(d_f + d_g p^2) - 1}$. In the first stage, a BKZ-LLL with blocksize $s = 2$ is applied to L to obtain an initial reduction B. Blocksize 2 guarantees that the execution time will be polynomial. Next, basis B undergoes a loop of M parallel processes. At each PU$_i$ (Processing Unit i) the columns of B are permuted according to a random permutation $\alpha \in B(\pi_0, k)$. Such a permuted basis is supplied to a local BKZ-LLL primitive, resulting in basis B'_i. At the main PU, the algorithm then examines the bases and selects among the B'_i, the basis B_{min} with minimal $\varphi(B'_i)$. If B_{min} is reduced relative to B, the algorithm loops back to the parallel stage, setting the input basis B to B_{min}. In case that $\varphi(B_{min}) \geq \varphi(B)$, the distance k is incremented by 1 before looping back to the parallel stage. However, when k reaches the maximum value of $2N + 1$, a BIROT routine is performed on B in order to escape the local minimum of the hill-descending approach. Following the BIROT, the algorithm resets the k value

```
Input: $L$ an NTRU lattice basis, $t$ the norm of target vector
Output: resolution basis of $L$
(* MASTER)
 $s \leftarrow 2$ (blocksize)
 $k \leftarrow 2$ (distance)
 $B \leftarrow L$
 START:
 for each processor $j$, $1 \leq j \leq M$, perform (* SLAVE $j$)
 select $B_{min}$ such that $\varphi(B_{min}) = \min\{\varphi(B_i) \,|\, 1 \leq i \leq M\}$
 if $v \in B_{min}$ such that $\|v\| = t$ then
     TERMINATE
 if $\varphi(B_{min}) < \varphi(B)$ then
     $B \leftarrow B_{min}$
     goto START
 $k \leftarrow k + 1$
 if $k \leq 2N$ then
     goto START
 $B_{birot} \leftarrow \mathrm{BIROT}(B_{min})$
 if $\mathrm{wt}(B_{birot}) < \mathrm{wt}(B_{min})$ then
     $B \leftarrow B_{birot}$
     $k \leftarrow 2$
     goto START
 $s \leftarrow s + 1$
 goto START
```

Table 2: Parallel hill-descending algorithm - MASTER

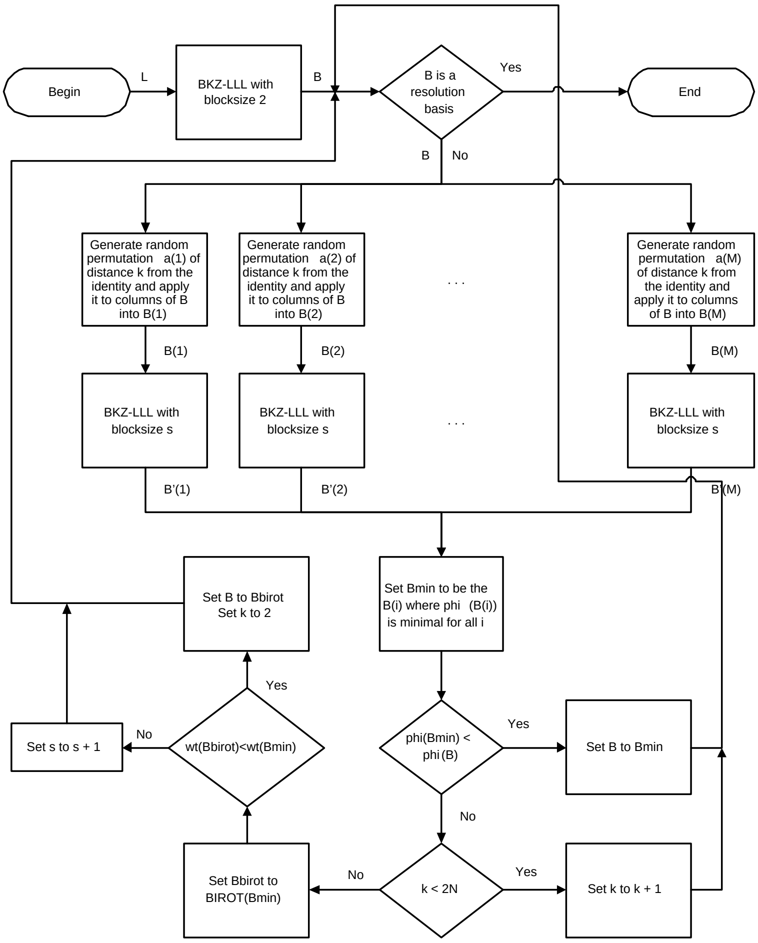
back to 2. In the case that BIROT does not result in further reduction, the blocksize s is increased by 1.

The algorithm is expected to run until it produces a resolved basis for the lattice $\mathcal{L}$ spanned by L. This is a non-deterministic, Las Vegas type algorithm, and it is not expected to invariably produce desired results. Its degree of success depends on the particular lattice $\mathcal{L}$.

The hill-descending (Las Vegas type) algorithm appears in Tables 2 and 3, and its schematic diagram is shown in Figure 1.

```
Input: $B_j$ a lattice basis, distance $k$, block size $s$
Output: reduced basis of $B_j$
(* SLAVE $j$)
  randomly select $\alpha \in B(\pi_0, k)$ where $\pi_0 \in S_{2N}$ is the identity permutation
  apply $\alpha$ to the order of vectors in basis $B_j$
  $B'_j \leftarrow \mathrm{BKZ-LLL}_s(B_j)$
  return $B'_j$
```

Table 3: Parallel hill-descending algorithm - SLAVE



Begin
L
BKZ-LLL with blocksize 2
B
B is a resolution basis
Yes
End
B
No
Generate random permutation a(1) of distance k from the identity and apply it to columns of B into B(1)
Generate random permutation a(2) of distance k from the identity and apply it to columns of B into B(2)
. . .
Generate random permutation a(M) of distance k from the identity and apply it to columns of B into B(M)
B(1)
B(2)
B(M)
BKZ-LLL with blocksize s
BKZ-LLL with blocksize s
. . .
BKZ-LLL with blocksize s
B'(1)
B'(2)
B'(M)
Set B to Bbirot Set k to 2
Set Bmin to be the B(i) where phi (B(i)) is minimal for all i
Yes
Yes
Set s to s + 1
No
wt(Bbirot)<wt(Bmin)
phi(Bmin) < phi(B)
Set B to Bmin
No
No
Yes
Set Bbirot to BIROT(Bmin)
k < 2N
Set k to k + 1

Figure 1: Scheme for the parallel hill-descending algorithm

3.2.7 Summary

A method for exploiting the symmetry of an NTRU lattice was introduced. This method replaces large vectors in the basis with birotated vectors of smaller norm. Alone, it leads to reduced bases, but was significantly improved through parallel processing. By using existing well-known lattice reduction techniques, in conjunction with the birotation method, a new hill-descending approach introduces a walk through the space of bases which ultimately leads to faster resolution when compared to existing lattice reduction methods. Furthermore, performance is enhanced through the parallelization of critical components, as introduced in the proposed algorithm. Although the hill-descending approach is more effective than the previously explored approaches, its non-deterministic nature implies variable performance and does not guarantee resolution.

