

PUBLIC-KEY CRYPTOGRAPHY
BASED ON
SIMPLE SEMIRINGS

Dissertation
zur
Erlangung der naturwissenschaftlichen Doktorwürde
(Dr. sc. nat.)
vorgelegt der
Mathematisch-naturwissenschaftlichen Fakultät
der
Universität Zürich
von
Jens Zumbrägel
aus
Deutschland

Promotionskomitee

Prof. Dr. Joachim Rosenthal (Leitung der Dissertation)

Prof. Dr. Markus Brodmann

Prof. Dr. Michele Elia (Begutachter)

Dr. habil. Marcus Greferath (Begutachter)

Zürich, 2008

Contents

Abstract	v
Zusammenfassung	vii
Acknowledgements	ix
1 Cryptography	1
1.1 Cryptosystems	2
1.1.1 Encryption schemes	2
1.1.2 Digital signatures	3
1.2 Perfect security: Shannon's theory of secrecy	5
1.2.1 Perfect security	5
1.2.2 Indistinguishability	6
1.3 Computational security	6
1.3.1 Principles of complexity theory	7
1.3.2 Efficient algorithms and cryptosystems	8
1.3.3 Public-key cryptography	9
1.3.4 Notions of security	11
1.4 One-way functions and trapdoor functions	14
1.5 Discrete logarithm based cyptosystems	17
1.5.1 Function problems	18
1.5.2 The discrete logarithm problem	19
1.5.3 The Diffie-Hellman key agreement protocol	21
1.5.4 ElGamal encryption	23
1.5.5 Schnorr identification and signature	24
2 Cryptosystems based on semigroup actions	29
2.1 Semigroup actions	29
2.2 Semigroup action problems	32
2.2.1 Noncommutative semigroup actions	34
2.2.2 Problems in related semigroup actions	36
2.2.3 Two-sided group actions	38
2.3 Cryptosystems	40
2.3.1 Semigroup action Diffie-Hellman key agreement	42
2.3.2 Semigroup action ElGamal encryption	43
2.3.3 Identification protocols and digital signatures	45

2.4	Semigroup action based cryptosystems in the literature . . .	49
2.4.1	Cryptosystems using the modular group	49
2.4.2	Braid groups and cryptography	51
2.4.3	MOR cryptosystem	53
2.4.4	Further problems in other groups	54
3	Simple semirings	57
3.1	Introduction to semirings	57
3.1.1	Homomorphisms, congruences, ideals	59
3.1.2	Semimodules over semirings	61
3.1.3	Simple semirings	62
3.2	Classification of finite simple semirings with zero	63
3.2.1	Statement of the main theorem	64
3.2.2	Endomorphism semirings	65
3.2.3	Simple semirings and irreducible semimodules	67
3.3	The family of finite simple semirings	72
3.3.1	Isomorphism	72
3.3.2	The case $ \mathcal{SR}(M) = 1$	74
3.3.3	Congruence-simple semirings of small order	75
4	Semigroup actions based on simple semirings	77
4.1	Matrices over semirings	78
4.1.1	Matrices describing homomorphisms	79
4.1.2	Associativity of matrix multiplication	81
4.1.3	Semigroup actions based on matrices over semirings	83
4.2	Large endomorphism semirings	85
4.2.1	Cryptosystems using simple semirings	87
	Bibliography	91
	Index	97

Abstract

The discrete logarithm problem is the basic ingredient of many public-key cryptosystems. It can be stated as follows: Given a cyclic group (G, \cdot) of order n , a generator g of G , and another element $h \in G$, find the unique integer $a \in [0, n)$ such that $h = g^a$. The integer a is called the *discrete logarithm* of h to the base g .

There are key agreement protocols, public-key encryption schemes, and digital signatures employing the discrete logarithm problem. One example is the Diffie-Hellman key agreement protocol [DH76]. It allows two parties, A and B, to agree on a secret key over an insecure channel. In order to achieve this goal they fix a finite cyclic group G and a generator g of G . Then A and B pick random integers a, b respectively and exchange $h_A = g^a$ and $h_B = g^b$. Finally they compute $h_B^a = g^{ba}$ and $h_A^b = g^{ab}$, and since $g^{ab} = g^{ba}$ this element can be used as their secret key.

It is clear that solving the underlying discrete logarithm problem is sufficient for breaking the Diffie-Hellman protocol. For this reason one has been searching for groups in which the discrete logarithm problem is considered to be a computationally hard problem. Among the groups that have been proposed as candidates are the multiplicative group of a finite field and the group over an elliptic curve. It should however be pointed out that the infeasibility of the discrete logarithm problem has not been proved in any concrete group.

Discrete logarithm based cryptosystems can be generalized in the framework of *semigroup actions* (see e.g. [Mon02, Maz03, MMR07]). Here, an action

$$\rho : A \times X \rightarrow X, \quad (a, x) \mapsto \rho(a, x) = a \cdot x$$

of a semigroup A on a set X substitutes the role of the exponentiation $(\mathbb{Z}_n, \cdot) \times G \rightarrow G$ in a cyclic group G of order n . The semigroup action must satisfy (at least) the following two conditions.

- The *semigroup action discrete logarithm* (SDL) problem is hard: Given elements $g, h \in X$ such that $h \in A \cdot g$, the orbit generated by g , find an element $a \in A$ such that $h = a \cdot g$.
- There is a way to generate pairs of commuting elements of A .

It is an open problem at this point whether the SDL problem is harder to solve than the discrete logarithm problem. If this is true, the parameter sizes could be reduced in comparison to the discrete logarithm based protocols, leading to more efficient cryptosystems. To explore this issue it is clearly beneficial to create and study many examples.

A novel and promising approach to build interesting semigroup actions (proposed in [MMR07]) is based on finite simple semirings. A concrete example of such a construction is a two-sided action of matrices over a semiring. In order to avoid a Pohlig-Hellman-type reduction attack it is important that the semiring involved is simple.

The theoretical main result of this thesis is a full classification of finite simple semirings, analogous to the Wedderburn-Artin theorem. The result provides numerous examples which come from monoid endomorphism semirings of finite lattices. Due to this result it is possible to construct very large simple semirings using moderate computational resources, and this leads to new constructions of interesting semigroup actions for public-key cryptography. It will require further research to analyze these new systems.

The present thesis deals basically with three matters:

- We discuss semigroup actions and their use in cryptography, aiming to clarify the requirements needed to construct secure cryptosystems.
- We introduce semirings and give the classification of finite simple semirings up to isomorphism.
- We study the applications of simple semirings to the construction of semigroup actions for cryptography.

The first chapter introduces encryption schemes and digital signature schemes, including rigorous definitions of their security. Some discrete logarithm based cryptosystems and their underlying security assumptions will be discussed.

The second chapter is about cryptography based on semigroup actions. We present generalizations of the discrete logarithm based cryptosystems, and discuss the hardness of the underlying semigroup action problems. Moreover, we show that many proposals of cryptosystems in the literature of the last decade can be embedded into the setting of semigroup actions.

The third chapter deals with semirings and gives a full classification of finite simple semirings with zero. The result states that a finite semiring of order > 2 with zero which is not a ring is simple if and only if it is isomorphic to a “dense” subsemiring of the endomorphism semiring of a finite idempotent commutative monoid. We also investigate those subsemirings further, considering e.g. the question of isomorphism.

In the final chapter we discuss the applications of the classification for cryptography: We present different methods to construct semigroup actions based on simple semirings.

Zusammenfassung

Das Diskreter-Logarithmus-Problem (DL-Problem) ist die Grundlage für viele neuere Verfahren der Kryptographie. Es lautet: Gegeben seien eine zyklische Gruppe (G, \cdot) der Ordnung n , ein Erzeuger g von G und ein weiteres Element $h \in G$, gesucht ist die eindeutig bestimmte ganze Zahl $a \in [0, n)$ mit $h = g^a$. Diese Zahl a wird *diskreter Logarithmus* von h zur Basis g genannt.

Auf dem DL-Problem basieren Systeme für Schlüsselvereinbarungen, Public-Key Verschlüsselungen und digitale Signaturen. Ein Beispiel ist das Diffie-Hellman-Protokoll zur Schlüsselvereinbarung [DH76]. Es erlaubt zwei Kommunikationspartnern, A und B, die über ein nicht abhörsicheren Kanal kommunizieren, einen geheimen Schlüssel zu vereinbaren. Hierfür bestimmen sie öffentlich eine endliche zyklische Gruppe G und einen Erzeuger g von G . Dann wählen A und B zufällige Zahlen a bzw. b und tauschen die Nachrichten $h_A = g^a$ bzw. $h_B = g^b$ aus. Schließlich berechnen sie $h_B^a = g^{ba}$ bzw. $h_A^b = g^{ab}$, und wegen $g^{ab} = g^{ba}$ kann dieses Element als Schlüssel verwendet werden.

Offenbar kann ein Angreifer, der die versendeten Nachrichten des Protokolls abhört, in Besitz des Schlüssels gelangen, wenn er das DL-Problem lösen kann. Deswegen werden Gruppen verwendet in denen das DL-Problem als rechnerisch möglichst schwierig angesehen ist, d.h. es sind keine effizienten Algorithmen bekannt, die das DL-Problem lösen. Zum Beispiel wird die multiplikative Gruppe eines endlichen Körpers oder die Gruppe über einer elliptischen Kurve verwendet. Jedoch sollte betont werden, dass für keine konkrete Gruppe bewiesen wurde, dass kein effizienter Algorithmus für das DL-Problem existiert.

Das DL-Problem als Basis für kryptographische Verfahren kann mittels *Halbgruppen-Operationen* verallgemeinert werden (siehe [Mon02, Maz03, MMR07]). Dabei wird die Exponentiation $(\mathbb{Z}_n, \cdot) \times G \rightarrow G$ in einer Gruppe durch eine Operation

$$\rho : A \times X \rightarrow X, \quad (a, x) \mapsto \rho(a, x) = a \cdot x$$

einer Halbgruppe A auf eine Menge X ersetzt. Die Halbgruppen-Operation muss dabei notwendigerweise folgende Eigenschaften haben:

- Das Analogon zum Diskreter-Logarithmus-Problem (SDL-Problem) ist schwierig: Gegeben seien Elemente $g, h \in X$ mit $h \in A \cdot g$ (die von g erzeugte Bahn), gesucht ist ein Element $a \in A$ mit $h = a \cdot g$.
- Man kann Paare von kommutierenden Elementen von A erzeugen.

Es ist aktuell eine ungelöste Frage, ob das SDL-Problem schwieriger zu lösen ist als das DL-Problem. Wäre dies der Fall, so könnte man die Parametergrößen im Vergleich zu den DL-basierten Protokollen reduzieren, was zu effizienteren Verfahren führen würde. Um diese wichtige Fragestellung zu untersuchen ist es hilfreich, viele Beispiele zu erstellen und zu studieren.

Ein neuartiger und vielversprechender Konstruktionsansatz für interessante Halbgruppen-Operationen (vorgeschlagen in [MMR07]) basiert auf endlichen einfachen Halbringen. Ein konkretes Beispiel einer solchen Konstruktion ist eine zweiseitige Operationen von Matrizen über einem Halbring. Um einen Pohlig-Hellman-artigen Angriff via Reduktion zu vermeiden ist es wichtig, dass der zugrundeliegende Halbring einfach ist.

Das theoretische Hauptresultat der vorliegenden Dissertation ist eine vollständige Klassifikation von endlichen einfachen Halbringen, analog dem Satz von Wedderburn-Artin. Das Resultat liefert zahlreiche Beispiele für einfache Halbringe, nämlich Monoidendomorphismen-Halbringe von endlichen Verbänden und gewisse Unterhalbringe hiervon. Dadurch ist es mit wenig rechnerischem Aufwand möglich, sehr große einfache Halbringe zu erstellen, und dies führt zu neuen Konstruktionen von interessanten Halbgruppen-Operationen für Kryptographie. Für eine Sicherheitsanalyse der neuen Kryptosysteme sind weitere Untersuchungen notwendig.

Diese Dissertation umfasst drei thematische Gebiete:

- Wir studieren Halbgruppen-Operationen und ihre Anwendungen in der Kryptographie, mit dem Ziel, die Voraussetzungen zu erfassen, die für ein sicheres Kryptosystem nötig sind.
- Wir geben eine Einführung in die Halbring-Theorie und beweisen die Klassifikation von endlichen einfachen Halbringen.
- Wir untersuchen die Anwendbarkeit der einfachen Halbringe für die Konstruktion von kryptographischen Halbgruppen-Operationen.

Im ersten Kapitel werden Verschlüsselungs- und Signatur-Schemata eingeführt, ihre Sicherheit wird rigoros definiert. Es werden weiterhin einige DL-basierte Verfahren vorgestellt und ihre Sicherheitsannahmen spezifiziert.

Das zweite Kapitel behandelt Halbgruppen-Operationen im Hinblick auf Kryptographie. Es werden Verallgemeinerungen der DL-basierten Verfahren dargestellt und die Schwierigkeit der zugrundeliegenden SDL-Probleme diskutiert. Wir zeigen schließlich, dass viele neuere Vorschläge für Kryptosysteme im Kontext von Halbgruppen-Operationen eingebettet werden können.

Das dritte Kapitel enthält eine Einführung in Halbringe und es wird eine vollständige Klassifikation von endlichen einfachen Halbringen mit Null gegeben. Das Resultat besagt, dass ein endlicher Halbring mit Null der Ordnung > 2 , der kein Ring ist, genau dann einfach ist, wenn er isomorph zu einem "dichten" Unterhalbring eines Endomorphismen-Halbring eines endlichen idempotenten kommutativen Monoids ist. Wir untersuchen diese Halbringe anschließend bzgl. Isomorphie.

Im letzten Kapitel diskutieren wir die Anwendungen des Klassifikationsresultats für Kryptographie: Wir präsentieren verschiedene Ansätze, um Halbgruppen-Operationen zu konstruieren, die auf einfachen Halbringen basieren.

Acknowledgements

I owe my thanks to several people, without whom this work had not been possible. First and foremost, I am truly grateful to my advisor Joachim Rosenthal for his exemplary commitment and great vision. His friendship, advice and confidence in me have been an essential support for the whole time.

Special thanks go out to Gérard Maze for many helpful discussions, and also for the lattice figure in Chapter 4 which was generated with the help of his python program.

I am grateful to the Institute of Mathematics at the University of Zurich and the Swiss National Foundation, for providing me an excellent research environment and financial support. I would like to thank the members of my defense committee, Marcus Greferath, Michele Elia and Markus Brodmann, for their time, observations and suggestions.

Finally, many thanks go out to all my friends and colleagues at the Institute for their help and support, to the whole Applied Algebra workgroup, to Elisa and Davide for being fantastic office mates, and to Alina and Alberto who proofread my dissertation and gave very valuable feedback.

Chapter 1

Cryptography

Cryptography, literally the science of secret writing, is about one of the oldest desires of humankind: confidential communication. This discipline has a long history which can be traced back to the Ancient Egyptians, but during the last decades it has been transformed from an art to a science. Because of the proliferation of computers and communications systems cryptography is now used more than ever in everyday life.

Modern cryptography can be seen as the study of methods related to different aspects of information security, concentrating mainly on three important goals:

- *Secrecy*. The information should not leak to any unauthorized party.
- *Integrity*. The information must be protected against data manipulation.
- *Authentication*. The information should identify the author.

In special situations there may be further aspects of information security to consider, like nonrepudiation, electronic payment, anonymity, electronic votes, zero-knowledge proofs, etc.

This chapter deals with the basic notions of cryptography on which the applied part of this thesis is based. Details can be found in common cryptography textbooks, e.g. [Gol01, Gol04, KL08, MvOV97, Vau06].

In the first section of this chapter we define the syntax of general encryption and digital signature schemes. The following two sections discuss two approaches to define security: First we present the classical or information theoretic approach of Shannon, then we present the modern or complexity theoretic approach to security, and we include a part on public-key cryptosystems. In the fourth section two basic cryptographic primitives, namely one-way functions and one-way trapdoor functions, will be defined. The final section of this chapter deals with cryptosystems based on the discrete logarithm problem.

1.1 Cryptosystems

We define the components of encryption and signature schemes, focusing on a “syntactical” framework. Security considerations are not yet included.

1.1.1 Encryption schemes

We start with the issue of secrecy in communication, which is the classical goal of cryptography. Cryptographers widely appreciate and follow Kerckhoffs’ principle, which says that a cryptosystem should be secure even if its complete structure is known. In other words, the security of the cryptosystem must be based solely on the secrecy of keys.

We now give the definition of an encryption system. Let \mathcal{X} generally be a probability space. It will be used to model probabilistic encryption and signing functions as well as key distributions.

Definition 1.1.1. Let $\mathcal{M}, \mathcal{C}, \mathcal{K}$ be finite or countable sets denoting the *message space*, the *cipher space*, and the *key space*, respectively.

A **(probabilistic) encryption scheme** is specified as follows:

- For every $e \in \mathcal{K}$ there is a map $E_e : \mathcal{M} \times \mathcal{X} \rightarrow \mathcal{C}$, called *(probabilistic) encryption function*.
- For every $d \in \mathcal{K}$ there is a map $D_d : \mathcal{C} \rightarrow \mathcal{M}$, called *decryption function*.
- For every $e \in \mathcal{K}$ there is exactly one $d \in \mathcal{K}$ such that $D_d(E_e(m, x)) = m$ for all $m \in \mathcal{M}, x \in \mathcal{X}$. We refer to (e, d) as a *key pair*.

We write the encryption scheme as $(\{E_e\}_{e \in \mathcal{K}}, \{D_d\}_{d \in \mathcal{K}})$ or simply as $(\{E_e\}, \{D_d\})$.

We can view the encryption $E_e(m)$ of a message $m \in \mathcal{M}$ as a \mathcal{C} -valued random variable. If the encryption functions E_e are in fact deterministic, i.e. they do not depend on $x \in \mathcal{X}$, we speak of a **deterministic** encryption scheme. In this case, the encryption functions can be viewed as ordinary functions $E_e : \mathcal{M} \rightarrow \mathcal{C}$. These are injective, and for any key pair (e, d) we have $D_d \circ E_e = \text{id}_{\mathcal{M}}$. Probabilistic encryption schemes are used to achieve stronger levels of security, as we will see later.

Definition 1.1.2. The encryption scheme is called **symmetric** if $d = e$ for every key pair (e, d) .

The communication diagram of a symmetric encryption scheme is displayed in Figure 1.1.

Note that the sender and the receiver have to agree on a common secret key prior to their secret communication. They need a secure channel

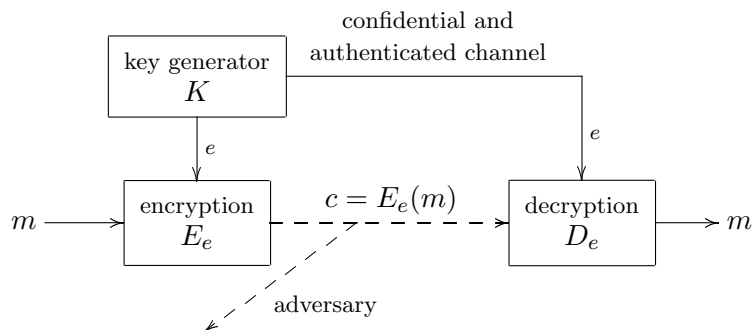


Figure 1.1: Symmetric encryption.

to exchange the key, i.e. a channel that provides both confidentiality and authentication. This is not necessary in certain nonsymmetric encryption schemes, namely public-key encryption schemes, see Section 1.3.3.

Cryptosystem 1.1.3. Let $(A, +)$ be a group and suppose that $\mathcal{M} = \mathcal{C} = \mathcal{K} = A$. Then $(\{E_e\}_{e \in A}, \{D_d\}_{d \in A})$, given by $E_e : A \rightarrow A, m \mapsto m + e$, and $D_d : A \rightarrow A, c \mapsto c - d$, is a symmetric deterministic encryption scheme.

If $(A, +)$ is the abelian group $((\mathbb{Z}_m)^n, +)$ then this example describes the so-called *one-time pad*. Note that the special case $(A, +) = (\mathbb{Z}_{26}, +)$ corresponds to Caesar's cipher (of a single letter).

There are technical extensions of Definition 1.1.1 concerning the message space, the cipher space, and the key space. They will be needed for some examples, see e.g. Cryptosystem 1.5.18 below.

- There is a distinction between the encryption key space and the decryption key space. These may be denoted \mathcal{K}_E and \mathcal{K}_D , respectively.
- The message space \mathcal{M}_e and the cipher space \mathcal{C}_e depend (partially) on the encryption key e . For example, the message length is restricted to be equal to the key length.

1.1.2 Digital signatures

Next we define general digital signature schemes. They are used to achieve the goal of authentication of information.

Definition 1.1.4. Let $\mathcal{M}, \mathcal{S}, \mathcal{K}$ be finite or countable sets denoting the *message space*, *signature space*, and *key space*, respectively.

A *(probabilistic) digital signature scheme* is specified as follows:

- For every $d \in \mathcal{K}$ there is a map $S_d : \mathcal{M} \times \mathcal{X} \rightarrow \mathcal{S}$, called *signing function*.

- For every $e \in \mathcal{K}$ there is a map $V_e : \mathcal{M} \times \mathcal{S} \rightarrow \{\text{yes, no}\}$, called *verification function*.
- For every $d \in \mathcal{K}$ there is exactly one $e \in \mathcal{K}$ such that $V_e(m, S_d(m, x)) = \text{yes}$ for all $m \in \mathcal{M}, x \in \mathcal{X}$. We again refer to (e, d) as a key pair.

We write the digital signature scheme as $(\{S_d\}_{d \in \mathcal{K}}, \{V_e\}_{e \in \mathcal{K}})$ or simply as $(\{S_d\}, \{V_e\})$.

In practice, $m \in \mathcal{M}$ is often a short extract of a longer message, say the value of a hash function (hash-and-sign method). As before, we can consider the signature $S_d(m)$ of a message m as an \mathcal{S} -valued random variable.

Definition 1.1.5. The digital signature scheme is *symmetric* if $d = e$ for every key pair (e, d) .

The communication diagram of a symmetric digital signature scheme is displayed in Figure 1.2.

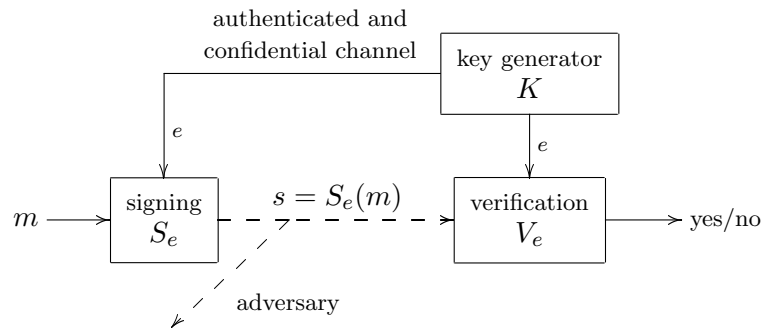


Figure 1.2: Message authentication.

As in the case of symmetric encryption schemes, the communicating parties have to agree on a common secret key in advance. Hence, only the legitimate receiver will be able to verify the signature. Symmetric digital signature schemes are also called *message authentication schemes*.

To complete the specification of encryption and digital signature schemes, it is necessary to say how the key pairs are generated. Recall that \mathcal{X} denotes a probability space.

Definition 1.1.6. A *key generator* for an encryption scheme or a digital signature scheme with key space \mathcal{K} is given by a map $K : \mathcal{X} \rightarrow \mathcal{K} \times \mathcal{K}$ such that any (e, d) in the image is a key pair.

From now on an encryption scheme or a digital signature scheme is called a *cryptosystem*.

1.2 Perfect security: Shannon's theory of secrecy

In this section we present the “classical” or information theoretic approach to the security of encryption schemes as developed by Shannon [Sha49]. Although the concept is of little practical relevance, it is conceptually easy and it leads to the notions of computational security developed in the next section.

Note that in Section 1.1 we gave only a syntactical definition for cryptosystems, so that also trivial¹ (i.e. insecure) cryptosystems are included. For the security evaluation of a cryptosystem we have to determine the security goal and the attack model. The *security goal* describes which type of breaks have to be prevented. The *attack model* defines the abilities of an adversary.

In this classical approach we examine only passive attacks, i.e. the adversary only intercepts ciphertexts, but has no access to messages together with their encryptions. Other types of attacks will be discussed in Section 1.3.4.

1.2.1 Perfect security

Let $(\{E_e\}_{e \in \mathcal{K}}, \{D_d\}_{d \in \mathcal{K}})$ be an encryption scheme that is symmetric², and let $K : \mathcal{X} \rightarrow \mathcal{K}$ be a key generator. We now state a very high security goal.

Definition 1.2.1. The encryption scheme is *perfectly secure* if every \mathcal{M} -valued random variable X (denoting a message) that is independent of the key K (and of all probabilistic encryptions $E_e(m)$) is also independent of the random variable $Y = E_K(X)$ (denoting its encryption).

The independence of X and Y is equivalent to the condition

$$P(X = x) = P(X = x | Y = y) \quad \text{for all } x \in \mathcal{M}, y \in \mathcal{C},$$

which says that the ciphertext Y reveals no further information about the distribution of the plaintext X . It can be shown that perfect security implies $|\mathcal{K}| \geq |\mathcal{M}|$, i.e. the key space has to be at least as large as the message space.

Example 1.2.2. The one-time pad, Cryptosystem 1.1.3, is a perfectly secure encryption scheme, provided the key is uniformly distributed over the key space. Of course, for each message to be encrypted one has to choose a new key to maintain the security.

¹For example, consider an encryption scheme $(\{E_e\}, \{D_d\})$ with $E_e = \text{id}_{\mathcal{M}}$ for all $e \in \mathcal{K}$.

²Since in a general encryption scheme the decryption key $d = d(e)$ is a function of e , we may assume in the information theoretic approach presented here that the encryption scheme is symmetric. This will be different in the complexity theoretic approach of Section 1.3, where nonsymmetric cryptosystems with not efficiently computable functions $d(e)$ are of importance.

1.2.2 Indistinguishability

We give another notion of security, which says that it is impossible to distinguish the encryptions of any two plaintexts.

Definition 1.2.3. The encryption scheme is *secure in terms of indistinguishability* if for every pair of messages $m_0, m_1 \in \mathcal{M}$ the random variables $E_K(m_0)$ and $E_K(m_1)$ denoting their encryptions are identically distributed.

It can be shown that security in terms of indistinguishability is equivalent to perfect secrecy. Furthermore, this definition is equivalent to the formulation given below, which will be modified in Section 1.3. Every adversary A has a chance of exactly $\frac{1}{2}$ to win the following game against a challenger C:

- (1) A chooses two messages $m_0, m_1 \in \mathcal{M}$ and sends them to C;
- (2) C chooses a bit $b \in \{0, 1\}$ uniformly at random, an encryption key e according to the distribution of K , and sends the encryption $c = E_e(m_b)$ to A;
- (3) A wins if it guesses correctly whether $b = 0$ or $b = 1$.

Since we assume that the adversary has unlimited computational resources one speaks of *unconditional security*. Later we will restrict the definition to adversarial algorithms that are efficient.

1.3 Computational security

Perfectly secure cryptosystems like the one-time pad are not very practical for two reasons. Firstly, a key as long as the longest possible message has to be generated and it has to be “truly random”. Secondly, the key has to be communicated between the parties in a secure way.

However, in practice perfect security is not needed, since actual adversaries do not have unlimited computational resources. This leads to the notion of *computational security*: The legitimate parties should be able to perform their tasks (e.g. encryption, decryption) efficiently, but the computational problem for malicious parties to abuse the system should be infeasible.

This concept was already mentioned by Shannon [Sha49, Part III] as “practical security”, but was fully established only decades later after the development of computational complexity theory and public-key cryptography in the 70s and probabilistic algorithms and cryptosystems in the 80s.