4 Private-Key Cryptosystems

The second major part of this dissertation deals with cryptanalysis of two private-key cryptosystems. All analyzed cryptosystems are block ciphers.

Many hard mathematical problems as well as some exotic problems from other fields can be in some way turned into a cryptosystem. The block cipher based on the Hopfield neural network is a typical example. A necessary introduction to neural networks is provided, then the block cipher is described, criticized, and cryptanalyzed.

Multimedia security is the underlying theme of the cryptanalysis of the remaining symmetric cryptosystems. Three recent cryptosystems that were proposed to ensure privacy of video streams were turned out to be fast but too simple to withstand basic attacks.

4.1 A Block Cipher based on the Hopfield Neural Network

In a Hopfield neural network, an input message converges to one of the special messages called attractors. It was shown that an overstoraged Hopfield neural network exhibits stochastic error in convergence. In particular, the messages in the attraction domain of an attractor are unpredictably related to each other.

Based on these facts, D. Guo, L.M.Cheng, and L.L.Cheng proposed a block cipher[20]. We[11] examine the security and efficiency of the proposed cryptosystem. Furthermore, the cryptanalysis leads to an interesting mathematical problem: Given two matrices that are conjugate of each other by a permutation matrix, determine this permutation matrix. This section contains a solution to this problem, too.

4.1.1 Hopfield neural network

Firstly, consider a fully interconnected *neural network* of N neurons (labeled $0, 1, \ldots, N-1$). The state of a neuron i at a time t is denoted $S_i(t)$, the initial state is denoted $S_i(0)$.

The next state of neuron i depends on the current states of all neurons as follows

$$S_i(t+1) = f\left(\sum_{j=0}^{N-1} T_{ij} S_j(t) + \vartheta_i \right),$$

where T_{ij} is the synaptic strength between neurons i and j, ϑ_i is the threshold value of neuron i, and $f(\cdot)$ is any non-linear function.

The *Hopfield Neural Network* (HNN) is a neural network with zero neuron threshold (i.e. $\vartheta_i = 0$ for every $i = 1, 2, \ldots, N-1$), and with $T = (T_{ij})$ being a symmetric matrix. J.J.Hopfield proved[24] that the energy function

$$E(t) = -\frac{1}{2} \sum_{i,j} T_{ij} S_i(t) S_j(t)$$

of such network is bounded during state evolution, therefore each initial state of the network must converge to a stable state, which is a local minima of $E(t)$. Call stable states *attractors.*

From now on, consider the Hopfield neural network to be *discrete* (i.e. $S_i(t) \in \{0,1\}$) and *clipped* (i.e. $T_{ij} \in \{-1, 0, 1\}$). The non-linear function f will be the sign function $\sigma(\cdot)$ where

$$\sigma(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} .$$

The state of the network at time t can be expressed as a row vector $S(t) = (S_0(t), S_1(t), \ldots, S_{N-1}(t))$. Call an initial state $S(0)$ a *message.* Let the function $\sigma(\cdot)$ acts on vectors elementwise, then the next state formula of a Hopfield neural network can be expressed in matrix form as follows:

$$S(t+1) = \sigma(S(t)T).$$

Call the HNN$_T$ network *overstoraged* if the number of attractors of this network is $2N$ or more. Overstoraged HNNs exhibit an interesting property of converging in few iterations. Specifically, the larger the number of attractors the more likely it is that a random input message to the network will cause it to converge in a single step, namely

$$S(t) = S(t+1) \quad \text{for} \quad t \geq 1.$$

This property is likely due to the large number of attractor domains spread out over the entire message space, causing a random input message to be close (in the terms of Hamming distance) to its attractor. The networks with the large number of attractors or nearly equidistant attractors are more likely to converge in a single step for any input.

Neural networks are known to have the non-linear mapping property. In addition, the HNN$_T$ model possesses an interesting chaotic property. In particular, the relation of any initial state $S(0)$ to its attractor (stable state) is irregular-chaotic[20]. For the maximum irregularity, the network requires equal concentrations of excitatory ($T_{ij} = 1$) and inhibitory ($T_{ij} = -1$) synapses. That is,

$$\sum_i T_{ij} = 0 \quad \text{and} \quad \sum_j T_{ij} = 0. \tag{5}$$

With chaos comes also order. The following property can be easily proved: Let x be any attractor of HNN$_T$, and let $A_x = \{y \mid x = \sigma(yT)\}$ be the domain of attraction for x. Moreover, let P be an $N \times N$ permutation matrix, and let $\hat{T} = PTP^{-1}$. Then xP^{-1} is an attractor of HNN$_{\hat{T}}$ and $\{yP^{-1} \mid y \in A_x\}$ is its domain of attraction.

Remains to say that HNN$_T$ neural network model (also called associative memory network) is suitable for fast implementations - implementation and execution can be performed in parallel.

4.1.2 The block cipher

Following is a brief description of the *Symmetric Probabilistic Encryption Scheme Based on the Chaotic Attractors*[20]. The description here is somewhat simplified in order to focus on the important parts of the cryptosystem. See the original proposal for the full details.

Fix N and fix an $N \times N$ matrix T over $\{-1, 0, 1\}$ such that #”1” = #”-1” ≈ #”0” ≈ $N/3$ (where # means ”the number of”) in each row and each column of T. The matrix T must be selected in such a way that the resulting Hopfield neural network HNN$_T$ with synaptic strength matrix T and sign function $\sigma(\cdot)$ will be overstoraged. The matrix T and $\sigma(\cdot)$ are public information.

Key generation: Choose an $N \times N$ permutation matrix H and compute new synaptic matrix $\hat{T} = HT\widetilde{H}$, where $\widetilde{H}$ denotes the transpose of H. Note that since H is a permutation matrix, $\widetilde{H} = H^{-1}$. Keyspace is of size $N!$.

Plaintext space: A subset of the attractors of the HNN$_{\hat{T}}$ network. Each attractor is a binary vector of length N.

Encryption: For a given plaintext x, randomly choose a binary vector y of length N from the domain of attraction of x. The vector y will become the ciphertext. Since there are many ciphertexts corresponding to one plaintext and the selection is random, the encryption is probabilistic.