1.3.1 Principles of complexity theory

Complexity theory investigates the hardness of computational problems. We state briefly some important principles of complexity theory, since they are of significant influence on theoretical cryptography and thus also on this thesis. For more details we refer to textbooks on complexity theory, e.g. [AB09, Gol08, Pap94].

Problems and problem instances. The term *problem* refers to a general description of a computational task, and the term *instance* of a problem means a particular case of the task. The problem can be, for example, to factorize integers and an instance can be the problem to factorize the number 8051. Complexity theory is concerned with the difficulty of a problem rather than of a particular instance.

In cryptography one encounters two basic types of problems:

- Compute a function, e.g. for encryption and decryption, or produce a random element of a given distribution, e.g. for key generation.
- “Break” a cryptosystem; for encryption schemes this means to gain (partial) information about the message out of its encryption (see Section 1.3.4 for details).

Algorithms and computational models. Computational models, like Turing machines and Boolean circuits, make the notion of algorithm precise. There are different models, e.g. for deterministic, nondeterministic, probabilistic and quantum computing algorithms. Deterministic and probabilistic algorithms are usually seen as the practical realizable ones.

Our primary computational model for algorithms will be that of a probabilistic Turing machine.

Asymptotic approach. The amount of resources (like time and space) needed by an algorithm is given as a function $f(k)$ in the input length k . Mostly we will concentrate on *running time*, i.e. the number of steps performed during execution. Since it is possible to improve every algorithm by a constant speed-up factor, constants are neglected in the analysis. Hence, one is interested in the asymptotic behaviour of the function f .

An algorithm will be considered efficient if its running time is bounded by a polynomial. Defining the class of efficient algorithms this way has the primary advantage that this class is closed under composition: An efficient algorithm with oracle access to another efficient algorithm (which can be viewed as a subroutine) is equivalent to an efficient algorithm without oracle access.

Algorithms solving problems. For a full specification of a computational problem it is necessary to state which algorithms are considered as solving the problem. There are two different approaches:

- An algorithm “solves” the problem only if it computes the solution correctly for every instance. This is the classical approach in complexity theory and corresponds to a worst-case analysis. This approach applies to the legitimate parties.
- An algorithm is considered to “solve” the problem already if it computes the solution correctly with some nonnegligible probability for a random instance. This approach applies to adversaries.

Conditional results and reductions. As indicated by the fact that the famous $P \neq NP$ conjecture³ is still unproven, it is very hard to give lower bounds for the inherent complexity of a problem. Rather than making absolute statements, one compares instead the difficulty of different problems via the notion of reduction. In this sense it is possible to expose the most difficult problems in the class NP, namely the NP-complete problems.

We point out that the asymptotic approach of complexity theory limits its direct applicability to analyze the security of concrete cryptosystems with a specified security parameter: Fixing the security parameter means that only instances of a particular input length are considered. Concepts from complexity theory are nonetheless indispensable to establish the foundations of cryptography, and they lead to new protocols. Furthermore, it is usually relatively easy to translate a guarantee of asymptotic security into a concrete security guarantee.

1.3.2 Efficient algorithms and cryptosystems

As indicated, our notion of efficient algorithms (as those that can be practically performed by both the legitimate parties and the adversaries) will be that of probabilistic polynomial-time algorithms.

A *probabilistic algorithm* A can be modeled as a Turing machine that for every state has two subsequent states and at every step it tosses a fair coin to decide which successive state it should enter. The output $A(x)$ of the algorithm on input x can thus be seen as a random variable which is distributed according to the internal coin tosses of the algorithm.

The computational model of a probabilistic algorithm can be seen as the most powerful which is still practical. In fact, the *strong Church-Turing thesis* states that any “reasonable” model of computation can be efficiently simulated on a probabilistic Turing machine.

Definition 1.3.1. An algorithm is called *efficient* if it is probabilistic and runs in polynomial time in its input length.

³The class P denotes all decision problems that can be solved in deterministic polynomial time. It is contained in the presumably larger class NP, which can be defined as all decision problems solvable in nondeterministic polynomial time.

Remark 1.3.2. There are other notions for “efficient algorithms” in the literature. One of these notions is based on circuit complexity (see e.g. [Pap94, Section 11.4]): An algorithm is given by a family of Boolean circuits $C = (C_k)_{k \in \mathbb{N}}$, one circuit C_k for each input length k . The circuit family C is called *polynomially bounded* if the number of gates in C_k is bounded by a polynomial in k . It can be shown that every problem solvable by an efficient algorithm (as in the definition above) can also be solved by a polynomially bounded family of circuits.

Let $\{0, 1\}^*$ be the set of all bitstrings of finite length. We denote by $|x|$ the length of a bitstring $x \in \{0, 1\}^*$. A function $f : D \times \mathcal{X} \rightarrow \{0, 1\}^*$, where $D \subseteq \{0, 1\}^*$ and \mathcal{X} is a probability space, is *computed* by the algorithm A , if the random variables $f(x)$ and $A(x)$ are identically distributed, for every $x \in D$.

From now on we will tacitly assume that every cryptosystem is *efficient*. This means that for every encryption scheme $(\{E_e\}_{e \in \mathcal{K}}, \{D_d\}_{d \in \mathcal{K}})$ the message space \mathcal{M} , the cipher space \mathcal{C} , and the key space \mathcal{K} are subsets of $\{0, 1\}^*$, and all encryption functions $E_e : \mathcal{M} \times \mathcal{X} \rightarrow \mathcal{C}$ and all decryption functions $D_d : \mathcal{C} \rightarrow \mathcal{M}$ are efficiently computable⁴. Similar conventions apply to signature schemes.

Because of the asymptotic approach of complexity theory we introduce a *security parameter* k , which is involved into the key generator.

Definition 1.3.3. A (*scalable*) *key generator* for a cryptosystem with key space \mathcal{K} is given by an efficiently computable map

$$K : \{1^k \mid k \in \mathbb{N}\} \times \mathcal{X} \rightarrow \mathcal{K} \times \mathcal{K}$$

such that any (e, d) in the image is a key pair.

Here, 1^k denotes a string of 1s with length k . We note that $K(1^k)$ can be seen as a “key pair valued” random variable.

1.3.3 Public-key cryptography

The concept of computational security makes public-key cryptosystems possible, which are certain nonsymmetric cryptosystems. Here we give a slightly informal definition of a public-key encryption scheme. The precise definition is linked with the security definition given later.

Definition 1.3.4. An encryption scheme $(\{E_e\}, \{D_d\})$ is called a *public-key encryption scheme*, if for a given encryption key $e \in \mathcal{K}$ and a given ciphertext $c \in \mathcal{C}$ in the image of E_e it is “infeasible” to find the corresponding message $m \in \mathcal{M}$, i.e. m such that $E_e(m, x) = c$ holds for some $x \in \mathcal{X}$.

⁴Since the decryption functions are deterministic one may even assume that the decryption functions are computable by a deterministic polynomial-time algorithm.

Thus in a public-key encryption scheme the decryption of messages should be infeasible even if the encryption key is known (“public”). In particular, for a given e we cannot find a d such that (e, d) is a key pair, because otherwise we would find m as $D_d(c)$. The communication diagram of a public-key encryption scheme is depicted in Figure 1.3.

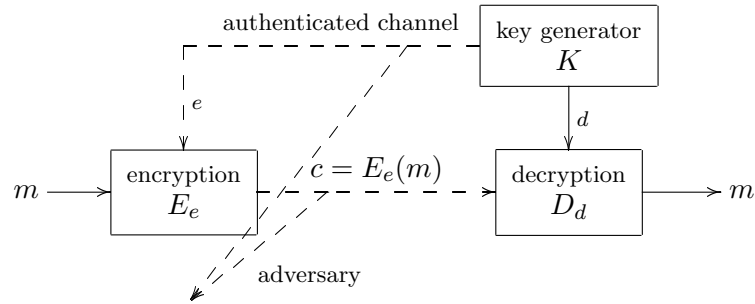


Figure 1.3: Public-key encryption.

Public-key encryption schemes are applied in the following way: Every party A has to generate and maintain only one key pair (e_A, d_A) for confidential communication with any of the other parties. A announces e_A and keeps d_A secret, so that e_A and d_A are referred to as A’s public and private key, respectively. Now everyone can encrypt a message m for A as $c = E_{e_A}(m)$. A uses its private key d_A to decrypt c as $D_{d_A}(c) = m$.

Hence, there is no need to exchange a key in a secure way prior to the communication. However, the public encryption keys must be authenticated, otherwise an impersonation attack is possible.

Definition 1.3.5. A digital signature scheme $(\{S_d\}, \{V_e\})$ is called a **public-key digital signature scheme** if for a given verification key $e \in \mathcal{K}$ and a given message $m \in \mathcal{M}$ it is “infeasible” to forge a valid signature, i.e. $s \in \mathcal{S}$ such that $V_e(m, s) = \text{yes}$.

For a given e in a public-key digital signature scheme we cannot find d such that (e, d) is a key pair, because otherwise $s = S_d(m)$ would be a valid signature for m . The communication diagram of a public-key digital signature scheme is depicted in Figure 1.4.

The use of public-key digital signature schemes in public-key cryptography is the following: Every party A generates a key pair (e_A, d_A) and announces e_A . Then, A’s signature $s = S_{d_A}(m)$ of a message m can be verified by everyone using $V_{e_A}(m, s)$.

Finally, there exist *key agreement protocols*, which use techniques from public-key cryptography, to establish a common key between two or more parties communicating over non-tap-proof channels. We do not give the formal definition here, since it is slightly involved, see e.g. [BWJM97]. An example of key agreement protocol will be given in Section 1.5.

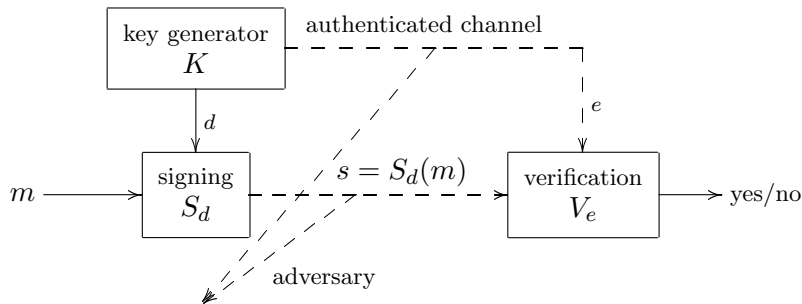


Figure 1.4: Digital signature.

1.3.4 Notions of security

Understanding the principles of security is extremely important for the design of new cryptosystems. We give definitions of the most important security notions for cryptosystems: what are the types of adversaries, what is considered a successful attack? We try to provide an overview of a huge area, still being a field of current research. For more detailed information the interested reader is referred to Goldreich's book [Gol04].

Security of encryption schemes

We recall that for a security definition of a cryptosystem one has to specify the abilities of malicious parties and what is considered a success of an attack. In other words one has to determine the attack model and the security goal.

There are different types of attack against encryption schemes. Weak adversaries can only perform a passive attack as in the classical approach of Section 1.2.

- *Ciphertext only attack*: Some ciphertexts are intercepted.

Adversaries with more capabilities will have access to messages together with their encryptions under the key being attacked. The following list states the most common types of attack, ordered by increasing strength.

- *Known plaintext attack*: Some messages with their encryptions can be received, without having control over the choice of messages.
- *Chosen plaintext attack*: The adversary may obtain encryptions of plaintexts of its choice, i.e. has access to an encryption oracle.
- *Chosen ciphertext attack*: The adversary can also obtain decryptions of ciphertexts of its choice, i.e. has access to both an encryption and a decryption oracle.

In the complexity theoretic approach presented here (and in contrast to Section 1.2) the adversaries are assumed to be efficient, i.e. they are probabilistic polynomial-time algorithms and may perform only polynomially many oracle-calls, so they can obtain only polynomially many plaintext/ciphertext pairs.

Now we define the security goal for encryption schemes. In computational security models the advantage of an attacker should often be “negligible” instead of 0, this is:

Definition 1.3.6. A function $\varepsilon : \mathbb{N} \rightarrow \mathbb{R}$ is called *negligible*, if for all $c \in \mathbb{N}$ there exists an $N \in \mathbb{N}$ such that for all $n \geq N$ we have

$$|\varepsilon(n)| \leq \frac{1}{n^c}.$$

More generally, a function $\nu : I \rightarrow \mathbb{R}$, where $I \subseteq \{0, 1\}^*$ is an infinite set, is called *negligible in $|i|$* , if there exists a negligible function $\varepsilon : \mathbb{N} \rightarrow \mathbb{R}$ such that $\nu(i) \leq \varepsilon(|i|)$ for all $i \in I$.

Thus a negligible function tends faster to 0 than any inverse of a polynomial function. We are now able to give the definition of polynomial indistinguishability.

Definition 1.3.7. Let $(\{E_e\}, \{D_d\})$ be a public-key encryption scheme with a scalable key generator K . The encryption scheme is called *secure (in terms of polynomial indistinguishability)* if an efficient adversary A has a chance of only $\frac{1}{2} + \varepsilon(k)$ to win the game below against a challenger C , where ε is a negligible function.

- (1) C chooses an encryption key e according to the distribution of $K(1^k)$ and sends it to A ;
- (2) A chooses two messages $m_0, m_1 \in \mathcal{M}$ and sends them to C ;
- (3) C chooses a bit $b \in \{0, 1\}$ uniformly at random, and sends the ciphertext $c = E_e(m_b)$ of the message m_b to A ;
- (4) A wins if it guesses correctly whether $b = 0$ or $b = 1$.

Note two differences between the game of the definition above and the game stated after Definition 1.2.3. Firstly, the adversary has to be efficient. In fact, no public-key encryption scheme would satisfy this security definition if computationally unbounded adversaries were allowed. Secondly, the adversary also knows the encryption key. In particular it can encrypt arbitrary messages by itself and hence can perform at least a chosen plaintext attack.

Secure public-key encryption schemes require probabilistic encryption functions. For, if the encryption functions are deterministic, the adversary

can use e to encrypt m_0 and m_1 and compare the result with the given cipher c .

If the adversary has also decryption abilities, one distinguishes two cases. Namely, if the oracle access to the decryption oracle is granted only before having received the challenge ciphertext, then one speaks of a *non-adaptive* chosen ciphertext attack. If the oracle access is granted even after having received the challenge ciphertext (in this case it is not allowed to use the decryption oracle for the target ciphertext), then the attack is called *adaptive*.

Security notions for symmetric encryption schemes are defined similarly, with the difference that the adversary is not given the encryption key e . In this case one has to distinguish between ciphertext only, known plaintext, and chosen plaintext attacks.

Remark 1.3.8. Another concept is that of semantic security, which extends Shannon's notion of perfect security for efficient adversaries. Speaking informally, an encryption scheme is *semantically secure* if whatever an adversary can compute efficiently about the plaintext given the ciphertext, the adversary can also compute efficiently without the ciphertext.

It can be shown that semantic security is equivalent to polynomial indistinguishability.

Remark 1.3.9. Another security notion, which is related to the issue of information integrity, is that of malleability. An encryption scheme is *malleable* if it is possible for an efficient adversary to transform a ciphertext into another ciphertext which decrypts to a related plaintext. That is, given an encryption of a plaintext m , it is possible to generate another ciphertext which decrypts to $f(m)$, for a known arbitrary function f , without necessarily knowing or learning m .

The main result here is that under adaptive chosen ciphertext attack, security in terms of polynomial indistinguishability is equivalent to non-malleability.

Security of digital signature schemes

For defining security of a digital signature scheme, one often assumes that the adversary can perform a *chosen message attack*, i.e. access is granted to a signing oracle. This attack can again be non-adaptive or adaptive.

We now define a strong notion of security for digital signature schemes.

Definition 1.3.10. Let $(\{S_d\}, \{V_e\})$ be a public-key digital signature scheme with a scalable key generator K . The signature scheme is *existentially unforgeable* if any efficient adversary A fails to create a valid signature for any message not signed before.

Precisely, if $(d, e) \sim K(1^k)$ is a key pair distributed according to the key generator, the probability $p_A(k)$ that A with input $(1^k, e)$ outputs a pair $(m, s) \in \mathcal{M} \times \mathcal{S}$ with the following properties is a negligible function:

- the signature s is valid for m , i.e. $V_e(m, s) = \text{yes}$,
- the message m is one for which A has not requested a signature during the attack.

Existentially unforgeable symmetric digital signature schemes (i.e. message authentication schemes) are defined similarly, with the difference that the adversary is not given the verification key e .

Finally, we mention two adversarial goals stronger than existential forgery (leading to weaker notions of security), which are also sometimes considered in the literature.

- *Selective forgery*: The adversary can choose some messages for which it can create a valid signature.
- *Universal forgery*: The adversary can create a valid signature for any message.

1.4 One-way functions and trapdoor functions

Now that we have defined different notions of practical security for encryption schemes and for digital signature schemes, the natural question is whether cryptosystems which satisfy certain security requirements exist and how to construct such schemes. A common approach is to study more elementary objects, whose existence can be proven, or for which at least some promising candidates are known, and then to construct cryptosystems out of them. These fundamental tools which form the basic ingredients of a cryptosystem are often called *primitives*.

One-way functions

One-way functions belong to the most important cryptographic primitives. Informally speaking, these functions are efficiently computable but computationally hard to invert. Their importance for cryptography was stressed by Diffie and Hellman in their seminal paper [DH76], although a rigorous treatment was not given until later.

Recall that an algorithm is called *efficient* if it is probabilistic and runs in polynomial time.

Definition 1.4.1. A map $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ is called *one-way function* if

- there is an efficient algorithm which computes $f(x)$ for a given x ,

- any efficient algorithm A fails to invert the function f .

Precisely, if x is a uniformly distributed random variable over $\{0, 1\}^k$, the probability $p_A(k)$ that A with input $(1^k, f(x))$ outputs z with $f(z) = f(x)$ is a negligible function in k .

It is not known whether one-way functions exist, but their existence is conjectured by many authors, and some candidates are given in the examples below. In fact, the existence of one-way functions would imply $P \neq NP$, which is perhaps the most well-known open conjecture in theoretical computer science. Conversely, it is not proven that $P \neq NP$ implies the existence of one-way functions, mainly because of the distinction between worst-case hardness and average-case hardness.⁵

One-way functions have multiple cryptographic applications. We remark that the existence would imply (and is in fact equivalent to each of) the existence of secure pseudo-random number generators, secure public-key digital signature schemes, and weakly collision-resistant families of hash functions. Details can be found in [Gol01].

Example 1.4.2. The multiplication map $\mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, $(m, n) \mapsto m \cdot n$, restricted to m, n of same binary length, is widely believed to be a one-way function (we use a standard encoding of natural numbers as bitstrings to obtain a map $\{0, 1\}^* \rightarrow \{0, 1\}^*$). The inversion of this function depends on the integer factorization problem, which is a computationally hard problem from experience.

Collections of functions

For convenience reasons we generalize the definition of one-way function to a collection of functions with finite domains. This is useful for the definition of one-way trapdoor functions and simplifies giving examples.

Recall that $|i|$ denotes the length of a bitstring $i \in \{0, 1\}^*$.

⁵In fact, $P \neq NP$ is equivalent to $FP \neq FNP$, the complexity theory statement in terms of relations (see e.g. [Pap94, Section 10.3]). The latter statement means exactly that there exists a relation $R \subseteq \{0, 1\}^* \times \{0, 1\}^*$ with the following properties:

- R is balanced, i.e. for any $(u, v) \in R$ the length of v is polynomially bounded in the length of u ,
- R is polynomial-time checkable ($R \in FNP$), i.e. for given (u, v) it can be decided in polynomial time whether $(u, v) \in R$,
- R is not polynomial-time computable ($R \notin FP$), i.e. no polynomial-time algorithm can compute for all $u \in \{0, 1\}^*$ an element $v \in \{0, 1\}^*$ with $(u, v) \in R$.

It is easy to see that for a one-way function f the inverse relation f^{-1} satisfies these properties. The third condition means that every (deterministic) polynomial-time algorithm fails to compute the relation for *at least one* argument u , but for a one-way function we need that even every probabilistic polynomial-time algorithm fails to compute the relation for *almost every* argument u . The existence of one-way functions is thus a stronger hardness assumption than $P \neq NP$.

Definition 1.4.3. Let $\{f_i\}$ be a collection of maps $f_i : \{0, 1\}^{|i|} \rightarrow \{0, 1\}^*$, indexed by $i \in \{0, 1\}^*$. Then, $\{f_i\}$ is called **one-way collection of functions** if

- there is an efficient algorithm which computes $f_i(x)$ on input (i, x) ,
- any efficient algorithm A' fails to invert the functions f_i .

Precisely, if i and x are uniformly distributed random variables over $\{0, 1\}^k$, the probability $p_{A'}(k)$ that A' with input $(i, f_i(x))$ outputs z with $f_i(x) = f_i(z)$ is a negligible function in k .

It is easy to see that if f is a one-way function, then $\{f_i\}$ defined by $f_i(x) = (i, f(x))$ is a one-way collection. Conversely, if $\{f_i\}$ is a one-way collection then a function f with $f(i, x) = (i, f_i(x))$ is one-way.

Example 1.4.4 (Exponentiation modulo p). Let I be the set of all pairs (p, g) , where p is a prime and g is a generator of the cyclic group \mathbb{Z}_p^* , and consider the collection of bijective functions

$$f_{(p,g)} : \mathbb{Z}_{p-1} \rightarrow \mathbb{Z}_p^*, \quad a \mapsto g^a$$

with $(p, g) \in I$. The problem of inverting these functions is called the *discrete logarithm* (DL) problem, and it is widely believed to be hard. Thus the collection $\{f_{(p,g)}\}$ is a candidate for a one-way collection.⁶

This example can be generalized to other groups in which the discrete logarithm problem is believed to be hard. Section 1.5 will deal with DL-based cryptosystems.

One-way trapdoor functions

For some cryptographic purposes special one-way functions are important, namely those with a supplementary information which enables efficient inversion.

Definition 1.4.5. A collection of one-way functions $\{f_i\}$ is called a **collection of one-way trapdoor functions** if there exists a binary relation $T \subseteq \{0, 1\}^* \times \{0, 1\}^*$ with the following properties:

- There is an efficient algorithm which for input 1^k outputs a pair $(i, t) \in T$ where i is uniformly distributed over $\{0, 1\}^k$.
- There is an efficient algorithm which for every input $((i, t), f_i(x))$ with $(i, t) \in T$ and $x \in \{0, 1\}^{|i|}$ outputs z with $f_i(x) = f_i(z)$.

⁶The given description is not complete, since we do not state the encoding maps. For the construction of a map from $\{0, 1\}^*$ to I we remark that there exists an efficient algorithm which generates primes p , together with the prime factorization of $p - 1$, which can be used to generate g . The encoding maps from \mathbb{Z}_{p-1} or \mathbb{Z}_p^* to $\{0, 1\}^k$ are straightforward.

We view the element t as a *trapdoor*, which allows for efficient computation of preimages of f_i . Note that there is no efficient algorithm which finds a trapdoor t on input the index i : Otherwise, one could construct an efficient algorithm which on input $(i, f_i(x))$ computes a preimage of $f_i(x)$, contradicting the assumption that $\{f_i\}$ is a collection of one-way functions. Despite this fact, the first property states that elements in T can be easily generated.

One-way trapdoor functions are closely related to public-key encryption schemes, see Definition 1.3.4. In fact, the encryption maps E_e of a deterministic public-key encryption scheme may serve as one-way trapdoor functions with the trapdoor information being the decryption key d .

We close with two examples of candidates for one-way trapdoor functions, without giving any encoding details.

Example 1.4.6 (RSA function). Let I consist of all pairs (n, e) , where $n = pq$ is a number composed of different primes p, q in the order of \sqrt{n} , i.e. having $\frac{1}{2} \log_2 n$ bits, and $e < n$ is a coprime number to $\varphi(n) = (p-1)(q-1)$. The bijective functions

$$f_{(n,e)} : \mathbb{Z}_n \rightarrow \mathbb{Z}_n, \quad m \mapsto m^e$$

form a candidate for a collection of one-way trapdoor functions. For each $(n, e) \in I$ the trapdoor t is the inverse d of e modulo $\varphi(n)$, which can be computed if the factorization of n is known. Once d is known, the inverse map of $f_{(n,e)}$ can be efficiently computed, since it is given by

$$f_{(n,e)}^{-1} : \mathbb{Z}_n \rightarrow \mathbb{Z}_n, \quad c \mapsto c^d.$$

However, it is not known whether the inversion problem of the function $f_{(n,e)}$ is equivalent to the factoring problem of n .

Example 1.4.7 (Rabin function). For n being a composite number as in the example before consider the squaring maps

$$f_n : \mathbb{Z}_n \rightarrow \mathbb{Z}_n, \quad m \mapsto m^2.$$

These form also a candidate for a collection of one-way trapdoor functions, in fact, it can be shown that the problem of inverting these maps is equivalent to the problem of factoring n . The trapdoor information is thus the factorization of n . Unfortunately, this map is not injective, because it is $4 : 1$ on \mathbb{Z}_n^* .

1.5 Discrete logarithm based cyptosystems

Many cryptosystems, including the Diffie-Hellman key agreement protocol and the ElGamal encryption and signature schemes, employ the intractability of the discrete logarithm problem. In this section we first introduce the

discrete logarithm problem and the related Diffie-Hellman problem, then we present some discrete logarithm based cryptosystems.

1.5.1 Function problems

Since the discrete logarithm problem is a computational function (or search) problem, we start with a short formal introduction of this concept, following [Gol08].

Definition 1.5.1. Let $R \subseteq \{0,1\}^* \times \{0,1\}^*$ be a binary relation. For $x \in \{0,1\}^*$ let $R(x) := \{y \in \{0,1\}^* \mid (x,y) \in R\}$.

The **function problem** or **search problem** for R is the algorithmic problem to compute on input $x \in \{0,1\}^*$ an element $y \in R(x)$. We call y a **solution** to the problem **instance** x . We speak simply of the “function problem R ” instead of the “function problem for the relation R ”.

The **domain** $D(R)$ of a function problem R is the set of all problem instances with a solution, $D(R) := \{x \in \{0,1\}^* \mid R(x) \neq \emptyset\}$. The function problem is *polynomially bounded* if there exists a polynomial p such that for every $y \in R(x)$ we have $|y| \leq p(|x|)$.

Polynomially bounded function problems guarantee that the length of a solution is not too long compared with the length of the problem instance.

Definition 1.5.2. Let A be a deterministic algorithm which halts on every input. Then A is said to **solve** the function problem R , if for all inputs $x \in D(R)$ A outputs a solution, i.e. $A(x) \in R(x)$, and for all inputs $x \notin D(R)$ A outputs a special symbol \perp , indicating that x has no solution.

Now we state a class of function problems which are easy to solve.

Definition 1.5.3. A polynomially bounded function problem is said to be in the class FP, if there exists a polynomial time algorithm A solving the problem.

To employ the hardness of a function problem for cryptography it is not enough that the problem is outside of FP. Rather we need the intractability of the problem in the average case, as defined below.

Definition 1.5.4. Let R be a polynomially bounded function problem and let $(\mu_k)_k$ be a sequence of probability distributions on the set $\{0,1\}^*$ of problem instances. The problem R is called **intractable** if for every efficient algorithm A the success probability averaged over the instance distribution μ_k ,

$$p_A(k) := \sum_{x \in \{0,1\}^*} \mathbf{P}(A(x) \in R(x)) \mu_k(x),$$

is a negligible function in k .

Example 1.5.5. Let $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ be a one-way function, let R be the inverse relation f^{-1} , and for every k let μ_k be the distribution of $f(U_k)$, where U_k is a uniformly distributed $\{0, 1\}^k$ -valued random variable. Then the inversion problem R is intractable.

1.5.2 The discrete logarithm problem

Now we introduce the discrete logarithm problem in general cyclic groups, following the presentations in [Bon98] and [CS03]. The following definition specifies the computational requirements for such groups. Due to the asymptotic approach of complexity theory we have to consider group families.

Definition 1.5.6. A *group family* G is a set of finite cyclic groups $G = \{G_i\}$, where i ranges over an infinite index set $I \subseteq \{0, 1\}^*$. We assume that:

- $G_i \subseteq \{0, 1\}^*$ and there are polynomial time (in $|i|$) algorithms computing the multiplication maps $G_i \times G_i \rightarrow G_i$ and the inversion maps $G_i \rightarrow G_i$;
- the group sizes $n_i = |G_i|$ can be efficiently computed.

An *instance generator* IG for G is an efficient algorithm that for given 1^k outputs some random index i and a generator g of G_i . The pair (i, g) is called a *group instance*.

Remark 1.5.7. If the group has efficiently computable operations, then also the powers g^α of a group element α can be computed efficiently using a square-and-multiply method (see e.g. [MvOV97, Algorithm 2.143]).

Example 1.5.8. The family $\{\mathbb{Z}_p^*\}_p$ of multiplicative groups of prime fields \mathbb{Z}_p can be seen as a group family. For this we have to specify an encoding of the elements of \mathbb{Z}_p^* as bitstrings in $\{0, 1\}^*$. The natural way to do this is to identify the quotient ring $\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}$ with the set $\{0, 1, \dots, p-1\}$ of representatives, and to encode these integers into their binary representations.⁷

The instance generator is used to select a member of G of the appropriate size. For example, on input 1^k the instance generator may generate a random k -bit prime p such that $\frac{p-1}{2}$ is also prime.

Other groups than \mathbb{Z}_p^* that are of interest in cryptography are e.g. the multiplicative group \mathbb{F}_q^* of a general finite field \mathbb{F}_q and the group of points on an elliptic curve defined over a finite field. In these examples there exists always “natural” encoding of group elements as bitstrings, which we do not state explicitly.

From now on we will identify $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$ as a set with $\{0, 1, \dots, n-1\}$.

⁷The actual encoding of the group elements as bitstrings is not important as long as the transition maps between different encodings are efficiently computable in both directions.

Definition 1.5.9. Let $G = \{G_i\}$ be a group family. The *discrete logarithm* (DL) problem in G is the following problem:

Given a triple (i, g, h) , where $i \in I$, g is a generator of the group G_i of order n_i , and $h \in G_i$, find an element $a \in \mathbb{Z}_{n_i}$ such that $h = g^a$.

The DL problem appears to be a computationally hard problem in general. The (bijective) exponentiation maps

$$f_i : \mathbb{Z}_{n_i} \rightarrow G_i, \quad a \mapsto g^a$$

form thus a candidate for a collection of one-way functions, see Example 1.4.4. We pin down the hardness assumption of the DL problem in the next definition.

Definition 1.5.10. Let $G = \{G_i\}$ be a group family with instance generator IG . Furthermore, let μ_k be the probability distribution on DL instances (i, g, g^a) such that (i, g) is distributed as $IG(1^k)$ and a is uniformly distributed on \mathbb{Z}_{n_i} , where $n_i = |G_i|$.

Then the group family satisfies the *DL assumption* if the DL problem is intractable with respect to $(\mu_k)_k$.

Recall from Definition 1.5.4 that intractability means that every efficient algorithm attempting to solve the DL problem has only negligible success probability on average.

The known algorithms to solve the DL problem fall into two classes (see e.g. [MvOV97, Section 3.6] for more details):

- Generic or black-box algorithms work in arbitrary groups, i.e. they only perform group operations and computations that do not involve the encoding of group elements. However, some of these algorithms perform well only in groups of certain orders.

It has been shown that any generic algorithm to solve the DL problem in a group of prime order p needs at least $\Omega(p^{1/2})$ group operations [Sho97].

This bound is achieved by both the baby-step-giant-step algorithm and Pollard's rho algorithm. If the group order is however smooth, so that it has only small prime factors, then the Pohlig-Hellman algorithm works faster (see the following remark).

- Other algorithms are efficient only in certain groups, i.e. they use properties of the encoding map of group elements as bitstrings. Examples are index-calculus algorithms.

The fastest index-calculus methods for the DL problem in prime fields or in fields of characteristic 2 run in subexponential time, namely $O(\exp(c(\log n)^{1/3}(\log \log n)^{2/3}))$, where n is the group size and $c > 0$ is some constant.

Remark 1.5.11. The main idea of the Pohlig-Hellman algorithm can be described algebraically as follows. Let G be a cyclic group of order n , so that G is naturally a \mathbb{Z}_n -module. Suppose n is composite, say $n = km$. Then the module G is not simple, i.e. there exists a nontrivial epimorphism onto some smaller \mathbb{Z}_n -module. Indeed, $\theta : G \rightarrow G, h \mapsto h^k$ is an endomorphism of \mathbb{Z}_n -modules whose image is a subgroup $H \leq G$ of order m .

Now let g be a generator of G and let (g, h) be an instance of a DL problem, where $h = g^a$. Then $h^k = (g^a)^k = (g^k)^a$ and hence solving the DL problem $(\theta(g), \theta(h)) = (g^k, h^k)$ in the subgroup H determines the discrete logarithm a modulo m .

If the subgroup H is much smaller than the original group G the DL problem is easier to solve in H . Now if n is a smooth number, the solution a to the original DL problem can be constructed in this way from several discrete logarithm computations in small subgroups of G .

This explains why simple modules G (which are groups of prime order) have the hardest DL problem and are thus desirable for cryptographic purposes. Furthermore, this *Pohlig-Hellman type reduction argument* gives motivation for the study of simple structures like (congruence-)simple semi-rings in Chapter 3 and 4 of this thesis.

1.5.3 The Diffie-Hellman key agreement protocol

The first discrete logarithm based cryptosystem we present establishes a common key between two parties, A and B, communicating over an insecure but authenticated channel, see [DH76].

Cryptosystem 1.5.12. The *Diffie-Hellman key agreement* protocol is the following: Let G be group family⁸ with instance generator IG .

- During setup phase a group instance (i, g) is selected and published by applying $IG(1^k)$, where k is the security parameter. Let n be the order of the cyclic group G_i with generator g .
- A chooses a random element $a \in \mathbb{Z}_n$ and sends $h_A = g^a \in G_i$ to B, retaining a secretly.
- B chooses a random element $b \in \mathbb{Z}_n$ and sends $h_B = g^b \in G_i$ to A, retaining b secretly.
- A computes $h_B^a = g^{ba}$ and B computes $h_A^b = g^{ab}$. Their common key is $e = g^{ab} = g^{ba} \in G_i$.

The security analysis for this protocol suggests to consider the following problem, which is related to the DL problem.

⁸Originally, Diffie and Hellman used the multiplicative groups \mathbb{Z}_p^* of prime fields \mathbb{Z}_p as the group family.

Definition 1.5.13. Let $G = \{G_i\}$ be a group family. The *computational Diffie-Hellman* (CDH) problem in G is the following problem:

Given (i, g, h_A, h_B) , where $i \in I$, g is a generator of the group G_i of order n_i , and $h_A = g^a$ and $h_B = g^b$ for some $a, b \in \mathbb{Z}_{n_i}$, find the element g^{ab} .

Let us define group families with difficult CDH problem.

Definition 1.5.14. Let $G = \{G_i\}$ be a group family with instance generator IG . Furthermore, let μ_k be the probability distribution on CDH instances (i, g, g^a, g^b) such that (i, g) is distributed as $IG(1^k)$ and a, b are uniformly distributed on \mathbb{Z}_{n_i} , where $n_i = |G_i|$.

Then the group family satisfies the *CDH assumption* if the CDH problem is intractable with respect to $(\mu_k)_k$.

There is another, stronger assumption which is very useful for proving security properties of cryptographic protocols. This assumption is the intractability of the decision version of the Diffie-Hellman problem.

Definition 1.5.15. Let $G = \{G_i\}$ be a group family. The *decision Diffie-Hellman* (DDH) problem in G is the following problem:

Given (i, g, h_A, h_B, h_C) , where $i \in I$, g is a generator of the group G_i of order n_i , and $h_A = g^a$, $h_B = g^b$ and $h_C = g^c$ for some $a, b, c \in \mathbb{Z}_{n_i}$, decide whether $c = ab$ holds.

Loosely speaking, the DDH assumption states that no efficient algorithm can distinguish between the two distributions (g^a, g^b, g^{ab}) and (g^a, g^b, g^c) , where a, b, c are chosen at random.

Definition 1.5.16. Let $G = \{G_i\}$ be a group family with instance generator IG . Let μ_k be the probability distribution on instances (i, g, g^a, g^b, g^c) , where (i, g) is distributed as $IG(1^k)$ and a, b, c are uniformly distributed on \mathbb{Z}_{n_i} , where $n_i = |G_i|$.

The group family satisfies the *DDH assumption* if every efficient algorithm D has negligible advantage

$$\Delta_D(k) := \mathbf{P}_{\mu_k}(D(i, g, g^a, g^b, g^{ab}) = 1) - \mathbf{P}_{\mu_k}(D(i, g, g^a, g^b, g^c) = 1).$$

It is easy to see that a DL solver with noticeable success can be efficiently transformed into a CDH solver with noticeable success. Indeed, given a CDH instance (i, g, g^a, g^b) we apply our DL algorithm to (i, g, g^a) which computes a with nonnegligible probability. In this case, g^{ab} is found as $(g^b)^a$.

Likewise, a CDH solver with noticeable success can be efficiently transformed into a DDH distinguisher with noticeable advantage. Indeed, given a DDH instance (i, g, g^a, g^b, g^c) the output h of the CDH solver equals g^{ab} with nonnegligible probability. Now the DDH algorithm decides $c = ab$ according to whether $g^c = h$.

In summary, we thus have

$$\text{DL assumption} \Rightarrow \text{CDH assumption} \Rightarrow \text{DDH assumption}.$$

Remark 1.5.17. Regarding the DDH assumption we make the following remarks. Details can be found in [Bon98].

- (1) If the groups G_i of the group family have orders with small prime factors the DDH assumption is not satisfied. Indeed, suppose $n = |G_i|$ has a small prime factor p . Then $ab \in \mathbb{Z}_n$ is more likely to be divisible by p than c , if $a, b, c \in \mathbb{Z}_n$ are uniformly distributed. This leads to an effective distinguisher between triples (g^a, g^b, g^{ab}) and (g^a, g^b, g^c) .
- (2) On the other hand, if the groups G_i are of prime order (or, more generally, the order has only large prime factors), then the DDH assumption follows from the weaker *perfect-DDH assumption*. The latter assumption states that no efficient algorithm decides (with overwhelming probability) for *any* given triple (h_A, h_B, h_C) whether it is of the form (g^a, g^b, g^{ab}) or not.

The equivalence of these assumptions follows from a randomized self-reduction argument, see [Bon98, Theorem 3.1].

- (3) There are several group families in which the best known algorithm for the DDH problem is a full discrete logarithm algorithm. One simple example is the group family $\{Q_p\}$, where p is a prime of the form $2q + 1$ with a prime q , and Q_p is the subgroup of \mathbb{Z}_p^* of order q .

1.5.4 ElGamal encryption

ElGamal [ElG85] constructed an encryption scheme and a digital signature scheme based on the discrete logarithm problem. We present first the encryption scheme and then a variant of the signature scheme proposed by Schnorr.

Cryptosystem 1.5.18. Let $G = \{G_i\}$ be a group family with instance generator IG . The *ElGamal encryption scheme* for G is the following probabilistic encryption scheme.

- The key generator $K(1^k)$ applies the instance generator IG as a subroutine and outputs the key pair $((i, g, h), a)$. Here, (i, g) is a group instance distributed as $IG(1^k)$, a is uniformly distributed on \mathbb{Z}_{n_i} , where $n_i = |G_i|$, and $h = g^a \in G_i$.

The public key is $e = (i, g, h)$, the private key is $d = a$.

- The encryption of a message $m \in G_i$ is $E_{(i,g,h)}(m) = (g^b, m h^b)$, where b is a random element distributed uniformly on \mathbb{Z}_{n_i} .
- The decryption of a ciphertext (c_1, c_2) is $D_a(c_1, c_2) = c_2 c_1^{-a}$.

Note that the decryption works, since for every a, b, m we have

$$D_a(E_e(m)) = D_a(g^b, m h^b) = m (g^a)^b (g^b)^{-a} = m.$$

We make the following remarks.

- ElGamal encryption is closely related to the Diffie-Hellman key agreement protocol: Suppose (g^a, a) is the key pair generated by party A. Then if a party B sends a secret message to A, it sends g^b , retaining b secretly. Both parties can compute the Diffie-Hellman key g^{ab} which is used to disguise the message m .

The difference is that A's key a is here a long term secret key in contrast to the short term secret keys in the Diffie-Hellman protocol.

- The message space is the group G_i . To encrypt arbitrary messages in $\{0, 1\}^k$ we assume that there exist efficiently computable and reversible injective maps from $\{0, 1\}^k$ into G_i , provided that $2^k \leq |G_i|$.
- In the encryption $E_e(m) = (g^b, m h^b)$ the operation $m \cdot h^b$ can be replaced by any unrelated group operation, say XOR.
- It can be shown that the security of the ElGamal encryption scheme (in terms of polynomial indistinguishability) is equivalent to the intractability of the DDH problem.

1.5.5 Schnorr identification and signature

The Schnorr identification and signature schemes [Sch90] are related cryptosystems whose security is based directly on the hardness of the DL problem. The Schnorr digital signature scheme is derived from the Schnorr identification protocol, which in turn is based on a zero-knowledge proof of the knowledge of a discrete logarithm.

Zero-knowledge proofs of knowledge

A zero-knowledge (ZK) proof of knowledge is a protocol between two parties, called the prover and the verifier, that allows the prover to demonstrate knowledge of a secret while revealing no information about the secret. They can be used in *identification protocols*, i.e. protocols that prove that a party is the one it claims to be.

Below we give a more detailed definition for ZK proofs of knowledge. Firstly, by a *protocol* between two communicating parties we mean a pair of algorithms interacting with each other, i.e. each algorithm receives the output of the other algorithm. Each interactive algorithm is described by its *message-specification function*, which determines the next message the algorithm sends, depending on the received messages. By a *secret* we mean

a solution $y \in R(x)$ to the problem instance x of a function problem R , i.e. an element y such that $(x, y) \in R$. Recall that $D(R) = \{x \mid R(x) \neq \emptyset\}$.

Definition 1.5.19. A *proof of knowledge* for the function problem R is a protocol between two parties P and V being efficient algorithms receiving a common input x , such that the following properties hold for any $x \in D(R)$.

- *Completeness.* If the prover P knows a secret $y \in R(x)$ (as private input) then the verifier V accepts the prover's claim, i.e. it outputs 1 after the interaction.
- *Soundness.* Let B be an algorithm impersonating the prover interacting with V, and let p be the probability that V accepts. Then B can be used in the following sense to reveal a secret:

There is an efficient algorithm K with oracle access to the message-specification function of B, which outputs $y \in R(x)$ with probability

$$s \geq f_{|x|}(p);$$

here $(f_k)_k$ is a family of convex functions $f_k : [0, 1] \rightarrow \mathbb{R}$ with the property that $r(k)$ is negligible whenever $f_k(r(k))$ is negligible. The algorithm K must not depend on x and B and is called *universal knowledge extractor*.

The soundness property formalizes the idea that only algorithms knowing the secret are able to convince the verifier. Examples for function families $(f_k)_k$ with the required properties are:

- (1) $f_k(r) = r$, corresponding to the simple inequality $s \geq p$,
- (2) $f_k(r) = r - \varepsilon(k)$, where $\varepsilon(k)$ is a negligible function,
- (3) $f_k(r) = r^2$.

Definition 1.5.20. A *ZK proof of knowledge* is a proof of knowledge for R between two parties P and V with the following additional property.

- *Zero-knowledge* property. A single algorithm can efficiently produce for all $x \in D(R)$, without interaction, an output which is indistinguishable from a protocol *transcript*, i.e. the collection of messages from P and V resulting from an execution on common input x .

This property formalizes the following idea: The verifier gains no knowledge from the interaction with the prover, since it could generate equivalent transcripts by itself, so that in this way the verifier is able to simulate the prover. In our definition of the zero-knowledge property we consider only transcripts of the interaction of P with the “honest” verifier V, and a more precise term for this property is *honest-verifier zero-knowledge*⁹.

⁹A more strict zero-knowledge property postulates that for every efficient algorithm W impersonating the verifier there exists an efficient noninteractive algorithm that simulates transcripts of the interaction of P with W, see [Gol01, Chapter 4].

The identification scheme

Many zero-knowledge proofs consist of three passes, namely a commitment of the prover, a challenge of the verifier, and a response of the prover to the challenge. A protocol of this type forms a crucial part of the following Schnorr identification protocol.

Cryptosystem 1.5.21. Let G be family of prime order groups with instance generator IG . The *Schnorr identification* protocol is this:

- During setup phase a group instance (i, g) is selected and published by applying $IG(1^k)$, where k is the security parameter. Let n be the order of the group G_i . Each claimant P chooses a private key $a \in \mathbb{Z}_n$ and publishes $h = g^a$ as its public key.¹⁰
- P identifies itself to a verifier V by proving knowledge of its private key a as follows.

(Commitment) P chooses randomly $b \in \mathbb{Z}_n$ and sends $r = g^b \in G_i$,

(Challenge) V sends a random $c \in \mathbb{Z}_n$,

(Response) P sends $s = b + ac \in \mathbb{Z}_n$.

The verifier V accepts if and only if $g^s = r h^c$.

The choice of b and c is according to the uniform distribution on \mathbb{Z}_n .

The identification procedure is a ZK proof of knowledge of a discrete logarithm. The corresponding function problem R consists of pairs $((i, g, g^a), a)$, where (i, g) is a group instance, $a \in \mathbb{Z}_{n_i}$, and $n_i = |G_i|$.

We justify briefly why the protocol has the required properties. Firstly, if P and V act as prescribed then $r h^c = g^b g^{ac} = g^s$ as required, i.e. the protocol is complete.

For the soundness property, let B be an algorithm impersonating the prover interacting with V , and let p be the probability that V accepts. The knowledge extractor K having oracle access to B 's message-specification function obtains B 's commitment r , where $r = g^b$, chooses two different random challenges $c_0, c_1 \in \mathbb{Z}_n$ and obtains B 's answers s_0, s_1 . If V would accept both answers we have $g^{s_i} = r h^{c_i}$ and thus $s_i = b + ac_i$ for $i = 0, 1$. Since $c_0 \neq c_1$ and n is prime K can compute the secret $a = (s_1 - s_0)/(c_1 - c_0)$ in this case. For K 's success probability we have $s \geq p^2 - \frac{1}{n}$, and $\frac{1}{n}$ is negligible in k .

Finally, the protocol has the zero-knowledge property, since an output which has the same distribution as the protocol transcript $(g^b, c, b + ac)$ can be generated without interaction as $(g^s/h^c, c, s)$, where s is distributed uniformly on \mathbb{Z}_n .

¹⁰Practically, the claimant P has to obtain a certificate from a trusted party binding P 's identity with its public key.

Furthermore, the security of the identification protocol depends on the assertion that the secret value a is knowledge that cannot be efficiently deduced from the public parameters. Hence it depends on the hardness assumption of the discrete logarithm problem.

The signature scheme

The Schnorr signature scheme is deduced from the Schnorr identification protocol. In order to make the protocol noninteractive, the challenge is replaced by the value of a hash function.

Definition 1.5.22. A *hash function family* $H = \{H_i\}_{i \in I}$ is a set of functions $H_i : \{0, 1\}^* \rightarrow A_i$ from the binary strings into a finite set A_i such that:

- $H_i(m)$ is efficiently computable;
- every efficient algorithm A fails to find a collision for H_i :

Precisely, the probability $p_A(i)$ that A on input i outputs a pair (m_0, m_1) such that $H_i(m_0) = H_i(m_1)$ is negligible in $|i|$.

We recall that the groups G_i of a group family are represented by subsets of $\{0, 1\}^*$, in particular hash functions can be applied to group elements.

Cryptosystem 1.5.23. Let $G = \{G_i\}$ be a family of prime order groups with instance generator IG , and let $H = \{H_i\}$ be a family of hash functions $H_i : \{0, 1\}^* \rightarrow \mathbb{Z}_{n_i}$, where $n_i = |G_i|$. The *Schnorr signature scheme* for G is the following probabilistic digital signature scheme.

- The key generator $K(1^k)$ applies the instance generator IG as a subroutine and outputs the key pair $((i, g, h), a)$. Here, (i, g) is a group instance distributed as $IG(1^k)$, a is uniformly distributed on \mathbb{Z}_n , where $n = |G_i|$, and $h = g^a \in G_i$.

The public key is $e = (i, g, h)$, the private key is $d = a$.

- The signature of a message m is

$$S_a(m) = (c, s) = (H_i(m, r), b + ac) \in \mathbb{Z}_n^2,$$

where b is a random element distributed uniformly on \mathbb{Z}_n , and $r = g^b$.

- The verification of a signature (c, s) for a message m is

$$V_{(i, g, h)}(m, (c, s)) = \text{yes} \text{ if and only if } H_i(m, g^s h^{-c}) = c.$$

Remark 1.5.24. Provided the group family satisfies the DL assumption, the Schnorr signature scheme is existentially unforgeable under an adaptive chosen-message attack, see [PS00, Theorem 14].

Chapter 2

Cryptosystems based on semigroup actions

The exponentiation map $\mathbb{Z}_n \times G \rightarrow G$, $(a, x) \mapsto x^a$ in a finite cyclic group (G, \cdot) of order n is crucial for the discrete logarithm based cryptosystems. This is an example for a semigroup action, namely the commutative semigroup (\mathbb{Z}_n, \cdot) acts on the set G . As observed by Maze, Monico, and Rosenthal ([MMR07], see also [Mon02] and [Maz03]) the framework of commutative semigroup actions leads to generalized Diffie-Hellman and ElGamal cryptosystems.

This chapter deals with semigroup actions and their use to create cryptosystems. We extend the framework of [MMR07] to include also noncommutative semigroups. This enables us to discuss a larger variety of examples and leads to more aspects and tools for studying the difficulty of the semigroup action problems. We also present new semigroup action based frameworks for two identification protocols and one digital signature scheme. In the last section we show that many proposals of cryptosystems in the literature of the last decade can be embedded into the setting of semigroup actions.

2.1 Semigroup actions

Definition 2.1.1. Let (A, \cdot) be a semigroup and X be a set. A *(left) semigroup action* of A on X is a map

$$\rho : A \times X \rightarrow X, \quad (a, x) \mapsto \rho(a, x) = a \cdot x,$$

such that $(a \cdot b) \cdot x = a \cdot (b \cdot x)$ for all $a, b \in A$ and $x \in X$. If such an action exists, the set X is called an *A-set*.

We often abbreviate $(a \cdot b) \cdot x$ as $ab \cdot x$. Note that $b \cdot x = b' \cdot x$ implies $ab \cdot x = ab' \cdot x$ for all $a, b, b' \in A$ and $x \in X$.

Remark 2.1.2. For any semigroup action of A on X , we have by definition a semigroup homomorphism

$$\begin{aligned} A &\longrightarrow T(X), \\ a &\longmapsto [x \mapsto a \cdot x] \end{aligned}$$

from A into the monoid $T(X)$ of all maps $X \rightarrow X$. Conversely, any semigroup homomorphism $\psi : A \rightarrow T(X)$ defines a semigroup action of A on X by $a \cdot x := [\psi(a)](x)$ for $a \in A$ and $x \in X$.

If (A, \cdot) is a group with neutral element 1 and the semigroup action satisfies $1 \cdot x = x$ for any $x \in X$, then we speak of a *group action*. In analogy to the above remark, group actions correspond to group homomorphisms $A \rightarrow S(X)$ from A into the group $S(X)$ of invertible maps $X \rightarrow X$.

We introduce some nonstandard, but convenient notation.

Definition 2.1.3. An action of a semigroup A on a set X is called *semitransitive* if there exists an element $g \in X$ such that $X = A \cdot g$, where $A \cdot g$ denotes the orbit $\{a \cdot g \mid a \in A\}$ of g . In this case, the set X is called a *monogenic* A -set, and g is called a *generator* for X .

For any A -set X and any $g \in X$, the orbit $A \cdot g \subseteq X$ will be a monogenic A -set in a natural way. For the subsequent applications we thus often assume the semigroup action to be semitransitive.

Remark 2.1.4. With respect to group actions semitransitivity, transitivity, and the existence of only one orbit are equivalent. In this case every $x \in X$ is a generator, and the surjective orbit map

$$\varphi_x : A \rightarrow X, \quad a \mapsto a \cdot x$$

induces a bijection between X and the set $A/\ell \text{Stab}(x)$ denoting the left cosets of A with respect to the stabilizer subgroup $\text{Stab}(x)$ of X . The group action is called *simply transitive* if the map φ_x is bijective. Simply transitive group actions in cryptography were studied by Couveignes [Cou06].

Example 2.1.5. The exponentiation map $\mathbb{Z}_n \times G \rightarrow G$, $(a, x) \mapsto a \cdot x := x^a$ in a cyclic group (G, \cdot) of order n with generator g makes G a monogenic \mathbb{Z}_n -set.

As mentioned in the beginning of this chapter, this is the motivating example for studying semigroup actions in cryptography. It may be helpful to think in general of A as a generalized “space of exponents” acting on a set X . As we will see, algebraic properties of A have implications on the security of the corresponding cryptosystems presented below.

Some examples of semigroup actions

Example 2.1.6. These are examples of semigroup actions $\rho : A \times X \rightarrow X$, $\rho(a, x) = a \cdot x$.

- (1) Let (A, \cdot) be a semigroup, $X = A$, and ρ the semigroup operation:

$$a \cdot x := a \cdot x .$$

- (2) Let X be a set, let A be a subsemigroup of the monoid $T(X)$ of all maps $a : X \rightarrow X$, and let ρ be the evaluation:

$$a \cdot x := a(x) .$$

- (3) Let (A, \cdot) be a group, $X = A$, and ρ be the group conjugation:

$$a \cdot x := a \cdot x \cdot a^{-1} .$$

This is a group action and obeys the following two special rules

$$a \cdot (x \cdot y) = (a \cdot x) \cdot (a \cdot y) , \quad a \cdot (x \cdot y) = (a \cdot x) \cdot (a \cdot y) .$$

- (4) Let $(R, +, \cdot)$ be a semiring (with zero), X a semimodule over R , $A = (R, \cdot)$, and ρ the R -multiplication:

$$a \cdot x := a x .$$

This action obeys the special rules

$$a \cdot (x + y) = a \cdot x + a \cdot y , \quad (a + b) \cdot x = a \cdot x + b \cdot x .$$

In particular, this example applies to the \mathbb{Z} -module structure of abelian groups, see Example 2.1.5.

Remark 2.1.7. Right and two-sided actions can be recognized also as left actions:

- (a) Let (R, \cdot) be a semigroup and X be a set. A *right semigroup action* of R on X is a map

$$X \times R \rightarrow X , \quad (x, r) \mapsto x \cdot r ,$$

such that $x \cdot (a \cdot b) = (x \cdot a) \cdot b$ for all $a, b \in A$ and $x \in X$.

Consider the *dual semigroup* R^{op} of R , i.e. the same set with reversed operation. Then

$$a \cdot x := x \cdot a$$

defines a left action of $A = R^{op}$ on X .

- (b) Let $L \times X \rightarrow X$ be a left and $X \times R \rightarrow X$ be a right semigroup action, and suppose that

$$(\ell . x) . r = \ell . (x . r) =: \ell . x . r$$

for $\ell \in L$ and $r \in R$. In this case we speak of a *two-sided semigroup action* of L and R on X .

This defines a left action of $A = L \times R^{op}$ on X by

$$(\ell, r) . x = \ell . x . r .$$

- (c) We restate (b) of this remark in different notation: Let $G \times X \rightarrow X$ and $H \times X \rightarrow X$ be actions of semigroups G and H on a set X , and suppose that

$$g . (h . x) = h . (g . x)$$

for $g \in G$ and $h \in H$. Then for $A = G \times H$ there is a composite action

$$(g, h) . x := g . (h . x) = h . (g . x) .$$

Example 2.1.8. These are further examples of semigroup actions.