Decryption: For a given ciphertext y (a binary vector of length N), set the initial state of $\mathrm{HNN}_{\hat{T}}$ to be the given ciphertext. Then the $\mathrm{HNN}_{\hat{T}}$ network will converge to the corresponding plaintext x. That is, in the network $\hat{S}(t+1) = \sigma(\hat{S}(t)HT\widetilde{H})$ with initial state $\hat{S}(0) = y$, as $t \to \infty$, $\hat{S}(t) \to x$.

4.1.3 Critique, efficiency, and attacks

1. For the computational security of the cryptosystem, only those attractors of $\mathrm{HNN}_{\hat{T}}$ with large domain of attraction should be considered. The plaintext space is therefore of size less than or equal to $2N$. One immediate problem is to determine which attractor has a domain of attraction large enough such that computational searches over this domain are infeasible. By a property from a previous section, there is one-to-one correspondence between domains of attraction of HNN_T and $\mathrm{HNN}_{\hat{T}}$. Hence if the cardinalities of domains of attraction of HNN_T are known in advance (precomputed), this problem is negligible. In other words, the problem of the cardinalities of domains of attractions is the same for HNN_T and $\mathrm{HNN}_{\hat{T}}$.

2. Another immediate problem is a realization of an encryption. For a given plaintext (attractor) x, one needs to find y in the domain of attraction of x. This can be accomplished in two ways. Either the domains of attraction are tablularized or the suitable ciphertext will be determined by random searching through the whole binary vector space of dimension N. The first approach requires huge storage and extensive pre-computations, and therefore is not practical for reasonably secure values of N. The second approach works as follows: For a given x, repeatedly and randomly choose a binary vector y of length N and test whether y converges to x in $\mathrm{HNN}_{\hat{T}}$. Thus a single encryption requires several decryptions, making the encryption process too slow. The number of decryptions while encrypting depends on the number of plaintexts k and the sizes of the domains of attraction.

3. Chosen-ciphertext attack: In this scenario, an attacker should be able to choose N ciphertext messages such that they converge to the attractor in just one step and when put as rows to a matrix I, the matrix will be an identity matrix. The attacker should put the corresponding plaintexts (as rows) to matrix V.

   Mathematically speaking, let e_i denote the N-bit vector with 1 at position i and 0's at all other positions. Suppose that every ciphertext e_i $(i = 1, \ldots, N)$ converges to an attractor in just one iteration and for every such ciphertext e_i the attacker can obtain corresponding plaintext v_i. That is, $v_i = \sigma(e_i HT\widetilde{H})$ for every $i = 1, \ldots, N$. Let $V = (v_i)$, $I = (e_i)$, and let $\sigma(\cdot)$ acts on a matrix elementwise. Then

$$V = \sigma(IHT\widetilde{H}).$$

Since $\widetilde{H} = H^{-1}$ and both H and H^{-1} are permutation matrices, one obtains the equation

$$V = \sigma(IHT\widetilde{H}) = \sigma(HTH^{-1}) = H\sigma(T)H^{-1}, \tag{6}$$

where V is known and $\sigma(T)$ can be easily computed since T is known. In the next section, it is shown how this equation can be solved for an unknown permutation matrix H. Since the solution - the permutation matrix H does not have to be unique, the attacker either obtains the secret key or its equivalence.

4.1.4 Conjugation by a permutation matrix problem

Definition. Let $\mathcal{P}$ be the set of all permutation matrices of the size $N \times N$. Let A, B be two abstract matrices such that there exists $P \in \mathcal{P}$ and

$$B = PAP^{-1}, \tag{7}$$

is valid. We will call the equation (7) a *conjugation by a permutation matrix problem.* For brevity, we refer to it as the *PROBLEM.*

For the record, the problem of finding permutation matrices P and Q in the equation

$$B = PAQ,$$

is equivalent to the well-studied *graph isomorphism problem*, where A and B represent the corresponding adjacency matrices of some graphs. So far, there is no polynomial-time algorithm for solving the graph isomorphism problem, and no one has proven that the problem is $\mathcal{NP}$-complete. In fact, it is believed that the problem is probably neither in $\mathcal{P}$ nor in $\mathcal{NP}$-complete classes of problems[27].

Notation. *A natural map from the permutation group S_N to the group of permutation matrices is the following one: Let $\pi \in S_N$. Then, the image of π in the group of permutation matrices is the following $N \times N$ matrix P_π:*

$$P_\pi = \begin{pmatrix} e_{\pi(1)} \\ \vdots \\ e_{\pi(N)} \end{pmatrix} = \begin{pmatrix} e_{\pi^{-1}(1)} & \cdots & e_{\pi^{-1}(N)} \end{pmatrix},$$

where e_i represents the i-th row (or column) of the identity matrix. Note that this implies that the matrix P_π^T is the following $N \times N$ matrix:

$$P_\pi^T = \begin{pmatrix} e_{\pi^{-1}(1)} \\ \vdots \\ e_{\pi^{-1}(N)} \end{pmatrix} = \begin{pmatrix} e_{\pi(1)} & \cdots & e_{\pi(N)} \end{pmatrix} = P_\pi^{-1} = P_{\pi^{-1}}.$$

For the simplicity of notation, we write P instead of P_π.

Theorem 4.1.1. *Let π be a permutation in S_N from which a permutation matrix P is constructed via the natural map. If the matrix A in the equation (7) contains an element at position (i, j), then this element is moved in the matrix B to the position $(\pi^{-1}(i), \pi^{-1}(j))$.*

Proof. Trivial. □

Corollary 4.1.2. *Let $a_{(i_1,j_1)}, \ldots, a_{(i_m,j_m)}$ be unique elements of the matrix A (and consequently B) from the equation (7). If the set, consisting of the unique elements of the multiset $\{i_1, j_1, \ldots, i_m, j_m\}$, is equivalent to the set $\{1, \ldots, N\}$, then all the moves of π are known.*

Proof. Let $a_{(i_1,j_1)}, \ldots, a_{(i_m,j_m)}$ be unique elements of A. Then, by the Theorem 4.1.1, $a_{(i_k,j_k)}$ is moved from position (i_k, j_k) in the matrix A to the position $(\pi^{-1}(i_k), \pi^{-1}(j_k))$ in the matrix B, for all $k \in \{1, \ldots, m\}$. Since the element $a_{(i_k,j_k)}$ is unique in both A and B, then the values $i_k, j_k, \pi^{-1}(i_k)$, and $\pi^{-1}(j_k)$ are all known. Thus, if the values $1, \ldots, N$ are in the multiset $\{i_1, j_1, \ldots, i_m, j_m\}$, then $\pi^{-1}(1), \ldots, \pi^{-1}(N)$ are known. □

On the other hand, if the matrix A (and consequently B) does not contain unique elements (as is the case with equation (6), where V and $\sigma(T)$ are matrices over $\{0, 1\}$), then the matrix A^n (and consequently B^n) for some positive integer n likely does. Here, note that

$$B^n = (PAP^{-1})^n = PA^nP^{-1}.$$

Use this "powering" method to obtain unique elements in the matrix A^n and apply Theorem 4.1.1 and Corollary 4.1.2 to obtain the unknown moves of the permutation π from which the matrix P is constructed. The remaining (possibly very few or none) moves can be obtained by exhaustive search. A reader should note that the "powering" method significantly narrows down the exhaustive search space, even if the matrix A has strong symmetry (e.g. circulant matrix).