- (1) Let (H, \cdot) be a semigroup. The semigroup operation defines a two-sided action of H on itself. By Remark 2.1.7 (b), there is an action of $A = H \times H^{op}$ on $X = H$, given by

$$(a_1, a_2) . x := a_1 \cdot x \cdot a_2 .$$

- (2) Let (X, \cdot) be a group, $G = X$, $H = \mathbb{Z}$, and consider the group conjugation $G \times X \rightarrow X$ and the exponentiation $H \times X \rightarrow X$. These actions commute as in Remark 2.1.7 (c), hence there is an action of $A = X \times \mathbb{Z}$ on X , given by

$$(a, n) . x := a \cdot x^n \cdot a^{-1} .$$

2.2 Semigroup action problems

In this section we state the analogs of the discrete logarithm problem and the Diffie-Hellman problems in the context of a semigroup action (cf. Example 2.1.5), and discuss the hardness of these problems.

From now on we assume that both the semigroup operation $A \times A \rightarrow A$ and the action map $A \times X \rightarrow X$ are efficiently computable. A formal definition of efficiency depends on an asymptotic setting and will be given later (Definition 2.3.1). In this section the intuitive meaning of efficiency will be sufficient.

The semigroup action discrete logarithm problem

Definition 2.2.1. Let X be a monogenic A -set and g be a generator. The *semigroup action discrete logarithm* (SDL) problem is this:

Given $h \in X$, find $a \in A$ such that $h = a \cdot g$.

We note that the SDL problem need not to have a unique solution, in fact every $a' \in A$ with $a \cdot g = a' \cdot g$ will also be one. We introduce a notation. For $x \in X$ define an relation \sim_x on A by

$$a \sim_x a' \quad :\Leftrightarrow \quad a \cdot x = a' \cdot x .$$

One readily verifies that \sim_x is an equivalence relation and a *left congruence*, i.e. $a \sim_x a'$ implies $ba \sim_x ba'$ for all $b \in A$. With this notation, the SDL problem is unique up to \sim_g .

The SDL problem has been called the semigroup action problem (SAP) in [Mon02, Maz03, MMR07]¹. We note that for solving the SDL problem there is an analog of Pollard's rho algorithm which works well if the semigroup A has a large fraction of invertible elements, see [Mon02, Algorithm 4.4] or [MMR07, Section 2.1].

Commutative semigroup action Diffie-Hellman problems

For the Diffie-Hellman problems we state the problems for simplicity first in the commutative case.

Definition 2.2.2. Let A be a commutative semigroup, let X be a monogenic A -set and let g be a generator. The *semigroup action computational Diffie-Hellman* (SCDH) problem is this:

Given $h_A, h_B \in X$, find the element $ab \cdot g = ba \cdot g$ such that $h_A = a \cdot g$ and $h_B = b \cdot g$ for some $a, b \in A$.

It is easy to show that in the commutative case the SCDH problem has a unique solution, i.e. the demanded element $ab \cdot g$ depends only on h_A and h_B , see Lemma 2.2.4 below. Clearly, if one has a solution of the SDL problem for either h_A or h_B , then also the SCDH problem is solved, since $ab \cdot g = a \cdot h_B = b \cdot h_A$.

Definition 2.2.3. Let A, X, g as before. The *semigroup action decision Diffie-Hellman* (SDDH) problem is this:

Given $h_A, h_B, h_C \in X$, decide whether the triple (h_A, h_B, h_C) is of the form $(a \cdot g, b \cdot g, ab \cdot g)$ for some $a, b \in A$ or not.

¹In order to differentiate between the analogs of the discrete logarithm problem and the various Diffie-Hellman problems in the semigroup action setting we apply the convention to use the common abbreviation of the problem (e.g. DL or CDH) preceded by s.

We note that the first two problems are search or function problems whereas the last is a “distinguish” problem. The difficulty of these semigroup action problems will be discussed in many examples appearing later in this section (Sections 2.2.2 and 2.2.3).

2.2.1 Noncommutative semigroup actions

Many applications of semigroup actions to public-key cryptography (see Section 2.4) use noncommutative semigroups. Before stating the semigroup action Diffie-Hellman problems in the noncommutative case we prove a simple but useful lemma.

Lemma 2.2.4. *Let X be an A -set, and let $a, a', b \in A$ and $x \in X$.*

- (1) *If $a \cdot x = a' \cdot x$ and both a and a' commute with b , then $ab \cdot x = a'b \cdot x$.*
- (2) *Suppose $a \cdot x = a' \cdot x$, $b \cdot x = b' \cdot x$, and both a and a' commute with either b or b' . Then $ab \cdot x = a'b' \cdot x$.*

Proof. (1) follows simply from

$$ab \cdot x = ba \cdot x = ba' \cdot x = a'b \cdot x.$$

For (2), if a and a' commute with b , then $ab \cdot x = a'b \cdot x$ by (1), hence $ab \cdot x = a'b' \cdot x$. Similarly, if a and a' commute with b' , then $a'b' \cdot x = ab' \cdot x$ by (1), hence $a'b' \cdot x = ab \cdot x$. \square

In general $a \cdot x = a' \cdot x$ does not imply $ab \cdot x = a'b \cdot x$, as the following example shows. Thus Lemma 2.2.4 (1) does not hold without any commutativity assumption (see also Remark 2.2.6).

Example 2.2.5. Suppose $A = S_3$ acts naturally on $X = \{1, 2, 3\}$. Let $x = 3$, and let $a = ()$, $a' = (12)$ and $b = (23)$. Then we have $a \cdot x = a' \cdot x = 3$, but $ab \cdot x = 2$ and $a'b \cdot x = 1$, so that $ab \cdot x \neq a'b \cdot x$.

Recall that $a \sim_x a'$ means $a \cdot x = a' \cdot x$, and that \sim_x is a left congruence. Let $\sim := \sim_x$. With this notation Lemma 2.2.4, (1) reads: Let $a, a', b \in A$ and suppose $a \sim a'$. If both a and a' commute with b , then $ab \sim a'b$. Indeed, $ab \sim ba \sim ba' \sim a'b$.

Remark 2.2.6. We discuss some possibly weaker conditions under which $a \sim a'$ implies $ab \sim a'b$.

- (1) The commutativity assumption $ab = ba$ and $a'b = ba'$ can obviously be weakened to $ab \sim ba$ and $a'b \sim ba'$ (what might be called *local commutativity*). It even suffices that b' exists with $b \sim b'$ and $a'b' \sim b'a$ and $a'b' \sim b'a'$.

- (2) Suppose $a \sim a'$ and a is invertible. Then $ab \sim a'b$ is *equivalent* to the commutativity condition $a^{-1}a'b \sim ba^{-1}a'$.

Proof. If a is invertible, then $ab \sim a'b$ is equivalent to $b = a^{-1}ab \sim a^{-1}a'b$. Now, $a \sim a'$ implies $b = ba^{-1}a \sim ba^{-1}a'$, thus the equivalence is proved. \square

Similarly, if a' is invertible, then $ab \sim a'b$ is equivalent to $(a')^{-1}ab \sim b(a')^{-1}a$.

Noncommutative semigroup action Diffie-Hellman problems

When the semigroup is noncommutative some care has to be taken in the definition of the Diffie-Hellman problem analogs. We consider two different semigroup action computational Diffie-Hellman problems.

Definition 2.2.7. Let X be a monogenic A -set and g be a generator.

- The *general semigroup action computational Diffie-Hellman* (GSCDH) problem is this:

Given $h_A, h_B \in X$, find an element $ab.g$ such that $h_A = a.g$ and $h_B = b.g$ for some $a, b \in A$.

- Let C_A, C_B be commuting subsets of A , i.e. $ab = ba$ for all $a \in C_A$ and $b \in C_B$. The *special semigroup action computational Diffie-Hellman* (SSCDH) problem with respect to C_A and C_B is this:

Given $h_A \in C_A.g$ and $h_B \in C_B.g$, find the element $ab.g = ba.g$ such that $h_A = a.g$ and $h_B = b.g$ for some $a \in C_A$ and $b \in C_B$.

Whereas the GSCDH problem has in general several solutions, the commutativity requirement implies by Lemma 2.2.4 that the SSCDH problem has a unique solution (dependent only on the inputs h_A and h_B).

Definition 2.2.8. Let X be a monogenic A -set with generator g , and let C_A, C_B be commuting subsets of A . For $h_A = a.g$ and $h_B = b.g$ with $a \in C_A$ and $b \in C_B$ define

$$\text{DH}(h_A, h_B) := ab.g = ba.g.$$

The resulting map $\text{DH} : C_A.g \times C_B.g \rightarrow X$ is called the *semigroup action Diffie-Hellman function*.

Note that if the semigroup A is commutative, the SCDH problem (see Definition 2.2.2), the GSCDH problem, and the SSCDH problem with respect to $C_A = A$ and $C_B = A$ are the same.

We remark that in general a solution to the GSCDH instance (h_A, h_B) can be deduced from a solution to the SDL problem for h_A , since $ab.g = a.h_B$. On the other hand, for the SSCDH problem solutions to the SDL problem might not be of any help.

2.2.2 Problems in related semigroup actions

We consider modified and extended semigroup actions, and compare the difficulty of the problems there with the original ones.

Modified semigroup actions

Let $\rho : A \times X \rightarrow X$ be an action of a semigroup A on a set X . We consider the following modifications of the semigroup action.

- (a) Replacement of the semigroup A by an isomorphic semigroup A' .
- (b) Replacement of the set X by an isomorphic A -set X' .

We will see that the hardness of the semigroup action problems may depend on the isomorphisms if they are not efficiently computable in both directions. This motivates the importance of clarifying how the sets A and X are represented.

For modification (a), let $\psi : A' \rightarrow A$ be a semigroup isomorphism and consider the derived action of A' on X , given by

$$\rho' : A' \times X \rightarrow X, \quad \rho'(a', x) := \rho(\psi(a'), x).$$

A generator g for the A -set X is also a generator for X as an A' -set X . We suppose that ρ' is like ρ efficiently computable (this is true e.g. if ψ is efficiently computable).

The difficulty of the SDL problems in ρ and ρ' are in general not comparable: If $h \in X$ is given, solutions a to the SDL problem in A , i.e. elements $a \in A$ with $h = a \cdot g$, correspond to solutions $\psi^{-1}(a)$ to the SDL problem in A' , but the function ψ^{-1} may not be efficiently computable.

However, the SCDH problems² in ρ and ρ' are equivalent: If $h_A, h_B \in X$ are given, any solution k to the SCDH problem in A , i.e. an element k such that $h_A = a \cdot g$, $h_B = b \cdot g$ and $k = a \cdot b \cdot g$ for some $a, b \in A$, is also a solution to the SCDH problem in A' .

Example 2.2.9. Let G be a cyclic group of order n (with efficiently computable group operation), and let ρ be the group operation. Any $g \in G$ is a generator for this action. The SDL problem in ρ is easy, since $a = h g^{-1}$ is a solution to the instance h . Hence, the SCDH problem is also easy.

Now let z be a generator of the cyclic group G and let ψ be the group isomorphism given by

$$\psi : \mathbb{Z}_n \rightarrow G, \quad a \mapsto z^a.$$

²This paragraph applies to both versions of the SCDH problem, namely the GSCDH and the SSCDH problem.

This leads to the action of $(\mathbb{Z}_n, +)$ on G given by $\rho'(a, x) := z^a x$. The SDL problem in ρ' now asks for a when given g and $z^a g$. This is equivalent to the discrete logarithm problem in the group G (and thus may be hard).

But the SCDH problem remains easy in ρ' , since for the instance (h_A, h_B) a solution is given by $k = h_A h_B g^{-1}$. Indeed, if $h_A = a \cdot g = z^a g$, $h_B = b \cdot g = z^b g$, then

$$h_A h_B g^{-1} = z^a g z^b g g^{-1} = z^{a+b} g = (a + b) \cdot g.$$

Regarding modification (b), let ρ' be an (efficiently computable) action of A on a set X' , and let $\varphi : X \rightarrow X'$ be an isomorphism of A -sets. This means that φ is bijective and $\varphi(a \cdot x) = a \cdot \varphi(x)$ for all $a \in A$ and $x \in X$. If X is monogenic with generator g , then X' is also monogenic, with generator $g' = \varphi(g)$.

In this case, neither the SDL problem nor the SCDH problem in ρ and ρ' are equivalent in general: An instance h' of the SDL problem in ρ' corresponds to the instance $\varphi^{-1}(h')$ in ρ , but the function φ^{-1} may not be efficiently computable. Similarly, an instance (h'_A, h'_B) of the SCDH problem in ρ' corresponds to the instance $(\varphi^{-1}(h'_A), \varphi^{-1}(h'_B))$ of the SCDH problem in ρ , and the solution k in ρ corresponds to the solution $\varphi(k)$ in ρ' , but the functions φ and φ^{-1} may not be efficiently computable.

Example 2.2.10. Let A be the semigroup (\mathbb{Z}_n, \cdot) , and let ρ be the action of A on $X = A$ given by the semigroup operation. Any $g \in \mathbb{Z}_n^*$ is a generator for this action. The SDL problem in ρ is easy, since $a = h g^{-1}$ is a solution to the instance h . Hence, the SCDH problem is also easy.

Now let G be a cyclic group of order n and consider the action ρ' of A on $X' = G$ by exponentiation, see Example 2.1.5. For every generator z of G the map

$$\varphi : \mathbb{Z}_n \rightarrow G, \quad x \mapsto z^x,$$

is an isomorphism of A -sets. But the SDL and the SCDH problems in ρ' are the usual discrete logarithm and Diffie-Hellman problems in the group G (and thus may be hard).

Extended semigroup actions

Definition 2.2.11. Let (A, \cdot) be a subsemigroup of a semigroup (\hat{A}, \cdot) , let X be a subset of a set \hat{X} , and let $\rho : A \times X \rightarrow X$ and $\hat{\rho} : \hat{A} \times \hat{X} \rightarrow \hat{X}$ be semigroup actions.

The action $\hat{\rho}$ is called an *extension* of ρ if $\hat{\rho}(a, x) = \rho(a, x)$ for all $a \in A$ and $x \in X$.

Remark 2.2.12. If A is not a subset of \hat{A} , but there exists a semigroup monomorphism $\iota : A \hookrightarrow \hat{A}$ that is efficiently computable in both directions,

then we can identify (also with respect to computational issues) A with $\iota(A)$, which is a subsemigroup of \hat{A} . In this way, one can also consider extensions of semigroup actions $A \times X \rightarrow X$ to $\hat{A} \times \hat{X} \rightarrow \hat{X}$.

If X is a monogenic A -set and if $g \in X$ is a generator of X , then we may assume that $\hat{X} = \hat{A} \cdot g$, so that \hat{X} is a monogenic \hat{A} -set with generator g .

The semigroup action problems are in general easier in the extended semigroup, since the “exponent space” A will be enlarged to \hat{A} . However, often one cannot use a solution of the problem in the extended semigroup action for the original problem. We discuss the situation for the SDL and the GSCDH problems.

Remark 2.2.13. Consider an instance of the SDL problem with respect to ρ : Given $h \in X$, find $a \in A$ such that $h = a \cdot g$.

This SDL problem instance might be easier in the extended action $\hat{\rho}$. Any element \hat{a} in the extended semigroup \hat{A} such that $h = \hat{a} \cdot g$ solves the problem. But this helps only in the case when \hat{a} is also in A .

Example 2.2.14. Let \hat{A} be the monoid $T(X)$ of all maps $X \rightarrow X$ and let A be a subsemigroup of \hat{A} , both acting naturally on X . Suppose there exists $g \in X$ with $A \cdot g = X$. Given $h \in X$, it is easy to find some map $\hat{a} \in \hat{A}$ with $\hat{a}(g) = h$ (take e.g. the transposition $\hat{a} = (gh)$), but it might be hard to find a particular map a in the subsemigroup A such that $a(g) = h$.

Remark 2.2.15. Consider an instance of the GSCDH problem with respect to ρ : Given $h_A, h_B \in X$, where $h_A = a \cdot g$ for some $a \in A$, find the element $a \cdot h_B$.

This GSCDH problem instance might again be easier in the extended action $\hat{\rho}$. Any $\hat{a} \cdot h_B$ for some $\hat{a} \in \hat{A}$ with $h_A = \hat{a} \cdot g$ will be a solution. But $\hat{a} \cdot h_B$ will in general not be a solution for the original problem.

However, if $h_B = \hat{b} \cdot g$ for some $\hat{b} \in \hat{A}$ and $h_A = a \cdot g$ for some $a \in A$ such that \hat{b} commutes with both \hat{a} and a , then $\hat{a} \cdot h_B$ will be a solution. Indeed we have $\hat{a} \cdot h_B = \hat{a} \hat{b} \cdot g = a \hat{b} \cdot g = a \cdot h_B$ by Lemma 2.2.4 (1).

2.2.3 Two-sided group actions

Several proposals of semigroup action based cryptosystems [Maz03, MMR07, SU06] use a particular kind of a two-sided semigroup action. In this section we study two-sided actions (see Example 2.1.8 (1)) in general, but restrict ourselves to group actions:

Example 2.2.16. Let G be a group and G^{op} its dual group. We consider the two-sided action of the group $A = G \times G^{op}$ on the set $X = G$, given by

$$(a_1, a_2) \cdot x := a_1 \cdot x \cdot a_2.$$

The SDL problem in this action is easy. Indeed, the solutions to the instance h with respect to a generator g are given by

$$(a_1, a_2) = (u g^{-1}, u^{-1} h),$$

where $u \in G$. Particular solutions are for example (g^{-1}, h) and $(h g^{-1}, 1)$.

Remark 2.2.17. If we restrict the action to a subgroup A of $G \times G^{op}$ the SDL problem might become harder. We discuss two examples.

- (1) $A = H_1 \times H_2$, where $H_1 \leq G$ and $H_2 \leq G^{op}$ are subgroups.

A solution $(u g^{-1}, u^{-1} h)$ to the SDL problem is in A if and only if $u g^{-1} \in H_1$ and $u^{-1} h \in H_2$, which is equivalent to $u \in H_1 g \cap h H_2$. It appears to be difficult in general to find such an u .

- (2) $A = \{(a, a^{-1}) \mid a \in G\}$.

The subgroup A is isomorphic to G and the corresponding action of G on G is the usual group conjugation, see Example 2.1.6, (3). A solution $(u g^{-1}, u^{-1} h)$ to the SDL problem is in A if and only if $u^{-1} h = g u^{-1}$, i.e. $h = u g u^{-1}$. Thus we have to find a conjugator, which appears also to be hard in general.

As noted in Remark 2.2.15 and Section 2.2.1, when a semigroup action computational Diffie-Hellman (SCDH) instance is given we can use a solution in an extended action only if some commutativity condition is satisfied. We present an example where an extension to a commutative semigroup action helps indeed to solve the SCDH problem.

Example 2.2.18. Let $G = S_n = S(M)$ be the symmetric group, i.e. the group of bijections of the set $M = \{1, 2, \dots, n\}$, and let $H \leq G$ be an abelian subgroup. Consider the two-sided action of $H \times H$ on G , given by

$$(a_1, a_2) \cdot x := a_1 \cdot x \cdot a_2.$$

Then, the SCDH problem for this group action appears to be easy.

Outline of the argument. Let K be a maximal abelian subgroup of $G = S(M)$ containing H , and consider the extended two-sided group action of $K \times K$ on G . We will sketch a method to solve the SDL problem in the extended action. This will enable us to solve the SCDH problem in the extended action, and since K is abelian, this solution will also be valid in the original action, see Remark 2.2.15.

Thus it suffices to consider the SDL problem in the case when H is a maximal abelian subgroup of $S(M)$. The maximal abelian subgroups of $S(M)$ are described by an article of Winkler, see [Win93, Theorem 1]. According to the article, there is a partition $P = \{C_1, \dots, C_r\}$ of $M = \{1, 2, \dots, n\}$ and there are abelian group operations $+_i$ on C_i for every i such that the

following holds: If we define an action of the abelian group $C_1 \times \cdots \times C_r$ on the set $M = C_1 \cup \cdots \cup C_r$ by setting

$$(c_1, \dots, c_r) \cdot x = c_i +_i x$$

whenever $x \in C_i$, then H is the image of the corresponding (injective) group homomorphism $C_1 \times \cdots \times C_r \rightarrow S(M)$. We remark that the orbits of this action are exactly the classes C_i of the partition P .

Now consider an instance of the SDL problem. That is, $g, h \in S(M)$ are given such that for some $a_i, b_i \in C_i$ we have

$$h(x) = (a_1, \dots, a_r) \cdot g((b_1, \dots, b_r) \cdot x)$$

for any $x \in M$. Then we can use the following information to get (b_1, \dots, b_r) .

- (1) Suppose $x \in C_i$ and $h(x) \in C_j$. Then $(b_1, \dots, b_r) \cdot x = b_i +_i x$ and $h(x) = a_j +_j g(b_i +_i x)$, so that $g(b_i +_i x) \in C_j$. Consequently,

$$b_i +_i x \in g^{-1}(C_j) \cap C_i.$$

- (2) Suppose $x \in C_{i_1}$ and $y \in C_{i_2}$ are such that $h(x), h(y) \in C_j$. Then their difference eliminates a_j , so that

$$h(x) -_j h(y) = g(b_{i_1} +_{i_1} x) -_j g(b_{i_2} +_{i_2} y).$$

Heuristically, to get information on the b_i (1) will be useful if there are many small classes C_i , and (2) will be useful if there are few large classes.

Once (b_1, \dots, b_r) is found, we can compute $a_j = h(x) -_j g(b_i +_i x)$ for any $x \in C_i$ with $h(x) \in C_j$ to get (a_1, \dots, a_r) . \square

2.3 Cryptosystems

We present cryptosystems based on semigroup actions in an appropriate asymptotic setting. First we adapt the notion of a group family, Definition 1.5.6, for semigroup actions.

Definition 2.3.1. A *family of semigroup actions* $(A, X) = \{(A_i, X_i)\}$ is a set of semitransitive actions ρ_i of a semigroup (A_i, \cdot_i) on a set X_i , where i ranges over an infinite index set $I \subseteq \{0, 1\}^*$. We assume that $A_i, X_i \subseteq \{0, 1\}^*$ and both the operation \cdot_i and the action ρ_i are efficiently computable.

An *instance generator* IG is an efficient algorithm that on input 1^k outputs some random index i and a generator g for the monogenic A_i -set X_i . The pair (i, g) is called a *semigroup action instance*.

For semigroup action based cryptosystems it is often necessary to construct pairs of semigroup elements with a certain commutativity property.

Definition 2.3.2. Let (A, X) be a family of semigroup actions. A *pair of compatible key generators* (K_A, K_B) consists of efficient algorithms K_A, K_B which for a given semigroup action instance (i, g) output random elements in A_i , such that if K_A outputs a and if K_B outputs b then always $a \cdot b \cdot g = b \cdot a \cdot g$.

Equivalently, the output range C_A of $K_A(i, g)$ and the output range C_B of $K_B(i, g)$ are subsets of A_i satisfying the commutativity condition

$$a b \sim_g b a$$

for all $a \in C_A$ and $b \in C_B$.

Example 2.3.3. Let $G = \{G_i\}$ be a group family. For every i consider the exponentiation map

$$\mathbb{Z}_{n_i} \times G_i \rightarrow G_i, \quad (a, x) \mapsto a \cdot x := x^a,$$

where $n_i := |G_i|$, which is a semitransitive action of the semigroup $A_i := (\mathbb{Z}_{n_i}, \cdot)$ on the set $X_i := G_i$. Since the group operation of G_i is efficiently computable, this action is also efficiently computable, see Remark 1.5.7. Furthermore, since the group size $n_i = |G_i|$ can be determined efficiently, the semigroup operation of $(\mathbb{Z}_{n_i}, \cdot)$ is also efficiently computable. Thus, $(A, X) = \{(\mathbb{Z}_{n_i}, G_i)\}$ is a semigroup action family.

An instance generator IG for the group family is also an instance generator for the corresponding semigroup action family in a natural way.

A pair (K, K) of compatible key generators is given by the algorithm K that on input (i, g) outputs random elements distributed uniformly on $A_i = \{0, 1, \dots, |G_i| - 1\}$.

The next definition states the analog of the decision Diffie-Hellman (DDH) assumption, Definition 1.5.16, for a family of semigroup actions. In view of Definition 2.2.7, it formulates the intractability of a “special semigroup action decision Diffie-Hellman (SSDDH)” problem.

Definition 2.3.4. Let (A, X) be a family of semigroup actions with instance generator IG , and let (K_A, K_B) be a pair of compatible key generators. Let μ_k be the probability distributions on quintuples (i, g, h_A, h_B, h_C) , where

- (i, g) is distributed as $IG(1^k)$;
- $h_A = a \cdot g$ and $h_B = b \cdot g$, where $a \in A_i$ is distributed as $K_A(i, g)$ and $b \in A_i$ is distributed as $K_B(i, g)$;
- h_C is uniformly distributed on X_i .

Let $\text{DH}(h_A, h_B)$ denote the semigroup action Diffie-Hellman function. The **semigroup action decision Diffie-Hellman assumption** (SDDH assumption) is satisfied if every efficient algorithm D has negligible advantage $\Delta_D(k)$, defined as the difference

$$\mathbf{P}_{\mu_k}(D(i, g, h_A, h_B, \text{DH}(h_A, h_B)) = 1) - \mathbf{P}_{\mu_k}(D(i, g, h_A, h_B, h_C) = 1).$$

Loosely speaking, the SDDH assumption states that triples of the form $(h_A, h_B, \text{DH}(h_A, h_B))$ are indistinguishable from triples of the form (h_A, h_B, h_C) .

Remark 2.3.5. A necessary condition for the SDDH assumption to be satisfied is that for every (i, g) the distribution of $\text{DH}(h_A, h_B)$ is computationally indistinguishable from the uniform distribution on X_i ; here, the distributions of h_A and h_B are induced by the key generators K_A and K_B as in the definition above.

If $A_i = (\mathbb{Z}_{n_i}, \cdot)$ acts on a cyclic group G_i as in Example 2.3.3, the SDDH assumption is the usual DDH assumption. In Remark 1.5.17 (1) we have seen that if the order $n = n_i$ contains a small prime factor p , then the distribution of $\text{DH}(h_A, h_B)$ is distinguishable from the uniform distribution on \mathbb{Z}_n . The reason for this was that the product $ab \in \mathbb{Z}_n$ is more likely to be divisible by p than a random element $c \in \mathbb{Z}_n$.

A similar situation occurs in any finite monoid A with a significant fraction of noninvertible elements $E := A \setminus A^*$. In this case $ab \in A$ is more likely to be noninvertible than a random element $c \in A$, since $ab \in E$ whenever $a \in E$ or $b \in E$. This may lead to an effective distinguisher between the distributions of $ab \cdot g$ and $c \cdot g$. To avoid this phenomenon one should choose a monoid A that has only very few noninvertible elements, so that, loosely speaking, it is “close to a group”.

If we consider group actions, there are examples where $\text{DH}(h_A, h_B)$ is exactly uniformly distributed. More precisely, let a group A act transitively on a set X , and let C_A and C_B be commuting subgroups of A , i.e. $ab = ba$ for $a \in C_A$ and $b \in C_B$, and suppose $A = C_A C_B$. In this case A is isomorphic to $(C_A \times C_B)/N$, where N is the kernel of the group epimorphism

$$C_A \times C_B \rightarrow A, \quad (a, b) \mapsto ab.$$

Thus, if $a \in C_A$ and $b \in C_B$ are uniformly distributed, then ab is uniformly distributed on A . Now, for any $g \in X$ we have $X \cong A/\ell \text{Stab}(g)$ by Remark 2.1.4, and hence $ab \cdot g$ is uniformly distributed on X .