It should be noted here that powering both sides of the equation (7) may produce an expanded solution space. In other words, a solution to the PROBLEM is also a solution to $B^n = PA^nP^T$, but not necessarily vice versa. Once a solution is found to the powered equation, the following helps determine all the solutions to the powered equation, from which all the solutions to the PROBLEM can be obtained.

Firstly, suppose that T is a circulant matrix (as suggested in [20]).

Notation. *If Q is a permutation matrix, let $C(Q)$ denote the centralizer of Q over the group of permutation matrices. More generally, if A is an $n \times n$ matrix, let $C(A)$ denote the set of all $n \times n$ permutation matrices that commute with A.*

Theorem 4.1.3. *If a permutation matrix Q is a solution to the equation (7), then the matrix QR is also a solution, for all $R \in C(A)$.*

Proof. Let Q be a solution to the equation (7), i.e., $B = QAQ^{-1}$. Let R be any element from $C(A)$. Then, $AR = RA \Rightarrow A = RAR^{-1}$. Therefore, QR is also a solution to (7) since

$$B = QAQ^{-1} = QRAR^{-1}Q^{-1} = (QR)A(QR)^{-1}.$$

□

Theorem 4.1.4. *Let $\sigma(T)$ be a circulant matrix, and U be a permutation matrix corresponding to the permutation cycle $(1\,2\,\ldots\,N)$ of length n. If a permutation matrix Q commutes with U, then Q commutes with $\sigma(T)$.*

Proof. By a result from elementary linear algebra[12], if $\sigma(T)$ is a right circulant matrix with the first row $c_0, c_1, \ldots, c_{N-1}$, then

$$\sigma(T) = \sum_{i=0}^{N-1} c_i U^i.$$

Take a permutation matrix $Q \in C(U)$. Then, Q commutes with $\langle U \rangle$ (a cyclic group generated by U). Therefore, Q commutes with $\sigma(T)$ since

$$Q\sigma(T) = c_0 QU^0 + \ldots + c_{N-1}QU^{N-1} = c_0U^0Q + \ldots + c_{N-1}U^{N-1}Q = \sigma(T)Q.$$

□

By the Theorems 4.1.3 and 4.1.4, if a single solution P to the equation (7) is obtained when A is circulant, then N distinct solutions are immediately available, namely PQ where $Q \in \langle U \rangle$.

More generally, suppose T is an arbitrary matrix, not necessary circulant, but still conforming to the rules in the equation (5). In this case, similar, but more complex approach could be used.

Notation. *A doubly stochastic matrix is a matrix over real numbers in which all the entries are non-negative and each row and each column adds to exactly 1.*

A classical theorem of Birkhoff[8] follows:

Theorem 4.1.5 (Birkhoff). *A matrix A over a field is doubly stochastic if and only if it is a convex combination of permutation matrices. That is, A is doubly stochastic if and only if $A = f_1R_1 + f_2R_2 + ... + f_kR_k$ for some permutation matrices R_i and some nonnegative real numbers f_i whose sum equals to 1.*

Theorem 4.1.6. *Let T be a matrix with row sums and column sums equal to 0. Denote by m the sum of any row of $\sigma(T)$ and let $M = \frac{1}{m}\sigma(T) = f_1R_1 + f_2R_2 + ... + f_kR_k$. If a permutation matrix Q commutes with every R_i $(i = 1, \ldots, k)$, then Q commutes with $\sigma(T)$. Therefore, if P_0 is a solution to the equation (6) then any $P \in (C(R_1) \cap C(R_2) \cap \ldots \cap C(R_k))P_0$ is also a solution.*

Proof. The matrix T conforms to the equations in (5). It follows, by the definition of $\sigma(\cdot)$, that all row sums and all column sums of $\sigma(T)$ are the same, say m. Then the matrix $M = \frac{1}{m}\sigma(T)$ is doubly stochastic. Thus by the Theorem 4.1.5, $M = f_1R_1 + f_2R_2 + \ldots + f_kR_k$ as stated. Now, suppose a permutation matrix Q commutes with all R_i $(i = 1, \ldots, k)$. Then, Q commutes with M since

$$QM = f_1QR_1 + \ldots + f_kQR_k = f_1R_1Q + \ldots + f_kR_kQ = MQ,$$

and therefore Q commutes with $mM = \sigma(T)$. Finally, if Q commutes with all R_i $(i = 1, \ldots, k)$ then Q is in the intersection of the centralizers of R_i's. □

By the Theorem 4.1.6, if one obtains a solution to the equation (7), as well as the intersection of the centralizers of the permutation matrices from the Birkhoff's decomposition from the Theorem 4.1.5, one has obtained t solutions to the equation (7), where t denotes the cardinality of the intersection. The Birkhoff's decomposition time-complexity is $O(N^{4.5})$, assuming the improved Birkhoff-von Neumann algorithm by Dulmage and Halperin[14]. Once a decomposition is obtained, the centralizers can be calculated using polynomial-time approaches such as the ones by Sims[43] or Buttler[9] that have time-complexities of about $O(N^{3.5})$. Even though generally the intersection of subgroups problem is considered intractable in the sense that there is no polynomial-time algorithm in N, this problem is solvable for the relatively small permutation groups that are suggested for use in [20].

4.1.5 Summary

Although the characteristics and phenomenons of Hopfield neural networks are interesting, they do not always lead to straightforward cryptographic applications.

The 1999 paper of Guo-Cheng-Cheng that proposes the use of specific Hopfield neural network as a basis for a symmetric block cipher is examined. It turns out that this approach, although worthwhile studying, is impractical in the terms of the speed of computation and communication. The cryptanalysis of various security aspects of this cipher revealed a few severe weaknesses. In particular, the cipher is vulnerable to ciphertext-only and chosen-ciphertext attack. And thus the analyzed cipher should not be used in applications where such attacks are feasible.

The chosen-ciphertext attack requires determining a hidden permutation matrix at some point. A solution for this *conjugation by a permutation matrix problem* is proposed. Namely, practical methods to obtain a set of permutation matrices that are solution to the mentioned problem are described.

4.2 Video Encryption Algorithms

Content security is an important issue in multimedia applications. In this section we[40] perform a cryptanalysis of several encryption algorithms that have been proposed to protect the privacy of MPEG video streams. In particular, we analyze the encryption based on permuting the Huffman codeword list, and two selective encryption algorithms - VEA and MVEA. Firstly, the MPEG-1 video encoding is described in the terms necessary for understanding the encryption algorithms. All analyzed algorithms and their characteristics are included. The cryptanalysis of some of the proposed MPEG-1 video encryption algorithms then follows.

4.2.1 Multimedia security

Multimedia content is a combination of any of the following media: text, still images, audio, animation, and video. Multimedia security deals with ways of protecting such content. In general, this is achieved by methods that are heavily based on cryptography.

These methods enable communication security, piracy protection (so called Digital Rights Management), or both.

Communication security of multimedia content can be accomplished by means of standard symmetric-key cryptography. In particular, viewing multimedia content as a sequence of binary data, protection can be thought of as applying conventional symmetric-key encryption techniques (such as AES) to the whole sequence. This method is referred to as a *naive algorithm*. Unfortunately, due to a variety of constraints, applying the naive approach to more complex multimedia streams (mostly video and audio) creates significant computational overhead.

Communication encryption of video and audio content is therefore harder to accomplish. It involves a careful analysis to determine and identify the optimal encryption method when dealing with audio and video content. Current research is focused on modifying and optimizing the existing cryptosystems for real-time audio and video content. It is also oriented towards exploiting the specific properties of many standard video and audio formats, in order to achieve desired speed and enable real-time streaming. This is referred to as *selective encryption*[34].

The challenges of video encryption come from several facts. First, the size of a typical MPEG compressed video file is often very large (for example, the size of a two hour MPEG-1 video is about 1 GB). Second, the decoding (as well as decryption) needs to be processed in real time (for example, the MPEG-2 video streams can be as large as 40 Mbps). Third, VCR-like functions such as fast forward or playback from any point should be available and reasonably fast. In addition, there is more to video security than just encryption. The already mentioned Digital Rights Management is an example of a security requirement that goes beyond communication security.

Protecting multimedia content is a very important issue. On one hand, protection by a naive algorithm provides maximum security, but requires a special, costly hardware for real-time decryption. On the other hand, most of the multimedia applications need to balance between security and cost of video streaming. So there is an apparent tradeoff between security and the speed of streaming. The goal is to design a reasonably fast and secure encryption method such that breaking this method requires an investment several times higher than the value of the content.

Selective encryption is only one part of communication security. It is, however, a crucial part, and thus studying these kind of techniques is extremely important.

4.2.2 MPEG-1 video encoding

While a detailed description of the MPEG-1 video encoding is not provided, the following paragraphs outline the core components of this process.

The MPEG-1 video encoding[15] scheme represents the video signal using the repetition of *group of pictures* (GOPs). Each GOP is a sequence of selected I, P and B frames. Typical GOP sequences are IBBBPBBB or IBBPBBPBB, but the relative frequency of I, P and B frames may be application dependent. I frames are encoded as standard JPEGs, without reference to other frames. Consequently, I frames are of the smallest compression ratio. A P frame is encoded with reference to the previous I frame, containing only the difference between these two frames. Since the time difference between these two frames is a fraction of a second, the difference between blocks of pixels is very small. Therefore, P frames have a much better average compression ratio than the I frames. Finally, B frames are bidirectionally interpolated using the previous closest I/P frame and the following closest I/P frame. The average compression ratio of B frames is usually the highest.

The encoding of I frames differs from that of the P and B frames. An I frame is encoded as a standard JPEG still image. JPEG encoding is a complex process. The JPEG encoding[26] is a lossy type of compression, a tradeoff between quality of the image and compression ratio. For our purposes, it is sufficient to know that an image is first divided into blocks of 8×8 pixels. All JPEG encoding operations are then performed on these blocks. The encoding starts with a Discrete Cosine Transformation (DCT), which computes the so-called DC coefficient - the coefficient containing the crucial information of the whole block, and 63 so-called AC coefficients. The next stages are quantization, the lossy stage, and zig-zag sequencing. Finally, the block undergoes entropy encoding using the Huffman code, where the codeword list is fixed.

4.2.3 VEA and MVEA algorithms

VEA and MVEA are MPEG-1 video encryption algorithms that were introduced in [2]. Provided is a description of the algorithms, followed by the cryptanalytic results.

Table 4 and Table 5 present the simplified pseudocode of the VEA and MVEA algorithms, respectively. From the pseudocode, it is clear that VEA and MVEA are almost the same. The only difference is the set of video stream bits that are encrypted. In both cases, the algorithm works with macro blocks of size 16×16 pixels subsampled into four 8×8 blocks Y representing luminescence, and two chrominance 8×8 blocks, Cr and Cb. For a more detailed description, refer to [2].

```
Input: key $k$ of length $m$ bits, video bit-stream $v$ of length $n$
Output: encrypted video bit-stream $w$
(* VEA encryption)
 for every bit $i = 1, \ldots, n$ in video bit-stream $v$ do
        1. if $v_i$ is a beginning of GOP, then $j \leftarrow 0$
        2. else if $v_i$ is a sign bit of DC coefficient or a sign bit of DC
           differential value of Y, Cr, or Cb block of I frame, then
           (a) $w_i \leftarrow v_i \oplus k_j$
           (b) $j \leftarrow j + 1 \pmod{m}$
        3. else $w_i \leftarrow v_i$
```

Table 4: VEA encryption

As mentioned before, there is no need to to encrypt the video bit by bit. The VEA and MVEA algorithms take advantage of this concept. That is, they perform encryption on predetermined (fixed) bits of the video stream, and therefore fall into the category of selective encryption algorithms.

One of the most significant properties of VEA and MVEA from [2] is the following: One can encrypt a MPEG-1 video many times, and decrypt it in one step. Let $E_k(P)$ denote the VEA or MVEA encryption of plaintext P using key k. For any two distinct keys k_1 and k_2, it holds that

$$E_{k_1}(E_{k_2}(P)) = E_{k_1 \oplus k_2}(P). \tag{8}$$

Therefore the decryption can be performed in one step using the key $k_3 = k_1 \oplus k_2$.

This fact is very useful in the case when a key needs to be changed. The change is cost-effective since the decryption process can be eliminated, unlike in other cases, where decryption must be performed before obtaining new ciphertext. In our analysis of VEA and MVEA, we show that this key-change has no impact on the security.