2.3.1 Semigroup action Diffie-Hellman key agreement

We present a generalization of the Diffie-Hellman key-agreement protocol, Cryptosystem 1.5.12, to the context of semigroup actions. It is a more complex version of the protocol for commutative semigroup actions presented in [Mon02, Protocol 4.2] and [MMR07, Protocol 2.1].

Cryptosystem 2.3.6. Let (A, X) be a family of semigroup actions with instance generator IG , and let (K_A, K_B) be a pair of compatible key generators. The following *semigroup action Diffie-Hellman key agreement* protocol establishes a key e shared by two parties, A and B, communicating over an insecure, but authenticated channel.

- During setup phase an index $i \in I$ and a generator $g \in X_i$ is selected and published by applying $IG(1^k)$, where k is the security parameter.
- A uses K_A to choose a random element $a \in A_i$ and sends $h_A = a \cdot g \in X_i$ to B, retaining a secretly.
- B uses K_B to choose a random element $b \in A_i$ and sends $h_B = b \cdot g \in X_i$ to A, retaining b secretly.
- A computes $a \cdot h_B$ and B computes $b \cdot h_A$. Their common key is $e = ab \cdot g = ba \cdot g \in X_i$.

Speaking informally, a key agreement protocol is *secure* (in the presence of an eavesdropper) if the agreed key is indistinguishable from a random key, even if one is given all transmitted protocol messages.

Remark 2.3.7. It is not hard to show that the semigroup action Diffie-Hellman key agreement protocol is secure if and only if the SDDH assumption is satisfied. The proof method is similar to the security proof of the semigroup action ElGamal encryption scheme, which is carried out in detail in the next section (Proposition 2.3.9).

2.3.2 Semigroup action ElGamal encryption

We present a generalization of the ElGamal encryption scheme, Cryptosystem 1.5.18, to the context of semigroup actions, and show that under the SDDH assumption the cryptosystem is secure under a chosen plaintext attack. It is a more complex version of the cryptosystem for commutative semigroup actions presented in [Mon02, Protocol 4.3].

Cryptosystem 2.3.8. Let (A, X) be a family of semigroup actions, and suppose that for every $i \in I$ there exists an efficiently computable group operation \oplus on X_i . Let IG be an instance generator for (A, X) , and let (K_A, K_B) be a pair of compatible key generators.

The *semigroup action ElGamal encryption scheme* is the following probabilistic encryption scheme.

- The key generator $K(1^k)$ applies the algorithms IG and K_A as sub-routines. It outputs a key pair $((i, g, h), a)$, where (i, g) is a semigroup action instance distributed as $IG(1^k)$, $a \in A_i$ is a key distributed as $K_A(i, g)$, and h is the element $a \cdot g \in X_i$.

The public key is $e = (i, g, h)$, the private key is $d = a$.

- The encryption of a message $m \in X_i$ is $E_{(i,g,h)}(m) = (b \cdot g, m \oplus b \cdot h)$, where the algorithm K_B is used to choose the random $b \in A_i$.
- The decryption of a ciphertext (c_1, c_2) is $D_a(c_1, c_2) = c_2 \ominus a \cdot c_1$.

Note that the decryption works, since for every a, b, m we have

$$D_d(E_e(m)) = D_a(b \cdot g, m \oplus b \cdot h) = m \oplus b a \cdot g \ominus a b \cdot g = m.$$

Proposition 2.3.9. *Let (A, X) be a family of semigroup actions with instance generator IG and let (K_A, K_B) be a pair of compatible key generators. If the SDDH assumption holds, then the semigroup action ElGamal encryption scheme is secure under a chosen plaintext attack.*

Proof. Let A be an efficient adversary. Let $\varepsilon_A(k)$ be its advantage in the indistinguishability experiment of Definition 1.3.7. We will use A to construct an efficient distinguisher D for the SDDH problem having the same advantage $\Delta_D(k) = \varepsilon_A(k)$. By the SDDH assumption this advantage is negligible, and hence the encryption scheme is secure.

Let (i, g, h_A, h_B, h_C) be the input of the distinguisher D . That is, (i, g) is a semigroup action instance distributed as $IG(1^k)$; $h_A = a \cdot g$ and $h_B = b \cdot g$ are elements in X_i , where $a \in A_i$ is distributed as $K_A(i, g)$ and $b \in A_i$ is distributed as $K_B(i, g)$; h_C is an element uniformly distributed on X_i . The algorithm D has to simulate the challenger in the indistinguishability experiment. It interacts with A as follows:

- (1) D publishes (i, g, h_A) as the public key;
- (2) A chooses two messages $m_0, m_1 \in X_i$ and sends them to D ;
- (3) D chooses a bit $\beta \in \{0, 1\}$ uniformly at random and sends $(h_B, m_\beta \oplus h_C)$ as the ciphertext c of the message m_β to A ;
- (4) A guesses the bit β , and D outputs 1 if and only if the guess is correct.

Note that A can perform a chosen plaintext attack, since the public encryption key is known. The behaviour of A depends on the input quintuple of D . There are two cases, depending on h_C .

- (a) It holds $h_C = \text{DH}(h_A, h_B)$. Then $h_C = b \cdot h_A$, and in A 's view D performed exactly like a challenger who is using the cryptosystem. By assumption A guesses correctly with probability $\frac{1}{2} + \varepsilon_A(k)$.
- (b) The element h_C is uniformly distributed on X_i (and is independent of h_A and h_B). In this case D did not perform like a challenger who is using the cryptosystem properly, since c is not an encryption of m_β . However, since $m_\beta \oplus h_C$ is uniformly distributed on X_i , as h_C is, no information about m_β is revealed. Thus the adversary A can guess correctly only with probability $\frac{1}{2}$.

We see that $\mathbf{P}(D(i, g, h_A, h_B, \text{DH}(h_A, h_B)) = 1) = \frac{1}{2} + \varepsilon_A(k)$ and $\mathbf{P}(D(i, g, h_A, h_B, h_C) = 1) = \frac{1}{2}$, thus the advantage $\Delta_D(k)$ of the distinguisher D equals $\varepsilon_A(k)$. \square

Remark 2.3.10. The converse of the proposition can also be shown, namely the security of the semigroup action ElGamal encryption scheme implies the SDDH assumption.

We sketch the proof. Given a distinguisher D for the SDDH problem, we construct an adversary A , which acts in the indistinguishability experiment as follows. Given the public key (i, g, h) , A chooses $m_0 = 0$ (the neutral element of the group (X_i, \oplus)) and $m_1 \in X_i$, uniformly at random. Upon receiving an encryption $(c_1, c_2) = (b \cdot g, m_\beta \oplus b \cdot h)$, A guesses $\beta = 0$ if and only if $D(i, g, h, c_1, c_2) = 1$. It is easy to see that if D has advantage $\Delta_D(k)$, then A has the same order of advantage, namely $\frac{1}{2}\Delta_D(k)$.

The semigroup action ElGamal encryption scheme is, like classical ElGamal encryption, vulnerable to a chosen ciphertext attack: Suppose the encryption

$$(c_1, c_2) = (b \cdot g, m \oplus b \cdot h)$$

of a message m is given, then for any m' one can apply the decryption oracle to the ciphertext

$$(c_1, m' \oplus c_2) = (b \cdot g, m' \oplus m \oplus b \cdot h)$$

to find out $m' \oplus m$ and thus m .

2.3.3 Identification protocols and digital signatures

We present semigroup action identification protocols, which are based on a zero-knowledge (ZK) proof of knowledge of a solution of the semigroup action discrete logarithm (SDL) problem. Then we create a digital signature scheme which is based on one of these identification protocols.

Provided that the SDL problem is intractable, knowledge of an SDL solution is indeed nontrivial knowledge, i.e. it cannot be efficiently deduced from public information. However, it appears that ZK proofs of knowledge of a discrete logarithm in a group cannot be transferred to general semigroup actions. Therefore, we will require more restricted settings.

Stated below is the general principle how to use ZK proofs of knowledge of an SDL solution inside an identification protocol.

Cryptosystem 2.3.11. Let (A, X) be a family of semigroup actions with instance generator IG (satisfying the SDL assumption), and let π be a ZK proof of knowledge of an SDL solution. The *semigroup action ZK-based identification* protocol is the following:

- During setup phase a semigroup action instance (i, g) is selected and published by applying $IG(1^k)$, where k is the security parameter. Each claimant P chooses a private key $a \in A_i$ and publishes $h = a \cdot g$ as its public key.
- P identifies itself to a verifier V by proving knowledge of its private key a using the ZK proof π .

We present two zero-knowledge proofs of knowledge of a solution to the SDL problem. The first is analogous to the Fiat-Shamir protocol [FS87] and was suggested in the context of braid groups [SDG02, Deh04]. We state the protocol for general semigroup actions, but prove its properties only for group actions.

Protocol 2.3.12. Let (A, X) be a family of semitransitive semigroup actions and let R be the SDL function problem consisting of all pairs $((i, g, a \cdot g), a)$, where (i, g) is a semigroup action instance and $a \in A_i$. A prover P demonstrates to a verifier V knowledge of a solution $a \in R(x)$ to the SDL problem instance $x = (i, g, h)$, i.e. $h = a \cdot g$, by executing the following subprotocol $k = |x|$ times.

(Commitment) P sends $r = b \cdot h \in X_i$, where $b \in A_i$ is chosen uniformly at random³,

(Challenge) V sends a bit $c \in \{0, 1\}$, chosen uniformly at random,

(Response) P sends $\ell = \begin{cases} b & \text{if } c = 0, \\ b a & \text{if } c = 1. \end{cases}$

At the end the verifier V accepts if and only if each time it holds $r = \ell \cdot h$ in the case $c = 0$ and $r = \ell \cdot g$ in the case $c = 1$.

The following proposition requires group actions. It is an open problem to examine weaker conditions under which the properties of a ZK proof of knowledge are preserved.

Proposition 2.3.13. *If A_i is a group then Protocol 2.3.12 is a zero-knowledge proof of knowledge of an SDL solution.*

Proof. Let $x = (i, g, h)$ be an SDL problem instance. If $a \in R(x)$ is a solution, i.e. $h = a \cdot g$, then $r = b \cdot h = b a \cdot g$, and hence P 's answers will be correct for both $c = 0$ and $c = 1$. Thus the protocol is complete.

For the soundness condition, let B be an algorithm impersonating the prover interacting with V . For each of the $k = |x|$ rounds the message-specification function of B specifies triples (r, ℓ_0, ℓ_1) , where $r \in X_i$ is a commitment and $\ell_0, \ell_1 \in A_i$ are the answers to the challenges $c = 0, 1$.

³We assume here that uniform random drawing from A_i can be done efficiently.

These triples may depend on the challenges in previous rounds. Let us call a triple *correct* if V would accept both answers, i.e. it holds

$$\ell_0 \cdot h = r = \ell_1 \cdot g.$$

In this case, since ℓ_0 is left invertible, it follows $\ell_0^{-1} \ell_1 \cdot g = \ell_0^{-1} \ell_0 \cdot h = h$, hence $\ell_0^{-1} \ell_1 \in R(x)$ is a valid solution. Now the knowledge extractor K examines the triples of B and outputs the solution if it finds a correct triple. If however the answers in the triple are correct for only one challenge c , then K lets B 's future triples depend on this challenge c . Let p be the probability that V accepts and let s be K 's success probability; we will show $s \geq p - \frac{1}{2^k}$. By averaging it suffices to consider an execution of B with its internal coin tosses fixed. Now if K examines no correct triple of B then there is at most one sequence of challenges letting V accept, i.e. $p \leq \frac{1}{2^k}$; in the other case $s = 1$. Hence we have $s \geq p - \frac{1}{2^k}$ in any case.

For the zero-knowledge property note that the transcripts of the protocol are $(b \cdot h, 0, b)$ if $c = 0$ and $(b \cdot h, 1, ba) = (ba \cdot g, 1, ba)$ if $c = 1$. Since ba is uniformly distributed on A_i as b is, it is clear that one can efficiently generate, without interaction, an output which is indistinguishable from a protocol transcript. \square

The second ZK proof of knowledge of an SDL solution is derived from the Schnorr identification protocol, Cryptosystem 1.5.21. We state the protocol for general semimodules over a semiring, but prove its properties only for particular modules over a ring.

Protocol 2.3.14. Let (A, X) be a family of monogenic semimodules X_i over semirings A_i and let R be the SDL function problem consisting of all pairs $((i, g, a \cdot g), a)$, where (i, g) is a semimodule instance and $a \in A_i$. A prover P demonstrates to a verifier V knowledge of a solution $a \in R(x)$ to the problem instance $x = (i, g, h)$, i.e. $h = a \cdot g$, by executing the following protocol.

(Commitment) P chooses randomly $b \in A_i$ and sends $r = b \cdot g \in X_i$,

(Challenge) V sends a random $c \in A_i$,

(Response) P sends $s = b + ca \in A_i$.

The verifier V accepts if and only if $s \cdot g = r + c \cdot h$.

The choice of b and c is according to the uniform distribution on A_i .

Proposition 2.3.15. *Let X_i be modules over rings A_i having a negligible fraction of nonunits. Then Protocol 2.3.14 is a zero-knowledge proof of knowledge of an SDL solution.*

Proof. Let $x = (i, g, h)$ be an SDL problem instance. If $a \in R(x)$ is a solution, i.e. $h = a \cdot g$, then $r + c \cdot h = (b + ca) \cdot g$, and hence V will accept the proof. Thus the protocol is complete.

Now suppose B is an algorithm impersonating the prover interacting with V, and let p be the probability that V accepts. The knowledge extractor K having oracle access to B's message-specification function obtains B's commitment r , chooses two different random challenges $c_0, c_1 \in \mathbb{Z}_n$ and obtains B's answers s_0, s_1 . If V would accept both answers we have $s_i \cdot g = r + c_i \cdot h$ for $i = 0, 1$, and consequently $(s_1 - s_0) \cdot g = (c_1 - c_0) \cdot h$. With overwhelming probability $c_1 - c_0$ will be left-invertible, and in this case we have

$$(c_1 - c_0)^{-1} (s_1 - s_0) \cdot g = (c_1 - c_0)^{-1} (c_1 - c_0) \cdot h = h,$$

hence $(c_1 - c_0)^{-1} (s_1 - s_0) \in R(x)$ is a valid solution. For K's success probability we have $s \geq p^2 - \varepsilon(x)$, where $\varepsilon(x)$ is a negligible function in $|x|$. This shows the soundness property.

For the zero-knowledge property note that the transcript $(b \cdot g, c, b + ca)$ of the protocol is identically distributed as $(s \cdot g - c \cdot h, c, s)$, where $s \in A$ is uniformly distributed. Hence one can efficiently generate, without interacting with P, transcripts indistinguishable from the original ones. \square

As the Schnorr signature scheme is deduced from the Schnorr identification protocol by replacing the challenge by the value of a hash function, we can deduce a signature scheme from the above ZK proof of knowledge.

Cryptosystem 2.3.16. Let (A, X) be a family of monogenic A_i -modules X_i such that the rings A_i have negligible fractions of nonunits, and let IG be an instance generator. Let $H = \{H_i\}$ be a family of hash functions $H_i : \{0, 1\}^* \rightarrow A_i$. The *semigroup action Schnorr signature scheme* for G is the following probabilistic digital signature scheme.

- The key generator $K(1^k)$ uses the instance generator IG as a subroutine and outputs the key pair $((i, g, h), a)$. Here, (i, g) is a module instance distributed as $IG(1^k)$, a is uniformly distributed on A_i , and $h = a \cdot g$.

The public key is $e = (i, g, h)$, the private key is $d = a$.

- The signature of a message m is

$$S_a(m) = (c, s) = (H_i(m, r), b + ac) \in A_i^2,$$

where b is a random element distributed uniformly on A_i , and $r = b \cdot g$.

- The verification of a signature (c, s) for a message m is

$$V_{(i,g,h)}(m, (c, s)) = \text{yes} \quad \text{if and only if} \quad H_i(m, s \cdot g - c \cdot h) = c.$$

Remark 2.3.17. Since the semigroup action Schnorr signature scheme is based on a ZK proof of knowledge its security can be proved exactly as in [PS00]. Precisely, provided that the module family (A, X) satisfies the SDL assumption, the signature scheme is existentially unforgeable under an adaptive chosen-message attack.

2.4 Semigroup action based cryptosystems in the literature

In this section we give an overview of some cryptosystems proposed in the last decade that use (semi-)group actions. A frequently used concept is the conjugation in certain nonabelian groups.

2.4.1 Cryptosystems using the modular group

The modular group is a fundamental object of study in number theory, geometry and algebra. It was used by Yamamura [Yam98, Yam99] to construct public-key encryption schemes. Although both proposed cryptosystems have been very successfully attacked by Blackburn and Galbraith [BG99], they are still worth mentioning because they seem to be the first cryptosystems using several group theory concepts, like presentations, group actions and conjugated elements, which are used in subsequent proposals of group-based cryptosystems.

The Yamamura encryption schemes use an action of the modular group $\mathrm{SL}_2(\mathbb{Z})$ on the upper halfplane by Möbius transformations:

Example 2.4.1. Every element

$$M \in \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{GL}_2(\mathbb{C})$$

in the general linear group defines a Möbius transformation

$$f_M : z \mapsto \frac{az + b}{cz + d}$$

of the extended complex plane $\hat{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$, and the corresponding map $\mathrm{GL}_2(\mathbb{C}) \rightarrow \mathrm{Aut}(\hat{\mathbb{C}})$, $M \mapsto f_M$ is a group homomorphism. Hence there is a group action of $\mathrm{GL}_2(\mathbb{C})$ on $\hat{\mathbb{C}}$, given by

$$M . z := f_M(z).$$

The *modular group* G is the subgroup $\mathrm{SL}_2(\mathbb{Z})$ of $\mathrm{GL}_2(\mathbb{C})$ consisting of matrices over \mathbb{Z} with determinant 1. Let \mathcal{H} be the upper halfplane

$\{z \in \mathbb{C} \mid \text{Im } z > 0\}$. The Möbius transformations associated to matrices in G preserve \mathcal{H} , so that G acts on \mathcal{H} .⁴

The set $\mathcal{F} := \{z \in \mathcal{H} \mid |z| \geq 1, |\text{Re } z| \leq \frac{1}{2}\}$ is called the *standard fundamental domain*. It intersects every orbit in at least one point, and in at most one point in the interior.

Another representation of the modular group is the presentation

$$G = \langle A, B \mid A^4 = I = B^6, A^2 = B^3 (= -I) \rangle,$$

where

$$A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix}.$$

We can thus characterize the modular group more abstractly as

$$G \cong (\mathbb{Z}_2 * \mathbb{Z}_3) \times \mathbb{Z}_2,$$

where $*$ denotes the free product and \times the cartesian product. It follows that every matrix $M \in \text{SL}_2(\mathbb{Z})$ can be uniquely written as

$$M = \pm A^\varepsilon B^{i_1} A \dots AB^{i_n} A^{\varepsilon'}$$

with $\varepsilon, \varepsilon' \in \{0, 1\}$ and $i_1, \dots, i_n \in \{1, 2\}$, called the normal form of M .

A crucial fact for cryptographic purposes is that the action of the modular group on \mathcal{H} gives rise to an efficient algorithm for computing the normal form of a matrix $M \in \text{SL}_2(\mathbb{Z})$. Indeed, there is an algorithm which for a given point $M.p$, where p is any point in the interior of the fundamental domain \mathcal{F} (say $p = 2i$), computes the normal form of M up to sign. It needs linear time in the length of the normal form, see e.g. [BG99] for details. Note that this means that the semigroup action discrete logarithm (SDL) problem for the modular group action is easily solvable.

Now we present the idea of the encryption scheme in [Yam99]. Let $V_1, V_2 \in \text{SL}_2(\mathbb{Z})$ be matrices that generate a free subgroup of $\text{SL}_2(\mathbb{Z})$ and such that any word in V_1, V_2 is in normal form. For example, $V_1 = (AB)^i$, $V_2 = (AB^2)^j$ is a valid choice for any $i, j \geq 1$. Let p be a point in the interior of the fundamental domain \mathcal{F} , and choose a secret matrix $M \in \text{GL}_2(\mathbb{C})$ as the private key. The public key consists of the point $q = M^{-1}.p$ and the conjugated matrices $W_1 = M^{-1}V_1M$ and $W_2 = M^{-1}V_2M$. A message $m = (i_1, \dots, i_n) \in \{1, 2\}^n$ is then encrypted to

$$q' = W_{i_1} \dots W_{i_n}.q.$$

To decrypt a message note that $M.q' = V_{i_1} \dots V_{i_n}.p$, so that i_1, \dots, i_n can be recovered by applying the algorithm for computing the normal form.

⁴The kernel of the corresponding homomorphism $\text{SL}_2(\mathbb{Z}) \rightarrow \text{Aut}(\mathcal{H})$ equals $\{I, -I\}$, so the induced map $\text{PSL}_2(\mathbb{Z}) \hookrightarrow \text{Aut}(\mathcal{H})$ is injective, where $\text{PSL}_2(\mathbb{Z}) := \text{SL}_2(\mathbb{Z})/\{I, -I\}$. Some authors refer to $\text{PSL}_2(\mathbb{Z})$ instead of $\text{SL}_2(\mathbb{Z})$ as the modular group.

However, as Blackburn and Galbraith show [BG99, Proposition 1], the ciphertext q' lies in easily distinguishable regions, depending on the first bit i_1 , so that the plaintext can be easily recovered by a bit-by-bit computation.

The encryption scheme in [Yam98] is based on a similar idea, but uses matrices over the polynomial ring $\mathbb{C}[x]$. The transformed generators are $W_i(x) = M^{-1}F_i(x)M$, where $F_i(a) = V_i$ for some secret $a \in \mathbb{C}$. However, its cryptanalysis can be reduced to the cryptanalysis of the point-based scheme above, see [BG99].

2.4.2 Braid groups and cryptography

Problems in combinatorial group theory

Combinatorial group theory deals with groups presented by generators and relations. The idea to use them for public-key cryptography originates from the fact that many problems arising in the context of recursively presented groups are computationally very hard in general. The recent textbook [MSU08] gives a good introduction to group-based cryptography.

One of the computationally hard problems in group theory is the *word problem*, which is the problem to decide whether two given words in the generators represent the same group element. A remarkable result, proved independently by Novikov and Boone in the 1950s, states the existence of a finitely presented group with unsolvable word problem. See Rotman's book [Rot73] for an elementary proof of the Novikov-Boone theorem.

Another hard problem is the *conjugacy problem*, which asks whether two given words in the generators represent the same conjugacy class. Also in this case, there exist finitely presented groups with unsolvable conjugacy problem. The related *conjugator search problem* is the problem to find for two given words x, y representing the same conjugacy class a word a such that $y = a x a^{-1}$ holds in the group.⁵

Cryptographers have been utilizing the hardness of the conjugacy search problem for constructing public-key cryptosystems. We note that the conjugacy search problem is a special case of the semigroup action discrete logarithm (SDL) problem when the group action is taken to be the conjugation, see Example 2.1.6, (3). As a word of warning we note that the hardness discussion of problems usually refer to worst-case hardness. As noted in Section 1.4 this is not sufficient for cryptographic purposes unless there is also an efficient method to generate hard instances.

⁵We note that the conjugacy search problem is always solvable, essentially by trying each possible conjugator a . The corresponding conjugacy problem being unsolvable now means that the sought-after conjugator a may become "extremely complex", namely its word length is not bounded by a recursive function. However, for the design of secure cryptosystems this property seems not applicable, since usually the conjugator is part of the secret key and thus has to be of moderate size.

Though the conjugacy search problem should be hard for a group-based cryptosystem one requires the word problem to be efficiently solvable. This is because secret keys and ciphertexts are usually encoded as group elements. In most of the cases, this issue is handled by a unique normal form together with an efficient algorithm to convert words into their normal forms.

Braid groups

In braid groups the word problem is efficiently solvable yet the conjugator search problem is computationally hard, so they seem to be well suited for public-key cryptography. Braid groups have been appearing in the cryptography literature from the pioneering work of Anshel, Anshel, Goldfeld [AAG99] and Ko et al. [KLC⁺00] onwards. Nowadays “braid-based cryptography” remains an active area of research, see Dehornoy’s article [Deh04] for a survey.

For an integer $n \geq 2$, the *braid group* B_n on n strands is defined by the finite presentation

$$B_n := \langle \sigma_1, \dots, \sigma_{n-1} \mid \begin{array}{ll} \sigma_i \sigma_j = \sigma_j \sigma_i & \text{if } |i - j| \geq 2 \\ \sigma_i \sigma_j \sigma_i = \sigma_j \sigma_i \sigma_j & \text{if } |i - j| = 1 \end{array} \rangle.$$

Hence for example, $B_2 \cong \mathbb{Z}$, $B_3 = \langle x, y \mid xyx = yxy \rangle$ and $B_4 = \langle x, y, z \mid xz = zx, xyx = yxy, yzy = zyz \rangle$.

Note that we have natural inclusions $B_2 \hookrightarrow B_3 \hookrightarrow B_4 \hookrightarrow \dots$ and epimorphisms $\pi_n : B_n \rightarrow S_n$ onto the symmetric group, given by $\sigma_i \mapsto (i, i+1)$.

Braid groups admit a normal form for elements, called the greedy normal form. For this, define the *positive braids* to be the submonoid B_n^+ of B_n generated by $\sigma_1, \dots, \sigma_{n-1}$. Then, define $\Delta_n \in B_n^+$ inductively by $\Delta_1 = 1$ and $\Delta_{k+1} = \Delta_k \sigma_k \dots \sigma_1$. We call a braid $b \in B_n^+$ *simple*, if it is a (generalized) prefix of Δ_n , i.e. there exists $c \in B_n^+$ such that $\Delta_n = bc$ holds in B_n . One can show that the simple braids correspond bijectively to S_n via π_n . For example, $\Delta_3 = xyx$ and the simple braids of B_3 are $\varepsilon, x, y, xy, yx, xyx$.

Now the *greedy normal form* of an element $b \in B_n$ is

$$b = \Delta_n^k b_1 \dots b_r$$

where $k \in \mathbb{Z}$ and b_1, \dots, b_r are simple braids $\neq 1, \Delta_n$ such that b_i is a maximal simple prefix of $b_i \dots b_r$. The number r is called the *complexity* of the braid b .

The normal form of a word w can be determined in quadratic time, hence the word problem in braid groups can be solved efficiently. On the other hand, though the conjugator search problem is also solvable, the only solutions proposed so far have a high algorithmic complexity.

Cryptosystems using braid groups

The hardness of the conjugator search problem was used by Ko et al. [KLC⁺00] to set up a key agreement protocol. It is in fact a special case of Cryptosystem 2.3.6, where the semigroup action is taken to be the conjugation in a braid group B_{2n} , and the commuting output ranges of the key generators are $C_A = \langle \sigma_1, \dots, \sigma_{n-1} \rangle$ and $C_B = \langle \sigma_{n+1}, \dots, \sigma_{2n-1} \rangle$.

The authors also propose an encryption scheme, which is derived from the key agreement protocol. It is very similar to Cryptosystem 2.3.8, but they use a hash function to transform the key, which is a braid group element, into a binary string of the message space.

Anshel, Anshel and Goldfeld [AAG99] also proposed a key agreement protocol based on the conjugator search problem in braid groups. More precisely, its security is based on the difficulty of the *multiple conjugator search problem*, which asks when given multiple pairs (x_i, y_i) with $y_i = ax_i a^{-1}$ for the common conjugator a . The idea of this key agreement protocol is the following: Two parties, A and B, want to agree on a key $k \in B_n$, which will be composed of their secret keys $r, s \in B_n$ as

$$k = (srs^{-1})r^{-1} = s(rsr^{-1})^{-1}.$$

Hence, B has to send A some information involving s so that it can compute srs^{-1} , but an eavesdropper should not reconstruct B's secret s . For this, A publishes braids p_1, \dots, p_ℓ and B publishes braids q_1, \dots, q_m . Then A chooses a word u on the letters p_i and their inverses, and B chooses a word v on the letters q_i and their inverses. Their secrets r and s will be the braids determined by the words u and v , respectively. Now B sends $sp_1s^{-1}, \dots, sp_\ell s^{-1}$ to A, so that it can compute srs^{-1} by replacing each p_i in the word u by $sp_i s^{-1}$. Similarly, A sends $rq_1r^{-1}, \dots, rq_m^{-1}r^{-1}$ from which B can compute rsr^{-1} .

There are also schemes for identification and digital signature based on braid groups, see [Deh04].

However, the initial enthusiasm for cryptography based on braid groups was lowered due to several attacks. Many of these attacks use the complexity or the length of braid words to solve the conjugator search problem. There are exact algorithms as in [Geb06] as well as heuristic algorithms as in [HS02, MSU05]. These attacks turn out to be effective when random braid group elements are used, but one might be able to select hard instances of the conjugator search problem which withstand the proposed attacks.

2.4.3 MOR cryptosystem

Paeng et al. [PHK⁺01] used conjugation in nonabelian groups to construct a public-key encryption scheme based on the difficulty of the discrete logarithm (DL) problem in the group of inner automorphisms. They argue that

even if the DL problem in the original group is subject to subexponential attacks, the DL problem in the inner automorphism group appears to be more difficult.

This MOR cryptosystem (as it was called in a follow-up paper [PKHK01]) is similar to ElGamal encryption, see Cryptosystem 1.5.18. Its idea is the following: Let $G = \langle \gamma_1, \dots, \gamma_n \rangle$ be a group with an efficient computable normal form to express group elements as products in the generators γ_i . Then, an inner automorphism

$$\text{Inn}_g : G \rightarrow G, \quad x \mapsto gxg^{-1}$$

is uniquely determined by the values $\text{Inn}_g(\gamma_i)$ and can thus be represented by the n -tuple $(\text{Inn}_g(\gamma_i))_{i=1}^n$. Let a be a random integer. The public key consists of Inn_g and $(\text{Inn}_g)^a = \text{Inn}_{g^a}$, and the private key is the integer a . To encrypt a message $m \in G$, a random integer r is chosen and the ciphertext

$$(c, \varphi) = ((\text{Inn}_g)^{ar}(m), (\text{Inn}_g)^r)$$

is sent. With the knowledge of a the decryption can be done as $\varphi^{-a}(c) = m$.

A relevant problem for the security analysis is the *special conjugacy problem*, which for a given Inn_g asks for an element $g' \in G$ such that $\text{Inn}_{g'} = \text{Inn}_g$.⁶ Note that a solution g' is unique up to the center $Z(G)$ of G . It follows that if the special conjugacy problem is efficiently solvable, the DL problem in Inn_G can be reduced to the DL problem in the quotient group $G/Z(G)$. For this reason the authors suggested to use groups with large center to prevent a reduction to the DL problem in G .

As a platform group G Paeng et al. proposed to use the subdirect product $\text{SL}_2(\mathbb{Z}_p) \times_{\theta} \mathbb{Z}_p$, where the homomorphism θ is given by

$$\theta : \mathbb{Z}_p \rightarrow \text{Aut}(\text{SL}_2(\mathbb{Z}_p)), \quad a \mapsto (\text{Inn}_{\alpha})^a \quad \text{with } \alpha \in \text{SL}_2(\mathbb{Z}_p) \text{ of order } p.$$

The MOR cryptosystem was analyzed by Tobias [Tob02], who found several weaknesses of the original system. He showed, for example, that breaking MOR using G is not harder than breaking MOR using $\text{SL}_2(\mathbb{Z}_p)$, and pointed out that the invariance of the trace under matrix conjugation enables one to reveal partial information of the message.

2.4.4 Further problems in other groups

The decomposition problem

Besides the conjugator search problem there are other problems with some relevance for cryptology. One of these is named *decomposition problem* by several authors. It is stated as follows: Let G be a group and let C_A, C_B

⁶This is in fact equivalent to the multiple conjugator search problem, since Inn_g is given by $\text{Inn}_g(\gamma_i) = g\gamma_i g^{-1}$.

be subgroups. For given elements $x, y \in G$, where $y \in C_A \cdot x \cdot C_B$, find elements $a \in C_A$ and $b \in C_B$ such that $y = a \cdot x \cdot b$. Thus it is the semigroup action discrete logarithm (SDL) problem in a two-sided group action, see Example 2.2.16 and Remark 2.2.17 (1).

Myasnikov, Shpilrain and Ushakov [MSU05] developed a length-based algorithm to solve a decomposition problem in braid groups and used it for cryptanalysis of the key agreement protocol by Ko et. al. [KLC⁺00].

At the same time, Shpilrain and Ushakov [SU06] used the decomposition problem as a base for a new key agreement protocol.⁷ As a platform they proposed to use Thompson's group F , or, more precisely, its (infinite) presentation

$$F = \langle x_0, x_1, x_2, \dots \mid x_i^{-1} x_k x_i = x_{k+1} \text{ for } k > i \rangle.$$

They show how a normal form of a word w can be computed in almost linear time in the length of w .

Their key agreement protocol is a special case of Cryptosystem 2.3.6, where the semigroup action is the two-sided action in Thompson's group. The commuting output ranges of the key generators are $C_A = A \times B$ and $C_B = B \times A$, where A, B are commuting subgroups of F , given by

$$A = \langle x_0 x_1^{-1}, \dots, x_0 x_s^{-1} \rangle \text{ and } B = \langle x_{s+1}, x_{s+2}, \dots \rangle$$

for some s .

However, Thompson's group is vulnerable e.g. to length-based attacks, as pointed out by Ruinskiy, Shamir and Tsaban [RST07], so that this cryptosystem can be considered insecure.

Endomorphisms of Artin groups

Shpilrain and Zapata presented in [SZ06] a general idea for constructing key agreement protocols based on semigroup actions. To explain the idea, consider first the action of a group G on itself, given by conjugation

$$(a, x) \mapsto a \cdot x \cdot a^{-1}.$$

Its associated map $a \mapsto [x \mapsto a x a^{-1}]$ is a group epimorphism $G \rightarrow \text{Inn}(G)$ onto the group of inner automorphisms of G .

Instead of inner automorphisms Shpilrain and Zapata considered more general endomorphisms of G . More precisely, they considered a homomorphism $T \rightarrow \text{End}(G)$ from a semigroup T into the endomorphism monoid $\text{End}(G)$, and proposed a key agreement protocol like Cryptosystem 2.3.6 using the corresponding semigroup action $T \times G \rightarrow G$.

⁷A related idea was developed by Maze [Maz03, Section 5.4], see also [MMR07].

As for the group G they proposed Artin groups of extra large type. An *Artin group* is given by a presentation of the form

$$G = \langle a_1, \dots, a_n \mid \mu_{ij} = \mu_{ji} \text{ for } i < j \rangle,$$

where the $\mu_{ij} = x_i x_j x_i \dots$ are alternating products of x_i and x_j of length m_{ij} , beginning with x_i . Furthermore, $m_{ji} = m_{ij} \in \{2, 3, \dots, \infty\}$, with the convention that there is no relation for x_i and x_j in the case $m_{ij} = \infty$. The braid groups B_n are examples of Artin groups, with the m_{ij} taken to be $m_{ij} = 2$ if $|i - j| \geq 2$ and $m_{ij} = 3$ if $|i - j| = 1$. We call the Artin group to be of *extra large type* if $m_{ij} \geq 4$ for all (i, j) . As can be shown these admit an efficient algorithm to solve the word problem.

Artin groups can be described by a weighted graph Γ having vertices $\{a_1, \dots, a_n\}$ and edges (a_i, a_j) with weight m_{ij} for all $i < j$ with $m_{ij} < \infty$. Conversely, for every weighted graph Γ with edge values in $\mathbb{N}_{\geq 2}$ there is an associated Artin group $A\Gamma$. Furthermore, every graph endomorphism of Γ induces a group endomorphism of $A\Gamma$.

This gives a way to construct commuting endomorphisms as needed for the key generators of Cryptosystem 2.3.6. More precisely, let Γ be a rooted tree such that the root has 2 branches and the edges have weights ≥ 4 . Hence, $A\Gamma$ is an Artin group of extra large type. Let Γ_A, Γ_B be the subtrees obtained by removing the root and let $A\Gamma_A, A\Gamma_B$ be the associated subgroups of $A\Gamma$. We let T be $\text{End}(A\Gamma_A) \times \text{End}(A\Gamma_B)$. The subsets $C_A = \text{End}(A\Gamma_A) \times \{\text{id}\}$ and $C_B = \{\text{id}\} \times \text{End}(A\Gamma_B)$ will then commute, and thus can be used for constructing key generators K_A and K_B of the key agreement protocol.

However, Shpilrain and Zapata stay rather general when describing their cryptosystem. They do not give details on how to choose the tree Γ and how to select endomorphisms out of the sets C_A, C_B .

Chapter 3

Simple semirings

The main result of this chapter states that a finite semiring of order > 2 with zero which is not a ring is congruence-simple if and only if it is isomorphic to a “dense” subsemiring of the endomorphism semiring of a finite idempotent commutative monoid. We also investigate those subsemirings further, considering e.g. the question of isomorphism.

Whereas Section 3.2 and Section 3.3 deal only with semirings having a zero element, the first section introduces them more generally.

3.1 Introduction to semirings

The notion of *semiring* is a natural generalization of the notion of ring, allowing the additive substructure to be only a commutative semigroup instead of an abelian group. Since their introduction by Vandiver in 1934 [Van34], there has been an active area of research in semirings. The interest in semiring theory evolved not only because it provides a natural generalization of ring theory, but because of its value as a tool in many significant applications in mathematics, computer science, and other fields. One reason for this is that semirings provide in a sense the weakest algebraic framework so that matrix multiplication over them is associative, see Proposition 4.1.9. The reader may consult the monographs of Golan [Gol99] and Hebisch/Weinert [HW93, HW98] for more detailed information on semirings.

Definition 3.1.1. A structure $R = (R, +, \cdot)$, consisting of a set R and two binary operations $+$ and \cdot on R , is called a *semiring* if

- $(R, +)$ is a commutative semigroup,
- (R, \cdot) is a semigroup,
- both distributive laws hold:

$$x \cdot (y + z) = x \cdot y + x \cdot z \quad \text{and} \quad (x + y) \cdot z = x \cdot z + y \cdot z.$$

If the commutative semigroup $(R, +)$ is an abelian group, the semiring R is called a *ring*. Otherwise, the semiring is called a *proper* semiring.

A *subsemiring* of a semiring R is a subset $S \subseteq R$ that is closed under addition and multiplication. Naturally, S itself is a semiring. The *order* of a semiring R is its number of elements.

As usual we sometimes omit the multiplication dot, i.e. $xy := x \cdot y$.

Definition 3.1.2. Let $(R, +, \cdot)$ be a semiring.

- If a neutral element 0 of the semigroup $(R, +)$ exists and it satisfies $0x = x0 = 0$ for all $x \in R$, then it is called **zero**.
- If a neutral element 1 of the semigroup (R, \cdot) exists, it is called a **one**.

Example 3.1.3. Let R be the set $\{o, b, c\}$ with the following operations:

$+$	o	b	c	\cdot	o	b	c
o	o	b	c	o	o	o	c
b	b	b	c	b	o	b	c
c	c	c	c	c	o	c	c

It can be shown that $(R, +, \cdot)$ satisfies the axioms for a semiring. We note that the element o is neutral in the semigroup $(R, +)$, but does not satisfy $oc = o$. The element b is neutral in the semigroup (R, \cdot) . Hence R is a semiring with a one, but without a zero.

The number of finite semirings is enormous. The following table¹ compares the number of semirings having a zero with the number of rings.

Order	Semirings with 0	Rings
2	4	2
3	22	2
4	283	11
5	4'717	2
6	108'992	4
7	8'925'672	2
total	9'039'691	23

Definition 3.1.4. Let R be a semiring. We define $R^* = R \setminus \{o\}$, if R has an additive neutral o , and $R^* = R$ otherwise. The semiring R is called **semifield** if (R^*, \cdot) is a group.

See [HW98, Corollary 5.9] for a proof of the following result.

¹These figures are outputs of a self-written Java program.

Proposition 3.1.5. *Every finite semifield is a field or has order ≤ 2 .*

Example 3.1.6. The *Boolean semifield* R is the set $\{0, 1\}$ with the following operations:

$$\begin{array}{c|cc} + & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 1 \end{array} \quad \begin{array}{c|cc} \cdot & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array}$$

It is the only proper finite semifield with zero.

3.1.1 Homomorphisms, congruences, ideals

Definition 3.1.7. Let R and S be semirings. A map $f : R \rightarrow S$ is called a *homomorphism of semirings* if it preserves the semiring operations:

$$f(x + y) = f(x) + f(y), \quad f(x \cdot y) = f(x) \cdot f(y).$$

If R and S have a zero, a homomorphism of semirings f is called a *homomorphism of semirings with zero* if it preserves also the zero element:

$$f(0) = 0.$$

Recall that, by the fact known as the first isomorphism theorem for rings, every homomorphism $f : R \rightarrow S$ of rings R and S induces an isomorphism

$$\hat{f} : R / \ker f \rightarrow \text{im } f, \quad [x] \mapsto f(x)$$

of the quotient ring $R / \ker f$ of R onto the subring $\text{im } f$ of S .

The situation is different for general semirings. Consider, for example, a homomorphism $f : R \rightarrow S$ of semirings with zero. The set of equivalence classes $[x] := f^{-1}(f(x))$ can in general not be described by the “kernel” $f^{-1}(0)$ of f . Instead, the equivalence relations induced by semiring homomorphisms are described by congruences.

Definition 3.1.8. Let R be a semiring. An equivalence relation \sim on R is called (*semiring*) *congruence* if it respects the semiring operations:

$$x \sim y \quad \text{implies} \quad a + x \sim a + y, \quad ax \sim ay, \quad xa \sim ya.$$

We note that every semiring R has at least two congruences, namely

- the equality relation $\sim = \text{id}_R$, defined by $x \sim y \Leftrightarrow x = y$,
- the total relation $\sim = R \times R$, where $x \sim y$ for all x, y .

Remark 3.1.9. Semiring congruences are related to semiring homomorphisms in the following way.

- (1) For every homomorphism $f : R \rightarrow S$ of semirings R and S , the equivalence relation \sim_f induced by f with classes $[x] = f^{-1}(f(x))$, i.e.

$$x \sim_f y \quad :\Leftrightarrow \quad f(x) = f(y),$$

is a congruence.

- (2) Given a congruence \sim on a semiring R , we can define operations $+$ and \cdot on its set of equivalence classes $R/\sim = \{[x] \mid x \in R\}$ by

$$[x] + [y] := [x + y] \quad \text{and} \quad [x] \cdot [y] := [x \cdot y],$$

turning $(R/\sim, +, \cdot)$ into a semiring, called the *quotient semiring*. The natural map $\pi : R \rightarrow R/\sim$ is an epimorphism of semirings, and its induced equivalence relation \sim_π equals the original congruence \sim .

- (3) The first isomorphism theorem for semirings can be stated as follows. Every homomorphism $f : R \rightarrow S$ of semirings R and S induces an isomorphism

$$\hat{f} : R/\sim_f \rightarrow \text{im } f, \quad [x] \mapsto f(x)$$

of the quotient semiring R/\sim_f of R onto the subsemiring $\text{im } f$ of S .

- (4) If R is a ring, there is a natural bijection between the semiring congruences on R and the ring-ideals of R , where a congruence \sim is mapped to the ideal being the 0-class $[0]$.

The notion of an ideal in a ring can be generalized to semirings.

Definition 3.1.10. Let R be a semiring. A nonempty subset $A \subseteq R$ is called

$$\begin{aligned} \mathbf{ideal} & \quad \text{if} \quad A + A \subseteq A, \text{ and } RA \subseteq A, AR \subseteq A; \\ \mathbf{bi-ideal} & \quad \text{if} \quad A \text{ is an ideal, and } R + A \subseteq A; \\ \mathbf{k-ideal} & \quad \text{if} \quad A \text{ is an ideal, and } A + A^c \subseteq A^c. \end{aligned}$$

Here, A^c denotes the complement $R \setminus A$. The condition $A + A^c \subseteq A^c$ means that for all $x \in R$ and $a \in A$ with $a + x \in A$ we have $x \in A$.

An ideal A of R is called *proper* if $A \neq R$.

We warn that if R is a ring, a semiring-ideal A as in the definition above is not necessarily a ring-ideal, because A is only a submonoid rather than a subgroup of $(R, +)$. However, if the ring R is finite or has a one, every semiring-ideal is also a ring-ideal. In general rings, the ring-ideals are the same as semiring-k-ideals.

Lemma 3.1.11. *Let R be a semiring. For any ideal A , there is a congruence relation on R defined by*

$$x \sim y \Leftrightarrow \exists a, b \in A : x + a = y + b.$$

If A is a k -ideal and o is a neutral element of $(R, +)$, then the \sim -class $[o]$ equals A .

Proof. Clearly, \sim is reflexive and symmetric. Now if we have $x, y, z \in R$ with $x \sim y$ and $y \sim z$, there exists $a, b, c, d \in A$ such that $x + a = y + b$ and $y + c = z + d$. It follows that

$$x + a + c = y + b + c = z + b + d \quad \text{and} \quad a + c, b + d \in A,$$

hence $x \sim z$, and so \sim is also transitive.

Furthermore, for every $u \in R$, we have $u + x + a = u + y + b$ and hence $u + x \sim u + y$. Also, we have

$$u x + u a = u y + u b \quad \text{and} \quad u a, u b \in A,$$

so that $u x \sim u y$, and similarly we have $x u \sim y u$. It follows that \sim is a congruence relation.

Now let o be a neutral element of $(R, +)$, and let $x \in R$. Then $x \sim o$ if and only if there exist $a, b \in A$ such that $x + a = b$. If A is a k -ideal this is equivalent to $x \in A$. \square

3.1.2 Semimodules over semirings

Let R be a semiring with zero.

Definition 3.1.12. A *(left) semimodule* M over R is a commutative monoid $(M, +)$ with neutral element $0 \in M$, together with an R -multiplication

$$R \times M \rightarrow M, \quad (r, x) \mapsto r \cdot x = r x,$$

such that, for all $r, s \in R$ and $x, y \in M$, we have

$$\begin{aligned} r(sx) &= (rs)x, & 0x &= 0, & r0 &= 0, \\ (r+s)x &= rx + sx, & r(x+y) &= rx + ry. \end{aligned}$$

Remark 3.1.13. If $(M, +)$ is a commutative monoid, any representation i.e. semiring homomorphism

$$T : R \rightarrow \text{End}(M), \quad r \mapsto T_r$$

turns M into a semimodule by defining $rx := T_r(x)$, for $x \in R$ and $x \in M$.

On the other hand, let M be any semimodule over R . For $r \in R$, the map $x \mapsto rx$ defines an endomorphism T_r of M , and the map $T : R \rightarrow \text{End}(M)$, $r \mapsto T_r$ is a representation.

Definition 3.1.14. Let M be a semimodule over R .

- A *subsemimodule* $N \subseteq M$ is a submonoid of $(M, +)$ with $RN \subseteq N$.
- An equivalence relation \sim on M is called (*semimodule*) *congruence* if

$$x \sim y \quad \text{implies} \quad a + x \sim a + y, \quad r x \sim r y,$$

for all $x, y, a \in M$ and $r \in R$.

Remark 3.1.15. Note that any subsemimodule $N \subseteq M$ itself is a semimodule over R . Also, given a congruence \sim on M , we can define an addition and an R -multiplication on its set of equivalence classes $M/\sim = \{[x] \mid x \in M\}$ by

$$[x] + [y] := [x + y] \quad \text{and} \quad r[x] := [r x]$$

turning M/\sim into a semimodule over R , called the *quotient semimodule*.

As in the case of semirings, semimodule congruences are related to semimodule homomorphisms. We will discuss this connection in more detail and give notions of irreducibility for semimodules in Section 3.2.3.

3.1.3 Simple semirings

There are multiple notions of simplicity for semirings. For example one might consider semirings which have only the trivial ideals. There was a development of an “ideal-based” structure-theory, including concepts like semiring Jacobson radical and irreducible semimodules, but the main results applied only to rather special classes of semirings (see e. g. [Bou51, BZ57, Iiz59]). Moreover, these ideal-simple semirings lack an important property one wishes to attribute to “simple” objects S : namely that every nontrivial homomorphism from S should be injective, so that smaller (and thus “simpler”) homomorphic images do not exist. This property is captured by the following definition.

Definition 3.1.16. A semiring R is called (*congruence-*)*simple* if its only congruences are the trivial ones, namely $\sim = \text{id}_R$ and $\sim = R \times R$.

Remark 3.1.17. A semiring R is simple if and only if any nonconstant homomorphism $f : R \rightarrow S$ into a semiring S is injective, see Remark 3.1.9. Hence, a ring is simple if and only if it is simple in the sense that there are only trivial ideals.

By this remark, finite simple semirings have indeed no smaller homomorphic images. It is exactly this property that makes them interesting for cryptographic purposes.

Simple semirings restrict the number of bi-ideals and k -ideals:

Proposition 3.1.18. *Let R be a simple semiring.*

- (a) *Any proper bi-ideal in R has exactly one element.*
- (b) *Let o be a neutral element in $(R, +)$. Then any proper k -ideal has exactly one element, namely o .*

Proof. If A is a bi-ideal, it is easy to see that $\sim = \text{id}_R \cup (A \times A)$, i.e.

$$x \sim y \quad :\Leftrightarrow \quad x = y \quad \text{or} \quad x, y \in A,$$

defines a congruence. So if R is simple and if A is proper we must have $\sim = \text{id}_R$ and hence $|A| = 1$.

Now let R contain an additive neutral o and let A be a k -ideal. By Lemma 3.1.11 there exists a congruence \sim on R such that its class $[o]$ equals A . If A is proper we must have $\sim = \text{id}_R$ and therefore $A = \{o\}$. \square

Nevertheless, we note that a simple semiring may have proper non-singleton ideals: the semiring of Example 3.1.3 is simple, yet $\{o, b\}$ is an ideal. Conversely, there exist semirings with no proper ideals, but having many congruences, as the following example shows:

Example 3.1.19. Let (R, \leq) be a totally ordered set and define operations $+$ and \cdot on R by

$$x + y = \max(x, y), \quad x \cdot y = y.$$

It can be shown that $(R, +, \cdot)$ is a semiring that has no proper ideals. On the other hand, every equivalence relation \sim which respects the order, i.e. $x \leq y \leq z$ and $x \sim z$ implies $x \sim y \sim z$, is a congruence relation.

3.2 Classification of finite simple semirings with zero

The study of (congruence-)simple semirings started around two decades ago (see e. g. [MF88]). But it was not until 2001, when El Bashir et al. achieved a classification of (multiplicatively) commutative congruence-simple semirings [EHJK01]. Later, Monico progressed on the classification of finite congruence-simple semirings [Mon04]; his main result states that congruence-simple semirings of size > 2 are either rings, have trivial addition ($|R + R| = 1$) or have idempotent addition. At that time, very few examples of congruence-simple semirings with zero of the latter case were known, namely the square matrices over either the Boolean semiring or over a 6-element semiring found by computer search, hence this case was open as the main task of further research.

In this section we state and prove a full classification of finite congruence-simple semirings with zero, as it was published in [Zum08]. For this, we now assume that every semiring has a zero, and that every semiring homomorphism preserves the zero element.

3.2.1 Statement of the main theorem

Example 3.2.1. Let $(M, +)$ be a commutative monoid. We call a map $f : M \rightarrow M$ an endomorphism if it preserves the monoid operation and the neutral element. On the set $\text{End}(M)$ of all endomorphisms of M we get operations $+$ and \circ by defining $f + g$ as pointwise addition and $f \circ g$ as composition of maps, for $f, g \in \text{End}(M)$.

It is straight-forward to verify that $(\text{End}(M), +, \circ)$ is a semiring with a one, which will be called *endomorphism semiring*.

The classification result uses subsemirings of some endomorphism semirings, which are rich or lie dense in the sense that they contain at least certain elementary endomorphisms.

Definition 3.2.2. Let M be an idempotent commutative monoid. A subsemiring $S \subseteq \text{End}(M)$ is called **dense** if it contains for all $a, b \in M$ the endomorphism $e_{a,b} \in \text{End}(M)$, defined by

$$e_{a,b}(x) := \begin{cases} 0 & \text{if } x + a = a \\ b & \text{otherwise} \end{cases} \quad (x \in M).$$

Now we can state the main result.

Theorem 3.2.3. *Let R be a finite semiring with zero which is not a ring. Then the following are equivalent:*

- (1) R is congruence-simple.
- (2) $|R| \leq 2$ or R is isomorphic to a dense subsemiring $S \subseteq \text{End}(M)$, where $(M, +)$ is a finite idempotent commutative monoid.

We point out Proposition 3.3.2 below which implies that if two monoids M_1 and M_2 are nonisomorphic then any dense subsemirings $S_1 \subseteq \text{End}(M_1)$ and $S_2 \subseteq \text{End}(M_2)$ are nonisomorphic.

Note that the classification of finite simple rings is a classical subject in algebra. By the Wedderburn–Artin theorem (see [Her68]), a finite ring R with nontrivial multiplication is simple if and only if R is isomorphic to the endomorphism ring $\text{Mat}_{n \times n}(\mathbb{F})$ of a finite-dimensional vector space \mathbb{F}^n over a finite field \mathbb{F} .

Remark 3.2.4. There are two proper semirings of order 2, namely the semirings $R_{2,a}$, $R_{2,b}$ given by

$$R_{2,a} : \begin{array}{c|cc} + & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 1 \end{array} \quad \begin{array}{c|cc} \cdot & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 0 \end{array} \quad R_{2,b} : \begin{array}{c|cc} + & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 1 \end{array} \quad \begin{array}{c|cc} \cdot & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array} .$$

$R_{2,b}$ is the Boolean semifield of Example 3.1.6, and can also be seen as the endomorphism semiring $\text{End}(L_2)$ for $(L_2, +) = (\{0, 1\}, \max)$. Trivially, $R_{2,a}$ and $R_{2,b}$ are simple.

The smallest simple semiring with zero of order > 2 has already 6 elements. It was probably first found by Monico in 2002 with the help of a computer search program:

Example 3.2.5. Let R_6 be the set $\{0, 1, a, b, c, d\}$ with the following operations:

$+$	0	a	b	c	1	d	\cdot	0	a	b	c	1	d
0	0	a	b	c	1	d	0	0	0	0	0	0	0
a	a	a	b	c	1	d	a	0	0	0	a	a	b
b	b	b	b	1	1	d	b	0	a	b	a	b	b
c	c	c	1	c	1	d	c	0	0	0	c	c	d
1	1	1	1	1	1	d	1	0	a	b	c	1	d
d	d	d	d	d	d	d	d	0	c	d	c	d	d

The semiring $(R_6, +, \cdot)$ is simple and can be identified as the endomorphism semiring $\text{End}(M)$ of the commutative monoid $(M, +) = (\{1, 2, 3\}, \max)$.

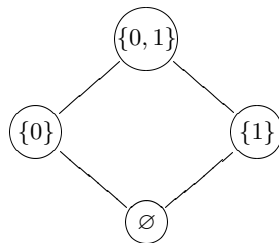
The proof of the direction (2) \Rightarrow (1) of the main result is given in Section 3.2.2, and the direction (1) \Rightarrow (2) will be proved in Section 3.2.3 with the help of irreducible semimodules.

3.2.2 Endomorphism semirings

In this subsection we prove the direction (2) \Rightarrow (1) of Theorem 3.2.3. We begin with a remark on idempotent commutative monoids and (semi-)lattices (see e.g. [Bir67, Sections I.5 and II.2]).

A *lattice* is an ordered set (L, \leq) in which every pair of elements has both a supremum (or join) and an infimum (or meet) in L . Finite lattices can be depicted by *Hasse diagrams*, which show only the covering pairs (y covers x if and only if $x < y$ and there is no z with $x < z < y$).