4.2.4 Observations and flaws

Let P denote the collection of compressed video bits for a single GOP that needs to be protected. P can be represented as $P = (p_0,\ p_1,\ \ldots,\ p_{t-1})$, where, in the case of VEA, p_i for $i = 0, \ldots, t-1$ are all the sign bits of the DC coefficients, and all the sign bits of the discrete cosine differential value of a Y, Cr, or Cb block of an I frame, in their

```
Input: key $k$ of length $m$ bits, video bit-stream $v$ of length $n$
Output: encrypted video bit-stream $w$
(* MVEA encryption)
 for every bit $i = 1, \ldots, n$ in video bit-stream $v$ do
     1. if $v_i$ is a beginning of GOP, then $j \leftarrow 0$
     2. else if $v_i$ is a sign bit of DC differential value of Y, Cr, or Cb
        block of I frame, then
         (a) $w_i \leftarrow v_i \oplus k_j$
         (b) $j \leftarrow j + 1 \pmod m$
     3. else if $v_i$ is a sign bit of differential value of motion vectors of
        B or P frame, then block of I frame, then
         (a) $w_i \leftarrow v_i \oplus k_j$
         (b) $j \leftarrow j + 1 \pmod m$
     4. else $w_i \leftarrow v_i$
```

Table 5: MVEA encryption

original order. When considering MVEA, the p_i's are all the sign bits of the discrete cosine differential values of a Y, Cr, or Cb block of an I frame, and all the sign bits of the differential values of the motion vectors of B and P frames, in their original order. The authors of the original algorithms give an analytical explanation as to why encrypting only these bits of the video stream leads to sufficient protection.

Let k be a key of length m bits. Again, k can be represented as a finite bit-stream $k = (k_0,\ k_1,\ \ldots,\ k_{m-1})$, where $k_i \in \{0, 1\}$ for every $i = 0, \ldots, m-1$. The key k should be expanded to length t by repeatedly concatenating the bits $k_0, \ldots, k_{m-1}$ and taking the first t bits of such concatenation. The VEA's and MVEA's encryption function $E_k(P)$ for a single GOP is defined as

$$E_k(P) = (c_0,\ c_1,\ \ldots,\ c_{t-1}) \tag{9}$$

where $c_i = p_i \oplus k_{(i \bmod m)}$ for every $i = 0, \ldots, t-1$, or if the expanded key k is considered, each c_i is simply $c_i = p_i \oplus k_i$.

Consider the property in equation (8). It turns out that this property is quite useless in the multimedia steeting, since the new key can be easily obtained: Suppose that the original key k was compromised or needs to be changed for some reason. As was stated before, decryption is not necessary to obtain a new ciphertext. Hence one only needs to encrypt the original ciphertext $E_k(P)$ with a new key, say ℓ, to obtain a new ciphertext.

The new ciphertext can be then decrypted using the (expanded) key $k \oplus \ell$.

Now, take any plaintext bit p_i, for some $0 \leq i < t$. The old and new ciphertexts are

$$E_k(p_i) = p_i \oplus k_i, \tag{10}$$

$$E_\ell(E_k(p_i)) = (p_i \oplus k_i) \oplus \ell_i, \tag{11}$$

respectively, where k and ℓ are considered to be expanded keys. Values of both equations represent a ciphertext, and so they are known to the attacker. By subtracting equation (10) from equation (11), one can obtain ℓ_i. This can be done for any i, and thus one can easily obtain the whole key ℓ. Therefore, there is no security advantage in applying an additional encryption to an already encrypted content.

We conclude our analysis of VEA and MVEA algorithms by observing that the encryption function, as defined in equation (9), is exactly the definition of one of the classical ciphers - the Vigenère cipher. It immediately follows that a known-plaintext attack and a ciphertext-only attack are possible. For the latter attack, one can easily obtain the frequencies of selected bits that need to be encrypted by examining other MPEG-1 video streams. According to the analytical explanation as to why encrypting only these bits leads to sufficient protection[2], we believe that such frequency distribution cannot be uniform, therefore it is possible to launch ciphertext-only attack. For explanation of the attacks on the Vigenère cipher, see [44]. Also note, that because of the existence of resynchronization points at the beginning of each GOP that forces the key to start from its first bit, the length of the key k cannot be arbitrarily large.

4.2.5 An attack on the "Huffman" cipher

The "Huffman" cipher[1], is a light-weight MPEG-1 video encryption algorithm which incorporates encryption and decryption with MPEG-1 video encoding and decoding, respectively, in one step. First, the algorithm description from [2] and properties of this algorithm are stated. The cryptanalysis of the algorithm follows.

Let H be the Huffman codeword list provided by the JPEG-1 standard, and let π be a permutation of the list of codewords H which preserves the length of codewords, i.e., let π be a permutation over H such that $|w| = |\pi(w)|$ for $\forall w \in H$, where $| \cdot |$ denotes the length (number of bits) of a given codeword. Call the permuted list (the image of π) H'.

The MPEG-1 video encryption and decryption is then embedded in compression and decompression, respectively, as follows:

- Encryption: during the MPEG-1 video compression process, H is replaced with H'.

- Decryption: during the MPEG-1 video decompression, for every word $w \in H'$, $\pi^{-1}(w)$ is used as the real Huffman codeword value.

The characteristics of the algorithm are:

- No overhead is added – The encryption computation time is decreased by combining MPEG-1 compression and encryption (by replacing Huffman codeword list H with H'). The same applies to decompression and decryption. In other words, these processes do not result in extra computation time.
- The same compression ratio is achieved – Because of the property that for every $w \in H: \; |w| = |\pi(w)|$, the compressed and encrypted output from the MPEG-1 encoding process will have exactly the same size as if encryption was not included.

The first observation is that given encrypted video content and its corresponding original content, the known-plaintext attack can be applied. Thus, by examining I frames and their encrypted equivalents, it is not at all hard to determine the permutation π which serves as a key in this cipher.

A ciphertext-only attack involves exploiting the properties of the Huffman code. By the construction of the Huffman code, shorter codewords are assigned to symbols of more frequent input. Consequently, input symbols which rarely appear are assigned long codewords.