Example 3.2.6. Let $L = \mathcal{P}(\{0, 1\}) = \{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$, ordered by inclusion. The Hasse diagram of the corresponding lattice is depicted below.



Remark 3.2.7. Let (L, \leq) be a lattice. Its supremum operation converts the lattice into a commutative idempotent semigroup, which is in the finite case even a monoid.

Conversely, let $(M, +)$ be an idempotent commutative monoid. By defining

$$x \leq y \quad :\Leftrightarrow \quad x + y = y$$

we get a partial order relation \leq on M , where $0 \leq x$ for any $x \in M$. Also, for all $x, y \in M$ there exists a supremum $x \vee y = x + y$, so that (M, \vee) is a join-semilattice. If in addition M is finite, for all $x, y \in M$ there exists an infimum $x \wedge y = \sum_{z \leq x, z \leq y} z$, so that (M, \vee, \wedge) is even a lattice.

Though finite idempotent commutative monoids and finite lattices are basically the same thing, we note that their homomorphisms slightly differ from each other. If $(M, +)$ is a finite idempotent commutative monoid, viewed as a lattice, the elements $f \in \text{End}(M)$ are maps $f : M \rightarrow M$ satisfying $f(0) = 0$ and $f(x \vee y) = f(x) \vee f(y)$ for all $x, y \in M$. In particular, f is order-preserving. But $f(x \wedge y) = f(x) \wedge f(y)$ is not generally true, i.e. f may not be a lattice endomorphism.

Now we state a lemma on the maps $e_{a,b}$ of Definition 3.2.2. Note that by Remark 3.2.7 we have

$$e_{a,b}(x) = \begin{cases} 0 & \text{if } x \leq a \\ b & \text{otherwise.} \end{cases}$$

Lemma 3.2.8. *For $a, b \in M$, we have $e_{a,b} \in \text{End}(M)$. Also, for $f \in \text{End}(M)$ and $a, b, c, d \in M$, we have $f \circ e_{a,b} = e_{a,f(b)}$ and*

$$e_{c,d} \circ f \circ e_{a,b} = \begin{cases} 0 & \text{if } f(b) \leq c, \\ e_{a,d} & \text{otherwise.} \end{cases}$$

If $(M, +)$ has an absorbing element $\infty \in M$, i.e. $x + \infty = \infty$ for all $x \in M$, then $e_{0,\infty}$ is absorbing for $(\text{End}(M), +)$.

Proof. Note that for all $x, y \in M$, we have $x \vee y \leq a$ if and only if $x \leq a$ and $y \leq a$. It follows that $e_{a,b}(x \vee y) = 0$ if and only if $e_{a,b}(x) = 0$ and $e_{a,b}(y) = 0$, that is, if and only if $e_{a,b}(x) \vee e_{a,b}(y) = 0$. Thus $e_{a,b} \in \text{End}(M)$.

Now if $f \in \text{End}(M)$ and $a, b \in M$ one easily verifies $f \circ e_{a,b} = e_{a,f(b)}$. Applying this formula twice yields

$$e_{c,d} \circ f \circ e_{a,b} = e_{c,d} \circ e_{a,f(b)} = e_{a,e_{c,d}(f(b))} = \begin{cases} 0 & \text{if } f(b) \leq c, \\ e_{a,d} & \text{otherwise.} \end{cases}$$

Finally, for any $h \in \text{End}(M)$ and $x \in M \setminus \{0\}$ we have $(h + e_{0,\infty})(x) = h(x) + \infty = \infty$, so that $h + e_{0,\infty} = e_{0,\infty}$. \square

Proposition 3.2.9. *Let $(M, +)$ be an idempotent commutative monoid with an absorbing element. Then any dense subsemiring $R \subseteq \text{End}(M)$ is simple. In particular, $\text{End}(M)$ itself is simple.*

Note that any *finite* idempotent commutative monoid M has an absorbing element, namely $\infty := \sum_{x \in M} x$.

Proof. Let $\sim \subseteq R \times R$ be a semiring congruence relation. Suppose that $\sim \neq \text{id}_R$, so that there exists $f, g \in R$ with $f \neq g$, but $f \sim g$. There is $b \in M$ with $f(b) \neq g(b)$, and without loss of generality, we may assume $f(b) \not\leq c := g(b)$.

For all $a, d \in M$ we have $e_{a,b} \in R$ and $e_{c,d} \in R$. Hence, since \sim is a congruence, we have

$$e_{c,d} \circ f \circ e_{a,b} \sim e_{c,d} \circ g \circ e_{a,b},$$

so that $e_{a,d} \sim 0$, by Lemma 3.2.8.

In particular $e_{0,\infty} \sim 0$, where $\infty \in M$ is the absorbing element. It follows that

$$e_{0,\infty} = h + e_{0,\infty} \sim h + 0 = h$$

for any $h \in R$, since \sim is a congruence. Therefore $\sim = R \times R$, so that R has no nontrivial congruence relations. \square

3.2.3 Simple semirings and irreducible semimodules

In this section we prove that any proper finite simple semiring is of the form described in Theorem 3.2.3. We start with a result established and proved by Monico for finite semirings, not assuming a zero element [Mon04, Theorem 4.1]. We give a simpler proof of this result, assuming a zero element, but without assuming finiteness of the semiring.

Proposition 3.2.10. *Let R be a simple semiring which is not a ring. Then the addition $(R, +)$ is idempotent.*

Proof. For $x \in R$ and $n \in \mathbb{N}_0 := \{0, 1, 2, 3, \dots\}$ let us write $nx := x + \dots + x$, summing x n -times. Also let $R + x := \{y + x \mid y \in R\}$. Now, for $x, y \in R$ define

$$x \sim y \quad :\Leftrightarrow \quad \exists m, n \in \mathbb{N}_0 : mx \in R + y, ny \in R + x.$$

Then it is easily verified that \sim is a congruence relation.

By congruence-simplicity it follows that $\sim = \text{id}_R$ or $\sim = R \times R$. In the first case, since $x \sim x + x$, we deduce that $(R, +)$ is idempotent. In the second case, for all $x \in R$, we have $x \sim 0$, so that $0 \in R + x$. This shows that $(R, +)$ is a group and thus R is a ring. \square

Remark 3.2.11. A simple semiring R with idempotent addition and trivial multiplication $RR = \{0\}$ has order ≤ 2 . Indeed, since $(R, +)$ is idempotent, $x + y = 0$ implies $x = y = 0$ for $x, y \in R$, so the equivalence relation \sim on R with classes $\{0\}$ and $R \setminus \{0\}$ is a congruence. Thus $\sim = \text{id}_R$ and hence $|R| \leq 2$.

Irreducible semimodules

If M is a semimodule over R , let us call the subsemimodules $\{0\}$ and M and also the quotient semimodules $M/\text{id}_M \cong M$ and $M/(M \times M) \cong \{0\}$ the trivial ones.

Definition 3.2.12. A semimodule M over R satisfying $RM \neq \{0\}$ is called

- **sub-irreducible** if it has only trivial subsemimodules,
- **quotient-irreducible** if it has only trivial quotient semimodules,
- **irreducible** if it is both sub-irreducible and quotient-irreducible.

Some authors refer to sub-irreducible and quotient-irreducible semimodules as minimal and simple semimodules, respectively.

By a semimodule homomorphism we mean a map $f : M \rightarrow N$ between semimodules over R which preserves the semimodule operations as well as the zero element. In this case, $f(M)$ is a subsemimodule of N , and the relation $x \sim_f y$ if and only if $f(x) = f(y)$, for $x, y \in M$, is a congruence on M . On the other hand, for any subsemimodule $N_0 \subseteq N$ and any quotient semimodule M/\sim_f there are natural homomorphisms $i : N_0 \rightarrow N$ and $p : M \rightarrow M/\sim_f$. This establishes the following

Remark 3.2.13. Let M be a semimodule over R such that $RM \neq \{0\}$. Then

- M is sub-irreducible if and only if any nonzero homomorphism $f : N \rightarrow M$ from a semimodule N is surjective,
- M is quotient-irreducible if and only if any nonzero homomorphism $f : M \rightarrow N$ into a semimodule N is injective.

Remark 3.2.14. To illustrate the use of irreducible semimodules we give a version of Schur's Lemma (see [Her68]): Let M be an irreducible semimodule over R with representation $T : R \rightarrow \text{End}(M)$, $r \mapsto T_r$. Then the commuting semiring

$$C(M) := \{f \in \text{End}(M) \mid f \circ T_r = T_r \circ f \text{ for all } r \in R\}$$

is a semifield, i.e. any nonzero element is invertible. Indeed, if $f \in C(M) \setminus \{0\}$, then $f : M \rightarrow M$ is a nonzero semimodule homomorphism, which by Remark 3.2.13 must be injective and surjective. It then easily follows that the inverse f^{-1} lies in $C(M)$.

In particular, if $(M, +)$ is finite and idempotent, then $C(M)$ is a finite proper semifield. By Proposition 3.1.5 it follows that $C(M)$ has order ≤ 2 , so that $C(M) = \{0, \text{id}_M\}$ is trivial. If the representation $R \rightarrow \text{End}(M)$ is faithful i.e. injective (this holds for example if R is simple and $RM \neq \{0\}$), it follows that R has trivial center, since

$$\{x \in R \mid xr = rx \text{ for all } r \in R\} = T^{-1}(C(M)) = \{0, 1\} \cap R.$$

Existence of irreducible semimodules

Proposition 3.2.15. *Any finite simple semiring R with $RR \neq \{0\}$ admits a finite irreducible semimodule.*

To prove this result we begin with two lemmas that guarantee the property $RM \neq \{0\}$ for certain semimodules M over R . By a nontotal semimodule congruence on M is meant a congruence $\sim \neq M \times M$, so that $M/\sim \neq \{0\}$.

Lemma 3.2.16. *Let R be a simple semiring with $RR \neq \{0\}$, considered as a semimodule over itself, and let \sim be a nontotal semimodule congruence on R . Then, for the quotient semimodule $M := R/\sim$ we have $RM \neq \{0\}$.*

Proof. Since \sim is a semimodule congruence, $r \sim s$ implies $x + r \sim x + s$ and $xr \sim xs$ for any $r, s, x \in R$. Now suppose $RM = \{0\}$. Then for any $r, x \in R$ we have $[rx] = r[x] = 0$, so that $rx \sim 0$. Hence $r \sim s$ implies also $rx \sim sx$, for any $r, s, x \in R$, so that \sim is even a semiring congruence. Since \sim is nontotal, we must have $\sim = \text{id}_R$ by congruence-simplicity. Hence $M = R$ and $RR = \{0\}$, which contradicts our assumption. \square

Lemma 3.2.17. *Let M be a semimodule over R such that $RM \neq \{0\}$.*

1. *If M is sub-irreducible, then $RP \neq \{0\}$ for all its nonzero quotient semimodules $P = M/\sim$.*
2. *If M is quotient-irreducible, then $RN \neq \{0\}$ for all its nonzero sub-semimodules $N \subseteq M$.*

Proof. (1) Let M have only trivial subsemimodules. Since $RM \subseteq M$ is a subsemimodule, we must have $RM = M$. Now let $P = M/\sim$ be a quotient subsemimodule with $RP = \{0\}$. Then we have $M = RM \subseteq [0]_\sim$, and therefore $M/\sim = \{0\}$.

(2) Let $A := \{x \in M \mid Rx = \{0\}\} \subseteq M$ be the annihilator of R in M . Then it is easy to check that A is a semimodule of M with the additional property that $x \in A$ and $x + y \in A$ implies $y \in A$. Also it is straightforward to check that by defining

$$x \sim y \quad :\Leftrightarrow \quad \exists a, b \in A : x + a = y + b$$

for $x, y \in M$ a congruence \sim on M is obtained such that its zero-class $\{x \in M \mid x \sim 0\}$ equals A . Finally note that $A \neq M$ by assumption.

Now if M has only trivial quotient semimodules, the relation \sim above must equal id_M , and hence $A = \{0\}$. It follows that any subsemimodule $N \subseteq M$ with $RN = 0$ must be zero. \square

Proof of Proposition 3.2.15. We recursively define a sequence of finite semimodules M_0, M_1, \dots, M_n over R of decreasing sizes such that

- for all $i = 0, \dots, n$ we have $RM_i \neq \{0\}$,
- for all $i = 1, \dots, n$ we have M_i is sub-irreducible or quotient-irreducible,
- M_n is irreducible.

We start with $M_0 := R$, so that $RM_0 = RR \neq \{0\}$.

Now let \sim be a maximal nontotal semimodule congruence on R (probably $\sim = \text{id}_R$) and let $M_1 := R/\sim$. Since \sim is nontotal we have $RM_1 \neq \{0\}$ by Lemma 3.2.16. By maximality of \sim it follows that M_1 is quotient-irreducible.

Suppose that M_i has been defined for some $i \geq 1$, so that $RM_i \neq \{0\}$ and M_i is sub-irreducible or quotient-irreducible. If M_i is even irreducible we set $n = i$ and stop.

Otherwise suppose that M_i is quotient-irreducible but has nontrivial subsemimodules. Take a minimal nonzero semimodule $M_{i+1} \subseteq M_i$. Then $RM_{i+1} \neq \{0\}$ by Lemma 3.2.17, (2), and furthermore M_{i+1} is sub-irreducible. Now consider the case where M_i is sub-irreducible but has nontrivial congruences. By taking a maximal nontotal congruence \sim and letting $M_{i+1} := M_i/\sim$, we have $RM_{i+1} \neq \{0\}$ by Lemma 3.2.17, (1), and furthermore M_{i+1} is quotient-irreducible.

The sequence has been constructed. Since R is finite and the cardinalities of M_1, M_2, \dots are strictly decreasing the sequence must terminate by an irreducible semimodule M_n over R . \square

A density result

Let R be a simple semiring and M be a semimodule over R with $RM \neq \{0\}$. Then the representation $R \rightarrow \text{End}(M)$ is nonzero and hence must be injective, so that R can be seen as a subsemiring of $\text{End}(M)$. If M is irreducible the question of the “density” of R in $\text{End}(M)$ arises. We have already seen in Remark 3.2.14 that the commutant semiring of R in $\text{End}(M)$ is trivial if $(M, +)$ is idempotent. Now we show another density result:

Proposition 3.2.18. *Let R be a finite simple semiring with idempotent addition and let M be a finite irreducible semimodule over R . Then $(M, +)$ is idempotent, and for all $a, b \in M$ there exists $r \in R$ such that*

$$rx = \begin{cases} 0 & \text{if } x + a = a \\ b & \text{otherwise} \end{cases} \quad (x \in M).$$

Thus R , seen as a subsemiring of $\text{End}(M)$, is dense (see Definition 3.2.2).

Proof. First note that $(M, +)$ is idempotent: By irreducibility, the subsemimodule RM of M is nonzero, hence $RM = M$. So, any $x \in M$ can be written as $x = ry$ with $r \in R$ and $y \in M$. It follows

$$x + x = ry + ry = (r + r)y = ry = x,$$

since $(R, +)$ is idempotent, so that $(M, +)$ is idempotent. Recall from Remark 3.2.7 that now on M there is an order relation \leq defined by $x \leq y$ if and only if $x + y = y$, for $x, y \in M$. Recall also that, since M is finite, there exists an absorbing element $\infty = \sum_{x \in M} x$ of $(M, +)$.

For $x \in M$ define

$$I_x := \{r \in R \mid rx = 0\},$$

which is a subsemimodule of R . We have $I_{x+y} = I_x \cap I_y$ for $x, y \in M$, since $rx + ry = 0$ implies $rx = ry = 0$ for $r \in R$, because $(M, +)$ is idempotent. Now we claim that by defining

$$x \sim y \quad :\Leftrightarrow \quad I_x = I_y \quad (x, y \in M)$$

we obtain a semimodule congruence on M : Indeed, if $x \sim y$ and $z \in M$, we have $I_{z+x} = I_z \cap I_x = I_z \cap I_y = I_{z+y}$, so that $z + x \sim z + y$. Also for $r, s \in R$ we have $r(sx) = (rs)x = 0$ if and only if $(rs)y = r(sy) = 0$, so that $I_{sx} = I_{sy}$ i.e. $sx \sim sy$.

Assume that $\sim = M \times M$. Then $I_x = I_0 = R$ for all $x \in M$, so that $RM = \{0\}$, which cannot hold. Since M is quotient-irreducible it follows that $\sim = \text{id}_M$. We conclude that $x \leq y$ is equivalent to $I_y \subseteq I_x$, for $x, y \in M$, since $x + y = y$ if and only if $I_x \cap I_y = I_{x+y} = I_y$.

Now let $a \in M$ be fixed. If $a = \infty$, then the assertion trivially holds with $r = 0$. So assume $a \neq \infty$. For any $x \in M$ with $x \not\leq a$ we have shown before that $I_a \not\subseteq I_x$, so the semimodule homomorphism $I_a \rightarrow M$, $r \mapsto rx$ is nonzero. Since M is sub-irreducible, it must be surjective, so in particular there exists $r_x \in I_a$ such that $r_x x = \infty$. Letting $s := \sum_{x \not\leq a} r_x \in I_a \subseteq R$, for $x \in M$ we have

$$sx = \begin{cases} 0 & \text{if } x \leq a, \text{ since then } sx = sx + sa = sa = 0, \\ \infty & \text{if } x \not\leq a, \text{ since then } sx \geq r_x x = \infty, \end{cases}$$

so we have shown the assertion for $b = \infty$.

Consider now the subsemimodule

$$N := \{r\infty \mid r \in R\}$$

of M . We have $\infty = s\infty \in N$, so that $N \neq \{0\}$. By sub-irreducibility of M it follows $N = M$, so for any $b \in M$ there exists $r \in R$ with $r\infty = b$. Then for $x \in M$ we have $(rs)x = 0$ if $x \leq a$, and $(rs)x = b$ otherwise, which completes the proof. \square

Now we complete the proof of the Theorem 3.2.3 by showing the direction (1) \Rightarrow (2). Let R be a proper finite simple semiring and suppose $|R| > 2$. Then $(R, +)$ is idempotent by Proposition 3.2.10 and $RR \neq \{0\}$ by Remark 3.2.11. Afterwards, Proposition 3.2.15 guarantees the existence of a finite irreducible semimodule M over R , so that R is isomorphic to a subsemiring S of $\text{End}(M)$. Finally, by Proposition 3.2.18 we have that S is a dense subsemiring of $\text{End}(M)$.

3.3 The family of finite simple semirings

Definition 3.3.1. Let M be an idempotent commutative monoid. We define $\mathcal{SR}(M)$ to be the collection of all dense subsemirings $R \subseteq \text{End}(M)$.

In this section we take a closer look at the families $\mathcal{SR}(M)$. By the main theorem, Theorem 3.2.3, these families form the collection of all finite simple semirings. First we consider the question of isomorphism and anti-isomorphism of these semirings. Then we give a criterion to decide whether the family $\mathcal{SR}(M)$ is trivial. Finally we list the dense endomorphism subsemirings of smallest order.

Throughout this section, let M, M_1 and M_2 be idempotent commutative monoids having an absorbing element.

3.3.1 Isomorphism

Proposition 3.3.2. *Let $R_1 \in \mathcal{SR}(M_1)$ and $R_2 \in \mathcal{SR}(M_2)$ be isomorphic semirings. Then also the monoids M_1 and M_2 are isomorphic.*

We first formulate and prove a lemma. Recall from Lemma 3.2.8 that if $\infty \in M$ is the absorbing element, then $e_{0,\infty}$ is an absorbing element in $(R, +)$ for any semiring $R \in \mathcal{SR}(M)$.

Lemma 3.3.3. *Let $R \in \mathcal{SR}(M)$ and let $z \in R$ be the absorbing element in $(R, +)$. Then the map*

$$\theta : M \rightarrow Rz, \quad b \mapsto e_{0,b}$$

defines an isomorphism between $(M, +)$ and the submonoid Rz of $(R, +)$.

Proof. Note that $f \circ e_{0,\infty} = e_{0,f(\infty)}$ for all $f \in R$, so in particular $e_{0,b} \circ e_{0,\infty} = e_{0,b}$ for all $b \in M$. This shows

$$Rz = Re_{0,\infty} = \{e_{0,b} \mid b \in M\},$$

so θ is well-defined and surjective. It is clear that θ is injective and a homomorphism. \square

Proof of Proposition 3.3.2. Suppose there is a semiring isomorphism $\phi : R_1 \rightarrow R_2$. For $i = 1, 2$, let $z_i \in R_i$ be the absorbing element in $(R_i, +)$. We then have $\phi(z_1) = z_2$ and thus $\phi(R_1 z_1) = R_2 z_2$. The restriction $\phi' = \phi|_{R_1 z_1} : R_1 z_1 \rightarrow R_2 z_2$ of ϕ is therefore an isomorphism between the submonoids $R_1 z_1$ and $R_2 z_2$ of $(R_1, +)$ and $(R_2, +)$, respectively. Now for $i = 1, 2$, let $\theta_i : M_i \rightarrow R_i z_i$ be the isomorphism defined in Lemma 3.3.3. Then we can construct an isomorphism

$$\theta_2^{-1} \circ \phi' \circ \theta_1 : M_1 \rightarrow M_2$$

between the monoids $(M_1, +)$ and $(M_2, +)$. \square

Next we identify anti-isomorphic pairs of simple semirings.

Remark 3.3.4. Let M be finite with corresponding lattice (M, \vee, \wedge) , so that $(M, +) = (M, \vee)$. Then also (M, \wedge) is a finite idempotent commutative monoid, which we denote by \tilde{M} . Its corresponding lattice is the *dual lattice* of M , obtained by reversing the ordered set (M, \leq) .

Let $(L_2, \vee) = (\{0, 1\}, \max)$ and let $M^* = \text{Hom}(M, L_2)$ be the set of all monoid homomorphisms $M \rightarrow L_2$. Defining addition pointwise, M^* becomes a finite idempotent commutative monoid.

Lemma 3.3.5. *The monoid M^* is isomorphic to \tilde{M} . In fact, the map*

$$M \rightarrow M^*, \quad a \mapsto e_a, \quad \text{where } e_a(x) = \begin{cases} 0 & \text{if } x \leq a, \\ 1 & \text{otherwise,} \end{cases}$$

is a bijection such that $e_{a \wedge b} = e_a \vee e_b$ for all $a, b \in M$.

Proof. This is rephrasing the well-known result in lattice theory: Any finite lattice is isomorphic to its lattice of ideals (see [Bir67, Section II.3]). \square

Proposition 3.3.6. *Let M be finite. The semirings $\text{End}(M)$ and $\text{End}(\tilde{M})$ are anti-isomorphic.*

Proof. By Lemma 3.3.5 we may assume $\tilde{M} = M^*$. Consider the map

$$\text{End}(M) \rightarrow \text{End}(M^*), \quad f \mapsto f^*, \quad \text{where } f^*(\phi) := \phi \circ f \text{ for } \phi \in M^*.$$

It is easy to see that this map is well-defined and that the following algebraic properties hold for $f, g \in \text{End}(M)$:

$$(f + g)^* = f^* + g^*, \quad 0^* = 0, \quad (f \circ g)^* = g^* \circ f^*.$$

To prove injectivity, suppose we have $f, g \in \text{End}(M)$ with $f^* = g^*$. With e_a as defined in Lemma 3.3.5 it follows $e_a(f(x)) = e_a(g(x))$ for all $a, x \in M$, so that $f(x) \leq a$ if and only if $g(x) \leq a$. For all $x \in M$ it follows $f(x) = g(x)$, hence $f = g$.

From injectivity it follows in particular $|\text{End}(M)| \leq |\text{End}(\tilde{M})|$. We can apply this result to \tilde{M} to yield $|\text{End}(\tilde{M})| \leq |\text{End}(M)|$. Thus $|\text{End}(M)| = |\text{End}(\tilde{M})|$ and the map is also surjective. \square

Corollary 3.3.7. *Let M be finite and suppose M as a lattice is isomorphic to its dual lattice. Then the semiring $\text{End}(M)$ is anti-isomorphic to itself.*

Corollary 3.3.8. *Let M_1 and M_2 be finite and let $R_1 \in \mathcal{SR}(M_1)$ and $R_2 \in \mathcal{SR}(M_2)$ be anti-isomorphic semirings. Then the monoids M_1 and \tilde{M}_2 are isomorphic.*

Proof. By Proposition 3.3.6, $\text{End}(M_2)$ is anti-isomorphic to $\text{End}(\tilde{M}_2)$, and thus R_1 is isomorphic to some $R'_2 \in \mathcal{SR}(\tilde{M}_2)$. Now the result follows from Proposition 3.3.2. \square

3.3.2 The case $|\mathcal{SR}(M)| = 1$

We now discuss under which circumstances the only dense subsemiring of $\text{End}(M)$ is $\text{End}(M)$ itself.

Proposition 3.3.9. *Let M be finite. Then we have $\mathcal{SR}(M) = \{\text{End}(M)\}$ if and only if the lattice (M, \vee, \wedge) satisfies the following condition:*

$$\forall z \in M : z = \bigvee_{a, z \not\leq a} \bigwedge_{x, x \not\leq a} x. \quad (\text{D})$$

Proof. If S is the subsemiring of $R := \text{End}(M)$ generated by the set $E := \{e_{a,b} \mid a, b \in M\}$, then we have $\mathcal{SR}(M) = \{\text{End}(M)\}$ if and only if $S = R$. Note that since E is closed under multiplication (see Lemma 3.2.8) S consists of all finite sums of elements in E . Writing $1 = \text{id}_M \in R$ we show that

$$S = R \quad \text{if and only if} \quad 1 = \sum_{(a,b) \in X} e_{a,b} \quad (*)$$

with $X := \{(a, b) \in M^2 \mid e_{a,b} \leq 1\}$.

Indeed, suppose $S = R$, so we can express in particular 1 as a sum of elements in E , say $1 = \sum_i e_{a_i, b_i}$. Surely, $e_{a_i, b_i} \leq 1$ and hence $(a_i, b_i) \in X$ for all i , so that

$$1 = \sum_i e_{a_i, b_i} \leq \sum_{(a,b) \in X} e_{a,b} \leq 1$$

and thus the right side of $(*)$ holds. On the other hand, suppose $1 = \sum_{(a,b) \in X} e_{a,b}$. Then for any $f \in R$ we have

$$f = f \circ 1 = \sum_{(a,b) \in X} f \circ e_{a,b} = \sum_{(a,b) \in X} e_{a, f(b)} \in S$$

(see Lemma 3.2.8), so that $S = R$. This proves the equivalence $(*)$.

Note next that $(a, b) \in X$ i.e. $e_{a,b} \leq 1$ if and only if $b \leq x$ for all $x \not\leq a$ which is equivalent to $b \leq \bigwedge_{x, x \not\leq a} x$. This shows that

$$\sum_{(a,b) \in X} e_{a,b} = \sum_{a \in M} e_{a, b_a} \quad \text{with} \quad b_a := \bigwedge_{x, x \not\leq a} x.$$

Now for all $z \in M$ we have

$$\sum_{(a,b) \in X} e_{a,b}(z) = \sum_{a \in M} e_{a,b_a}(z) = \bigvee_{a, z \not\leq a} b_a = \bigvee_{a, z \not\leq a} \bigwedge_{x, x \not\leq a} x,$$

which together with (*) concludes the proof. \square

Remark 3.3.10. The condition (D) given in proposition 3.3.9 is fulfilled if and only if the lattice M is distributive, or equivalently, M is isomorphic to a ring of subsets (cf. [Bir67, Section III.3]).