There are two fixed Huffman codeword lists used in standard JPEG encoding. The first codeword list, summarized in Table 6, is used to encode the DC coefficients, while the second list, summarized in Table 7, is used to encode the AC coefficients.

length of codewords	2	3	4	5	6	7	8	9
number of codewords	1	5	1	1	1	1	1	1

Table 6: Huffman codeword list for DC coefficients

There are only $5! = 120$ possible permutations for the first Huffman codeword list. However, there are

$$2! \cdot 3! \cdot 3! \cdot 2! \cdot 4! \cdot 3! \cdot 5! \cdot 5! \cdot 4! \cdot 4! \cdot 123! \approx 2^{718}$$

possible permutations of the second list, leading to an extremely large keyspace. By ignoring permutations involving long codewords (length 16), that is, by ignoring rare AC coefficients, the keyspace of the second list can be reduced to

$$2! \cdot 3! \cdot 3! \cdot 2! \cdot 4! \cdot 3! \cdot 5! \cdot 5! \cdot 4! \cdot 4! \approx 2^{38}$$

length of codewords	number of codewords
2	2
3	1
4	3
5	3
6	2
7	4
8	3
9	5
10	5
11	4
12	4
15	1
16	123

Table 7: Huffman codeword list for AC coefficients

allowing for a computationally feasible exhaustive search. In this case, we start by selecting the permutation π such that it fixes all the codewords of length 16. Then there are roughly 2^{38} possibilities for completing the permutation such that $|w| = |\pi(w)|$ for every $w \in H$.

Experiments showed that this attack on a 512×512 pixel image (4096 blocks) resulted in decryption failure of only 4 blocks, roughly 0.1% failure. All other blocks were decrypted successfully and allowed for almost no degradation of the original image. In addition, the incorrectly deciphered blocks can be approximated from their surrounding blocks. The reconstructed 512×512 pixel image, as produced by this attack, is in Figure 2.

4.2.6 Summary

Protecting multimedia content is a very important issue. On one hand, protection by a naive algorithm provides maximum security, but requires a special, costly hardware for real-time decryption. On the other hand, most of the multimedia applications need to balance between security and cost of video streaming. So there is an apparent tradeoff between security and the speed of streaming. The goal is to design a reasonably fast and secure encryption method such that breaking this method requires an investment several times higher than the value of the content.

Selective encryption is only one part of communication security. It is, however, a crucial

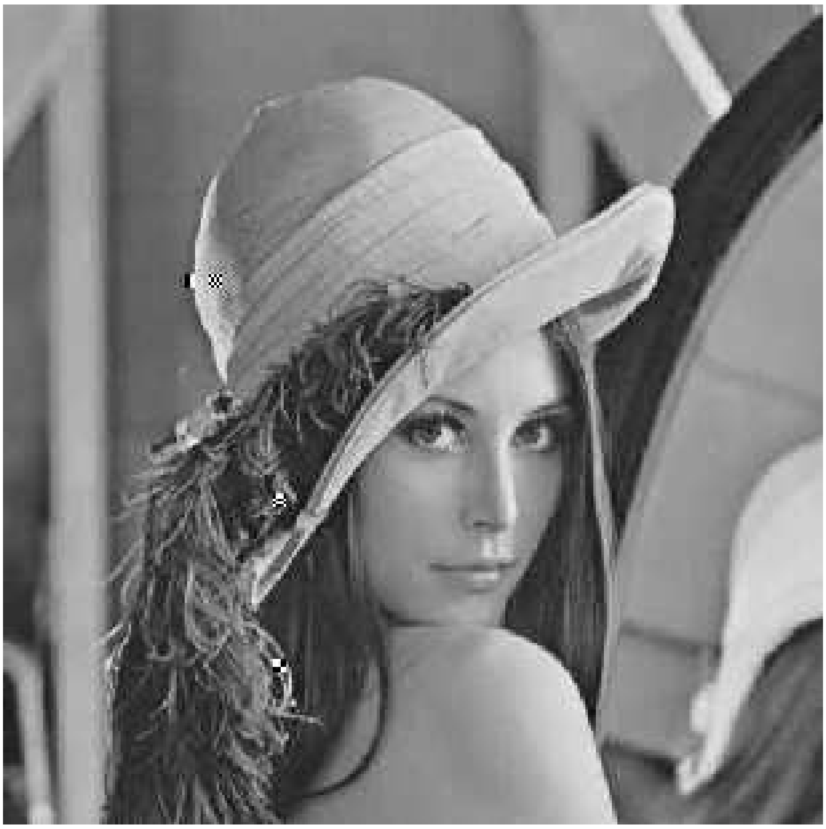


Figure 2: Reconstructed JPEG's Lena picture

part, and thus studying these kind of techniques is extremely important. Cryptanalysis of two selective MPEG-1 video encryption algorithms and one light-weight MPEG-1 video encryption algorithm based on permuting the Huffman codeword list was performed. All three of these methods were designed for efficiency, but unfortunately, as it was illustrated, they all lack security.

References

[1] Bharat Bhargava and Changgui Shi. Light-Weight MPEG Video Encryption Algorithm. In *Proc. of the Int'l Conf. on Multimedia - Multimedia 98, Shaping the Future*, pages 55–61, New Delhi, India, 1998.

[2] Bharat Bhargava, Changgui Shi, and Sheng-Yih Wang. MPEG Video Encryption Algorithms. *Multimedia Tools and Applications*, 24(1):57–79, 2004.

[3] Jean-Camille Birget. Time-complexity of the word problem for semigroups and the Higman Embedding Theorem. *International Journal of Algebra and Computation*, 8:235–294, 1998.

[4] Jean-Camille Birget. Circuits, coNP-completeness, and the groups of Richard Thompson. *International Journal of Algebra and Computation*, to appear. Preprint (2003), `http://arxiv.org/pdf/math/0310335`.

[5] Jean-Camille Birget. The groups of Richard Thompson and complexity. *International Journal of Algebra and Computation*, to appear. Preprint (2002), `http://arXiv.org/abs/math.GR/0204292`.

[6] Jean-Camille Birget, Spyros S. Magliveras, and Michal Sramka. On public-key cryptosystems based on combinatorial group theory. *Tatra Mt. Math. Publ.*, to appear. Cryptology ePrint Archive, Report 2005/070, `http://eprint.iacr.org/2005/070`.

[7] Jean-Camille Birget, Alexander Yu. Olshanskii, Eliyahu Rips, and Mark V. Sapir. Isoperimetric functions of groups and computational complexity of the word problem. *Annals of Mathematics*, 156(2):467–518, 2002.

[8] Garret Birkhoff. Three Observations on Linear Algebra. *Univ. Nac. Tucuman*, Rev. Ser. A(5):147–151, 1946.

[9] Greg Butler. An Inductive Schema for Computing Conjugacy Classes in Permutation Groups. *Math. Comp.*, 62(205):363–383, 1994.

[10] Don Coppersmith and Adi Shamir. Lattice attacks on NTRU. In *Advances in Cryptology - EUROCRYPT '97*, LNCS 1233, pages 52–61. Springer, 1997.

[11] Dubravko Culibrk, Daniel Socek, and Michal Sramka. Cryptanalysis of the Block Cipher based on the Hopfield Neural Network. *Tatra Mt. Math. Publ.*, submitted.

[12] Philip J. Davis. *Circulant Matrices.* John Wiley & Sons, New York, 2^{nd} edition, 1994.

[13] Ding-Zhu Du and Ker-I Ko. *Theory of Computational Complexity.* Wiley, 2000.

[14] L. Dulmage and I. Halperin. On a Theorem of Frobenius-Konig and J. von Neumann's Game of Hide and Seek. *Trans. Roy. Soc. Canada*, 3(49):23–29, 1955.

[15] Didier Le Gall. MPEG: A Video Compression Standard for Multimedia Applications. *Communications of the ACM*, 34(4):46–58, 1991.

[16] Maria Isabel Gonzalez-Vasco. *Criptosistemas Basados en Teoria de Grupos.* Tesis doctoral, Universidad de Oviedo, Spain, July 2003. `http://www.criptored.upm.es/paginas/investigacion.htm`.

[17] Maria Isabel Gonzalez-Vasco, Consuelo Martinez, and Rainer Steinwandt. *Toward a uniform description of several group based cryptographic primitives.* Cryptography ePrint Archive, Report 2002/048, `http://eprint.iacr.org/2002/048`, 2002.

[18] Maria Isabel Gonzalez-Vasco and Rainer Steinwandt. Reaction attacks on public key cryptosystems based on the word problem. Cryptography ePrint Archive, Report 2002/139, `http://eprint.iacr.org/2002/139`, 2002.

[19] Martin Grotschel, Laszlo Lovasz, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, Berlin, New York, 1991.

[20] Donghui Guo, L. M. Cheng, and L. L. Cheng. A New Symmetric Probabilistic Encryption Scheme Based on Chaotic Attractors of Neural Networks. *Applied Intelligence*, 10:71–84, 1999.

[21] Chris Hall, Ian Goldberg, and Bruce Schneier. Reaction attacks against several public-key cryptosystems. In *Information and Communication Security - ICICS'99*, LNCS 1726, pages 2–12. Springer, 1999.

[22] Graham Higman. Finitely presented infinite simple groups. *Notes on Pure Mathematics, The Australian National University, Canberra*, 8, 1974.

[23] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A Ring Based Public Key Cryptosystem. In *Algorithmic Number Theory - ANTS III*, LNCS 1423, pages 267–288. Springer, 1998.

[24] John J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. In *Proc. Natl. Acad. Sci. USA*, vol. 79, pages 2554–2558. Plenum Press, 1982.

[25] Nick Howgrave-Graham, Phong Q. Nguyen, David Pointcheval, John A. Proos, Joseph H. Silverman, Ari Singer, and William Whyte. The Impact of Decryption Failures on the Security of NTRU Encryption. In *Advances in Cryptology - CRYPTO '03*, LNCS 2729, pages 226–246. Springer, 2003.

[26] International Telecommunication Union (ITU-T). *Standard T.81 - Digital compression and coding of continuous-tone still images.* Switzerland, 1992.

[27] Birgit Jenner, Johannes Kobler, Pierre McKenzie, and Jacobo Toran. Completeness Results for Graph Isomorphism. *Journal of Computer and System Sciences*, 66:549–566, 2003.

[28] David Kahn. *Commemorating the 50th Anniversary of the National Security Agency.* A speech delivered at NSA, Fort Meade, MD, `http://www.fas.org/irp/eprint/kahn.html`, 2002.

[29] Auguste Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, IX:5–38 (Jan), 161–191 (Feb), 1883.

[30] Arjen K. Lenstra, Hendrik W. Lenstra, and Laszlo Lovasz. Factoring Polynomials with Rational Coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[31] Roger C. Lyndon and Paul E. Schupp. *Combinatorial group theory.* Springer-Verlag, New York, 1977.

[32] Klaus Madlener and Friedrich Otto. Pseudo-natural algorithms for the word problem for finitely presented monoids and groups. *Journal of Symbolic Computation*, 1:383–418, 1985.

[33] Wilhelm Magnus, Abraham Karrass, and Donald Solitar. *Combinatorial group theory; presentations of groups in terms of generators and relations.* Interscience Publishers, New York, 1966.

[34] Juergen Meyer and Frank Gadegast. *Security mechanisms for multimedia-data with the example MPEG-1-video.* Proj. description of SECMPEG, Technische Universitat Berlin, Germany, 1995.

[35] John A. Proos. *Imperfect Decryption and Partial Information Attacks in Cryptography.* Ph.d. thesis, University of Waterloo, Ontario, Canada, 2003.

[36] Mark V. Sapir, Jean-Camille Birget, and Eliyahu Rips. Isoperimetric and isodiametric functions of groups. *Annals of Mathematics*, 156(2):345–466, 2002.

[37] Bruce Schneier. A Self-Study Course in Block-Cipher Cryptanalysis. *Cryptologia*, 24(1):18–34, 2000.

[38] Claus P. Schnorr. Block Korkin-Zolotarev Bases and Successive Minima. Technical Report TR-92-063, Berkeley, CA, 1992.

[39] Elizabeth A. Scott. A construction which can be used to produce finitely presented infinite simple groups. *Journal of Algebra*, 90:294–322, 1984.

[40] Tanya E. Seidel, Daniel Socek, and Michal Sramka. Cryptanalysis of Video Encryption Algorithms. *Tatra Mt. Math. Publ.*, 29:1–9, 2004.

[41] Tanya E. Seidel, Daniel Socek, and Michal Sramka. Parallel Symmetric Attack on NTRU using Non-Deterministic Lattice Reduction. *Design, Codes and Cryptography*, 32(1-3):369–379, 2004.

[42] Adi Shamir. A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem. In *Advances in Cryptology - CRYPTO '82*, pages 279–288. Plenum Press, 1983.

[43] Charles C. Sims. Determining the Conjugacy Classes of a Permutation Group. In *Computers in Algebra and Number Theory, Proc. of Symposium in Applied Mathematics*, SIAM-AMS Proc. vol. 4, pages 191–195. AMS, 1971.

[44] Douglas R. Stinson. *Cryptography: Theory and Practice.* Chapman & Hall/CRC Press, Boca Raton, FL, 2^{nd} edition, 2002.

[45] Neal R. Wagner and Marianne R. Magyarik. A public-key cryptosystem based on the word problem. In *Advances in Cryptology - CRYPTO '84*, LNCS 196, pages 19–36. Springer-Verlag, 1985.

[46] Celia Wrathall. The word problem for free partially commutative groups. *Journal of Symbolic Computation*, 6:99–104, 1988.