Indeed, assume that (M, \cup, \cap) is a ring of subsets, i.e. a sublattice of a power set lattice $(\mathcal{P}(\Omega), \cup, \cap)$. For $\omega \in \Omega$ define $A_\omega := \bigcup_{X \in M, \omega \notin X} X \in M$. Then for $X \in M$ we have $X \subseteq A_\omega$ if and only if $\omega \notin X$. It follows

$$Z \supseteq \bigcup_{A, Z \not\subseteq A} \bigcap_{X, X \not\subseteq A} X \supseteq \bigcup_{\omega, Z \not\subseteq A_\omega} \bigcap_{X, X \not\subseteq A_\omega} X = \bigcup_{\omega, \omega \in Z} \bigcap_{X, \omega \in X} X \supseteq Z$$

for all $Z \in M$, so M satisfies property (D).

On the other hand, if we have a lattice (M, \vee, \wedge) with condition (D), let $\Omega := \{b_a \mid a \in M\}$ with $b_a := \bigwedge_{x, x \not\leq a} x$. Consider the representation of M given by

$$\Phi : M \rightarrow \mathcal{P}(\Omega), \quad z \mapsto \{b_a \mid a \in M, z \not\leq a\}.$$




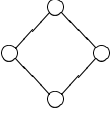
We can see directly that $z_1 \leq z_2$ implies $\Phi(z_1) \subseteq \Phi(z_2)$. On the other hand, with the help of (D) we conclude that $\Phi(z_1) \subseteq \Phi(z_2)$ implies $z_1 = \bigvee_{a, z_1 \not\leq a} b_a \leq \bigvee_{a, z_2 \not\leq a} b_a = z_2$. It follows that Φ is a lattice monomorphism, so that M is isomorphic to a sublattice of $(\mathcal{P}(\Omega), \cup, \cap)$.

3.3.3 Congruence-simple semirings of small order

Table 3.1 shows the smallest nontrivial idempotent commutative monoids M (up to isomorphism), represented by the Hasse-diagram of the corresponding lattices, together with the semirings in the collection $\mathcal{SR}(M)$. We write R_m for a semiring with m elements.

These, together with $R_{2,a}$ from Remark 3.2.4, are the smallest congruence-simple semirings which are not rings. The smallest such semiring not shown in Table 3.1 has order 98.

Note that $R_{50,a}$ and $R_{50,b}$ are anti-isomorphic to each other by Proposition 3.3.6, whereas the other semirings in Table 3.1 are self-anti-isomorphic by Corollary 3.3.7. Furthermore, all semirings in Table 3.1 except R_{42} and R_{44} have a one-element.

M	$\mathcal{SR}(M)$
	$\{R_{2,b}\}$ (the Boolean semiring)
	$\{R_6\}$
	$\{R_{20}\}$
	$\{R_{16}\}$ (the 2×2 -matrices over $R_{2,b}$)


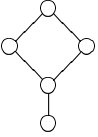
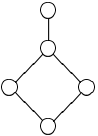
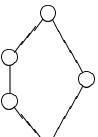
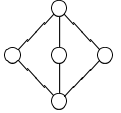
M	$\mathcal{SR}(M)$
	$\{R_{70}\}$
	$\{R_{50,a}\}$
	$\{R_{50,b}\}$
	$\{R_{43}, R_{42}\}$
	$\{R_{50,c}, R_{47}, R_{46,a}, R_{46,b}, R_{46,c}, R_{45}, R_{44}\}$ (where $R_{46,a}, R_{46,b}$ and $R_{46,c}$ are isomorphic)

Table 3.1: The smallest lattices together with the corresponding endomorphism semirings.

Chapter 4

Semigroup actions based on simple semirings

The idea to use simple semirings for constructing semigroup action based cryptosystem originates from the work of Monico, Maze and Rosenthal [Mon02, Maz03, MMR07].

Simple semirings appear to be well-suited for cryptographic purposes, because of the following reasons:

- They have enough structure for a sensible matrix multiplication. Hence they can be used as building blocks for large objects, like a family of semigroup actions.
- Simple semirings avoid a Pohlig-Hellman analogous reduction attack, since they do not admit smaller homomorphic images.
- Many linear algebra tools for fields like diagonalization are not applicable to proper semirings.¹

The classification of finite simple semirings with zero, Theorem 3.2.3, provides new methods to construct semigroup actions. There are essentially two approaches for constructing a family of semigroup actions with difficult SDL problem, based on simple semirings.

- (1) Consider a fixed simple semiring of small or moderate size, given as a “black-box”, and use it as a building block to construct larger objects. The operation tables are precomputed and stored explicitly in memory, thus providing maximal efficiency.

¹Note that a generalization from fields to simple rings does not lead to something new: By the Wedderburn–Artin theorem, any finite simple ring R with $R^2 \neq \{0\}$ is isomorphic to the matrix ring $\text{Mat}_{n \times n}(\mathbb{F}_q)$ over a finite field; hence matrix rings over R are isomorphic to matrix rings over \mathbb{F}_q , in fact $\text{Mat}_{m \times m}(R) \cong \text{Mat}_{mn \times mn}(\mathbb{F}_q)$.

- (2) Consider huge simple semirings, given as the endomorphism semirings of lattices of moderate size, and use them directly. The semiring operations are provided implicitly by storing only the lattice structure.

The sections of this chapter deal with the two approaches outlined above. The first section gathers some results from matrix theory over semirings and presents two semiring-based cryptosystems that have already been proposed. In the second section large endomorphism semirings are considered with respect to their cryptographic applicability.

In this chapter we assume that every semiring has a zero element.

4.1 Matrices over semirings

A practical method to construct large scalable objects out of smaller semirings is to use matrices. We consider matrix semirings and provide a link between endomorphism semirings and matrix semirings. Then we investigate the conditions needed for the associativity of matrix multiplication. Afterwards we consider semigroup actions based on matrices over semirings.

Definition 4.1.1. Let R be a commutative monoid. Denote by $\text{Mat}_{m \times n}(R)$ the commutative monoid of all $m \times n$ matrices with entries in R , where for $A = (a_{ij}), B = (b_{ij}) \in \text{Mat}_{m \times n}(R)$ the **matrix sum** $A + B = (a_{ij} + b_{ij})$ is defined component-wise.

Let R be a semiring. For matrices $A = (a_{ij}) \in \text{Mat}_{m \times n}(R)$ and $B = (b_{jk}) \in \text{Mat}_{n \times p}(R)$ define the **matrix product** AB to be the matrix $(c_{ik}) \in \text{Mat}_{m \times p}(R)$, where

$$c_{ik} = \sum_{j=1}^n a_{ij} b_{jk} .$$

The set of square matrices $\text{Mat}_{n \times n}(R)$, together with matrix sum and matrix product, forms a semiring. It is called the **matrix semiring**.

If the base semiring has a one, then also the matrix semiring has a one. We cite from [Maz03, Theorem 4.14] or [MMR07, Theorem 5.5]:

Proposition 4.1.2. *Let R be a semiring with one. For every semiring congruence \approx on $\text{Mat}_{n \times n}(R)$ there is a semiring congruence \sim on R such that*

$$(a_{ij}) \approx (b_{ij}) \quad \Leftrightarrow \quad \forall i, j : a_{ij} \sim b_{ij} .$$

In particular, if R is simple then also $\text{Mat}_{n \times n}(R)$ is simple.

4.1.1 Matrices describing homomorphisms

As matrices over a field K are used to describe linear maps between vector spaces over K , matrices over semirings can be used to describe semimodule homomorphisms between free semimodules. In this section we present some elementary results, most of them can be found in [Gol99, Sections 14,17].

We start our discussion with homomorphisms of commutative monoids. Let M_1, \dots, M_n be commutative monoids, and let $\prod_{i=1}^n M_i$ be the Cartesian (or direct) product of the monoids. Consider for $j = 1, \dots, n$ the natural monomorphism $\varepsilon_j : M_j \rightarrow \prod_{i=1}^n M_i$ and the natural epimorphism $\pi_j : \prod_{i=1}^n M_i \rightarrow M_j$.

Let N be another commutative monoid. There are isomorphisms of monoids

$$\begin{aligned} \text{Hom}\left(\prod_{i=1}^n M_i, N\right) &\cong \prod_{i=1}^n \text{Hom}(M_i, N) \\ f &\mapsto (f \circ \varepsilon_i)_{i=1}^n \end{aligned} \quad (4.1)$$

(saying that $\prod_{i=1}^n M_i$ together with the maps ε_i is the categorical coproduct), and

$$\begin{aligned} \text{Hom}\left(N, \prod_{i=1}^n M_i\right) &\cong \prod_{i=1}^n \text{Hom}(N, M_i) \\ f &\mapsto (\pi_i \circ f)_{i=1}^n \end{aligned} \quad (4.2)$$

(saying that $\prod_{i=1}^n M_i$ together with the maps π_i is the categorical product).

By combining (4.1) and (4.2) we obtain:

Lemma 4.1.3. *For any commutative monoids M_1, \dots, M_m and N_1, \dots, N_n we have an isomorphism of monoids*

$$\text{Hom}\left(\prod_{j=1}^n N_j, \prod_{i=1}^m M_i\right) \cong \prod_{i,j} \text{Hom}(N_j, M_i).$$

Under this isomorphism a map $f \in \text{Hom}\left(\prod_{j=1}^n N_j, \prod_{i=1}^m M_i\right)$ corresponds to the matrix $(\pi_i \circ f \circ \varepsilon_j)_{i,j} =: (f_{ij})_{i,j}$.

Conversely, a matrix $(f_{ij})_{i,j}$ with $f_{ij} \in \text{Hom}(N_j, M_i)$ corresponds to the map $f \in \text{Hom}\left(\prod_{j=1}^n N_j, \prod_{i=1}^m M_i\right)$ defined by

$$f((m_j)_{j=1}^n) := \left(\sum_{j=1}^n f_{ij}(m_j)\right)_{i=1}^m.$$

Regarding composition of maps, we have:

Lemma 4.1.4. *Let M_i, N_j, O_k for i, j, k be commutative monoids. If $f \in \text{Hom}\left(\prod_{j=1}^n N_j, \prod_{i=1}^m M_i\right)$ and $g \in \text{Hom}\left(\prod_{k=1}^o O_k, \prod_{j=1}^n N_j\right)$ we have $f \circ g \in \text{Hom}\left(\prod_{k=1}^o O_k, \prod_{i=1}^m M_i\right)$, and for the (i, k) -entry h_{ik} of the representation of $h = f \circ g$ as a matrix we have*

$$h_{ik} = \sum_{j=1}^n f_{ij} g_{jk},$$

which is the usual matrix product.

Proof. We have $h_{ik} = \pi_i \circ f \circ g \circ \varepsilon_k$. With the notation $f \circ g = f \cdot g = fg$ we compute

$$\begin{aligned} h_{ik} &= \pi_i \cdot fg \cdot \varepsilon_k = \pi_i f \cdot \text{id}_{\prod_{j=1}^n N_j} \cdot g \varepsilon_k \\ &= \pi_i f \cdot \sum_{j=1}^n \varepsilon_j \pi_j \cdot g \varepsilon_k \\ &= \sum_{j=1}^n \pi_i f \varepsilon_j \cdot \pi_j g \varepsilon_k = \sum_{j=1}^n f_{ij} g_{jk}. \quad \square \end{aligned}$$

The following result follows immediately from Lemmas 4.1.3 and 4.1.4.

Proposition 4.1.5. *Let M, N be commutative monoids. Then*

- (1) $\text{Hom}(N^n, M^m) \cong \text{Mat}_{m \times n}(\text{Hom}(N, M))$ as commutative monoids.
- (2) $\text{End}(N^n) \cong \text{Mat}_{n \times n}(\text{End}(N))$ as semirings.

The results of Lemmas 4.1.3 and 4.1.4 also hold if the category of commutative monoids is replaced by the category of left or right semimodules over a semiring, since also here the finite Cartesian product serves as the categorical coproduct and product. For example, let R be a semiring and N_R and M_R be right semimodules over R , then

$$\text{Hom}_R(N_R^n, M_R^m) \cong \text{Mat}_{m \times n}(\text{Hom}_R(N_R, M_R))$$

as commutative monoids; here Hom_R denotes the semimodule homomorphisms.

Lemma 4.1.6. *Let R be a semiring with one. Denote by R_R the semiring R , seen as a right module over itself. Then $R \cong \text{End}_R(R_R)$ as semirings.*

Proof. Consider the map

$$\begin{aligned} T : R &\rightarrow \text{End}(R, +) \\ r &\mapsto T_r : x \mapsto rx, \end{aligned}$$

which is a semiring homomorphism. If $T_r = T_s$ then $r = T_r(1) = T_s(1) = s$, hence T is injective. It remains to prove $\text{im} T = \text{End}_R(R_R)$.

For each $r \in R$ we have $T_r(xs) = r(xs) = (rx)s = T_r(x)s$, hence $T_r \in \text{End}_R(R_R)$. Conversely, for $f \in \text{End}_R(R_R)$ let $r := f(1)$, then $f(x) = f(1x) = f(1)x = rx$ for all $x \in R$, and hence $f = T_r \in \text{im} T$. \square

As a corollary we get the following interpretation of matrices as endomorphisms:

Proposition 4.1.7. *Let R be a semiring with one. Then*

- (1) $\text{Hom}(R_R^n, R_R^m) \cong \text{Mat}_{m \times n}(R)$ as commutative monoids.
- (2) $\text{End}(R_R^n) \cong \text{Mat}_{n \times n}(R)$ as semirings.

Under these isomorphisms, a matrix $(r_{ij}) \in \text{Mat}_{m \times n}(R)$ corresponds to the map $f \in \text{Hom}(R_R^n, R_R^m)$ defined by $f((m_j)_{j=1}^n) := \left(\sum_{j=1}^n r_{ij} m_j \right)_{i=1}^m$.

Remark 4.1.8. Let R be a semiring with one. A matrix $A \in \text{Mat}_{n \times n}(R)$ is called *invertible* if there exists a matrix $B \in \text{Mat}_{n \times n}(R)$ such that $AB = BA = I_n$, where I_n denotes the identity matrix.

It follows from Proposition 4.1.7, (2) that a matrix A is invertible if and only if its corresponding endomorphism is an isomorphism.

Furthermore, if R is finite then a left-invertible matrix A over R is already invertible. Indeed, if A has a left inverse, then also the corresponding endomorphism $f_A \in \text{End}_R(R_R^n)$ has a left inverse. Consequently, f_A is injective and since R is finite, f_A has to be bijective. This implies that f_A is an isomorphism and hence A is invertible. A similar argument shows that a right-invertible matrix is already invertible.

The property that $AB = I_n$ implies $BA = I_n$ is also true for commutative infinite semirings, as shown by Reutenauer and Straubing [RS84].

4.1.2 Associativity of matrix multiplication

It is natural to ask which axioms a general ring-like algebraic structure $(R, +, \cdot)$ must obey in order that matrix multiplication over R is associative. We will clarify this question and see that under very weak assumptions exactly the semiring axioms have to be satisfied, i.e. semirings are the most general structures such that matrix multiplication is associative.

Thus let $(R, +, \cdot)$ be any algebraic structure of type $(2, 2)$, i.e. R is a set with two binary operations $+$ and \cdot on R . On the set $\text{Mat}_{n \times n}(R)$ of square

matrices with entries in R , we can formally define the matrix multiplication: If $A = (a_{ij})$, $B = (b_{ij})$ are in $\text{Mat}_{n \times n}(R)$, then let

$$AB = (c_{ik}) \quad \text{with} \quad c_{ik} = \sum_{j=1}^n a_{ij} b_{jk},$$

where we agree to evaluate first the products and then the sum right-associatively.

Proposition 4.1.9. *Let $(R, +, \cdot, 0)$ be an algebraic structure of type $(2, 2, 0)$, i.e. $+$ and \cdot are binary operations on R and 0 is an element of R . Assume that the identities $0 + x = x = x + 0$ and $0x = 0 = x0$ hold and that $R \cdot R = R$.*

Let n be an integer, $n \geq 2$. If the multiplication of $n \times n$ matrices over R is associative, then $(R, +, \cdot)$ is a semiring.

Proof. It is easy to show that the map $\text{Mat}_{2 \times 2}(R) \rightarrow \text{Mat}_{n \times n}(R)$ given by

$$A \mapsto \begin{pmatrix} A & 0 & \cdots \\ 0 & 0 & \\ \vdots & & \ddots \end{pmatrix}$$

is a groupoid monomorphism. Therefore, it suffices to consider the case $n = 2$.

Assume that the associativity condition $(AB)C = A(BC)$ holds for all $A, B, C \in \text{Mat}_{2 \times 2}(R)$. Let

$$A = \begin{pmatrix} e & f \\ * & * \end{pmatrix}, \quad B = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad C = \begin{pmatrix} g & * \\ h & * \end{pmatrix}$$

with $a, b, c, d, e, f, g, h \in R$. Then the equation $(AB) \cdot C = A \cdot (BC)$ reads

$$\begin{pmatrix} ea + fc & eb + fd \\ * & * \end{pmatrix} \cdot \begin{pmatrix} g & * \\ h & * \end{pmatrix} = \begin{pmatrix} e & f \\ * & * \end{pmatrix} \cdot \begin{pmatrix} ag + bh & * \\ cg + dh & * \end{pmatrix},$$

so the $(1, 1)$ -entries give the equation

$$(ea + fc)g + (eb + fd)h = e(ag + bh) + f(CG + dh). \quad (4.3)$$

Now letting $e = h = 0$ we have $(fc)g = f(CG)$, hence (R, \cdot) is associative. If we let only $e = 0$ in (4.3), then $f(CG) + f(dh) = f(CG + dh)$, hence $fu + fv = f(u + v)$ for all $u, v \in R$, since $RR = R$. Similarly, if $h = 0$ it follows $(ea + fc)g = (ea)g + (fc)g$ and thus $(u + v)g = ug + vg$ for all $u, v \in R$.

Furthermore, letting $c = 0$ in (4.3) yields

$$eag + (ebh + fdh) = (eag + ebh) + fdh,$$

and since $RRR = R$ this implies that $(R, +)$ is associative. Finally, let $a = d = 0$ in (4.3) to see $fcg + ebh = ebh + fcg$ and thus $(R, +)$ is also commutative. \square

4.1.3 Semigroup actions based on matrices over semirings

We present two ideas to construct interesting semigroup actions using a fixed semiring. Both of them involve matrices over semirings.

The first semigroup action is a special case of a semimodule action, see Example 2.1.6, (4). It was studied by Monico [Mon02, Section 4.3], see also [MMR07, Section 4].

Example 4.1.10. Let R be a semiring and let M be a semimodule over R . Consider the natural action of the semiring $\text{Mat}_{n \times n}(R)$ on the set M^n , given by

$$((a_{ij}), (x_j)) \mapsto \left(\sum_{j=1}^n a_{ij} \cdot x_j \right).$$

Note that a special case of this example for $n = 1$ is the exponentiation in a cyclic group, see Example 2.1.5.

Remark 4.1.11. Consider the case when R is the ring $(\mathbb{Z}_\ell, +, \cdot)$ acting on an abelian group (M, \cdot) of order ℓ by exponentiation. The action is

$$\text{Mat}_{n \times n}(\mathbb{Z}_\ell) \times M^n \rightarrow M^n, \quad ((a_{ij}), (x_j)) \mapsto \left(\prod_{j=1}^n x_j^{a_{ij}} \right)_j.$$

Monico investigated the hardness of the SDL problem for this special case. He showed that there is a Pohlig-Hellman type reduction to the case $\ell = p^k$, where p is a prime. Also, if M is a cyclic group, the SDL problem reduces to several discrete logarithm problems in the group M .

However, the difficulty of the SDL problem in the general case is unclear. It might be possible to construct interesting semigroup actions out of a semimodule over a proper semiring.

Our second semigroup action was proposed by Maze, see [Maz03, Section 5.4] and [MMR07, Section 5]. It is a special case of a two-sided action, see Example 2.1.8, (1).

Example 4.1.12. Let R be a semiring, and let $M_n(R) := \text{Mat}_{n \times n}(R)$ be the semiring of $n \times n$ matrices. Consider the following two-sided semigroup action

$$\begin{aligned} \rho_n : (M_n(R) \times M_n(R)^{op}) \times M_n(R) &\rightarrow M_n(R), \\ ((A_1, A_2), X) &\mapsto A_1 X A_2. \end{aligned}$$

The SDL problem in this semigroup action seems to be hard to solve in general.

Commutative subsemirings of matrix semirings

Recall that several cryptographic applications of semigroup actions (Cryptosystems 2.3.6 and 2.3.8) depend on the ability to generate pairs of commuting elements of the semigroup. The semigroup actions of Examples 4.1.10 and 4.1.12 (as stated there) are not commutative. One possible method to generate commuting elements is to restrict the semigroup action to a commutative subsemigroup. This approach was pursued in the original work [Mon02, Maz03, MMR07].

In order to make the two-sided action of Example 4.1.12 commutative the authors provided a method for constructing commutative subsemirings of matrix semirings, which we outline below.

Definition 4.1.13. Let R be a semiring with center

$$C = \{r \in R \mid rs = sr \text{ for all } s \in R\},$$

and let $A \in \text{Mat}_{n \times n}(R)$ be a matrix. Define $C[A]$ to be the set of polynomials in A with coefficients in C .

Lemma 4.1.14. *Let R, C , and A as above. The set $C[A]$ is a commutative subsemiring of the matrix semiring $\text{Mat}_{n \times n}(R)$.*

Proof. The center C is a commutative subsemiring of R . Therefore, the polynomial semiring $C[x]$ over C is also commutative. Now $C[A]$ is the image of the semiring homomorphism

$$C[x] \rightarrow \text{Mat}_{n \times n}(R), \quad p(x) \mapsto p(A),$$

and thus a commutative subsemiring of $\text{Mat}_{n \times n}(R)$. \square

Remark 4.1.15. For security reasons one is interested in large commutative subsemirings $C[A]$.

- (1) If R is a commutative ring with one, the Cayley-Hamilton theorem (see e.g. [Bro93]) applies: We have $\chi_A(A) = 0$, where $\chi_A(x)$ is the characteristic polynomial of A . In particular,

$$|C[A]| = |R[A]| \leq |R|^{\deg \chi_A} \leq |R|^n.$$

- (2) A general lower bound is given by

$$|C[A]| \geq \text{ord}(A) := |\{A^i \mid i \in \mathbb{N}_0\}|,$$

the *order* of A . It can be shown (see e.g. [MMR07, Proposition 5.11]) that for any semiring R with one there exist matrices $A \in \text{Mat}_{n \times n}(R)$ having order $\geq g(n)$. Here, $g(n)$ is *Landau's function*, defined as

$$g(n) = \max\{\text{ord}(\sigma) \mid \sigma \in S_n\},$$

where S_n is the permutation group on n elements. Its asymptotic behaviour is $\log g(n) \sim \sqrt{n} \log n$, see [Lan03].

Towards a concrete cryptosystem

To propose concrete cryptosystems based on the two-sided semigroup action of Example 4.1.12 we have to specify (see Section 2.3):

- (1) A family of semigroup actions.
- (2) An instance generator that outputs a semigroup action instance (i, g) , depending on a security parameter k .
- (3) A pair (K_A, K_B) of compatible key generators.

For (1), we take the family of semigroup actions ρ_n of Example 4.1.12 using $n \times n$ matrices over a fixed semiring R .

For (2), we generate a semigroup action instance (i, g) . Here we may choose $i = k = n$, and as the generator g an arbitrary matrix $X \in \text{Mat}_{n \times n}(R)$.

For (3), the key generator $K = K_A = K_B$ depends on a choice of matrices A_1, A_2 of large order. It outputs matrices $M_1 = p_1(A_1) \in C[A_1]$ and $M_2 = p_2(A_2) \in C[A_2]$, e.g. by generating polynomials $p_1(x), p_2(x) \in C[x]$ of some bounded degree.

See [MMR07] for details. In particular, experiments using the simple semiring with 6 elements (see Example 3.2.5) showed that the sizes of the subsemirings $C[A]$ are usually much larger than the lower bound provided by Landau's function g .

4.2 Large endomorphism semirings

Another, novel approach to build families of semigroup actions is to start with a lattice $L = (L, \vee, \wedge)$ of moderate size. From that one constructs a huge simple semiring as the endomorphism semiring of the monoid (L, \vee) , according to Theorem 3.2.3.

The semiring operations cannot be stored explicitly in this case. They are provided implicitly by storing only the lattice structure. We note that by Proposition 4.1.5 the matrix-based approach of the previous section can be understood as a special case of the "lattice-based" approach presented here.

The cryptosystems are still under development and we present some open problems.

Example 4.2.1. We describe the endomorphism semirings $R = \text{End}(L, \vee)$ for some special cases of the lattice (L, \vee) .

- (1) Let L be a totally ordered set of order n , say $L = \{1, \dots, n\}$ and $a \vee b = \max(a, b)$ for $a, b \in L$. In this case the lattice endomorphisms

$f : L \rightarrow L$ are exactly the monotone functions such that $f(0) = 0$. We have:

$$|\text{End}(L, \vee)| = \binom{2n-2}{n-1} \sim c \frac{4^n}{\sqrt{n}}.$$

In particular, the order of $\text{End}(L, \vee)$ is exponential in the order of L .

- (2) Let L be a Boolean lattice, i.e. L is isomorphic to an n -fold direct product M^n of the lattice $M = \{0, 1\}$. By Proposition 4.1.5, $\text{End}(L) \cong \text{Mat}_{n \times n}(R)$, where $R = \text{End}(M)$ is the Boolean semifield. It follows:

$$|\text{End}(L, \vee)| = 2^{n^2},$$

whereas $|L| = 2^n$. With $m = |L|$ we thus have $|\text{End}(L, \vee)| = m^{\log_2 m}$.

- (3) Let X be a set with $n - 2$ elements, and let $L = X \cup \{0, 1\}$, ordered such that $0 \leq x$ and $x \leq 1$ for all $x \in X$, but x and y are incomparable for every $x, y \in X$, $x \neq y$, see Figure 4.1. Then L is a lattice of order n .

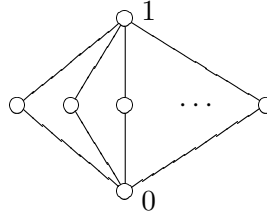


Figure 4.1: The lattice of Example 4.2.1, (3).

We derive a formula for $|\text{End}(L, \vee)|$ for this lattice L . A map $f : L \rightarrow L$ satisfying $f(0) = 0$ is an endomorphism of (L, \vee) if and only if

- (i) $f(x) \leq f(1)$ for all $x \in L$,
- (ii) $f(x) + f(y) = f(1)$ for all distinct $x, y \in X$.

Let $k := |X| = n - 2$. Now if f is nonzero we distinguish two cases:

- (a) $f(1) \in X$.

Let $z = f(1) \in X$. For all $x \in L$ by (i) we have $f(x) \in \{0, z\}$, and for all distinct $x, y \in X$ by (ii) we have $f(x) = z$ or $f(y) = z$. Therefore, $f(x) = z$ for all $x \in X$, except possibly one. This gives $k(k + 1)$ endomorphisms.

- (b) $f(1) = 1$.

By (ii) for all distinct $x, y \in X$ either one of $f(x), f(y)$ is 1 or $f(x), f(y)$ are distinct elements of X . Let $A := \{x \in X \mid f(x) \neq 1\}$. If $|A| = 1$, say $A = \{x\}$, then $f(x) \in X \cup \{0\}$ can be arbitrary, and if $|A| \geq 2$, then $f|_A : A \rightarrow X$ has to be injective. This gives $1 + k(k + 1)$ endomorphisms for $|A| \leq 1$ and $\sum_{j=2}^k \binom{k}{j} k \cdots (k - j + 1)$ endomorphisms for $|A| \geq 2$.

Let $a(k) := \sum_{j=0}^k \binom{k}{j} \frac{k!}{j!}$ be the number of partial injective transformations on a k -element set. Putting everything together we see

$$|\text{End}(L, \vee)| = a(k) + (k + 1)^2.$$

Clearly, $|\text{End}(L, \vee)| \geq a(k) \geq k!$. In fact, $a(k)$ is sequence no. A002720 in Sloane's on-line encyclopedia of integer sequences [Slo09], and it can be shown that

$$\frac{a(k)}{k!} \sim \frac{\exp(2\sqrt{k})}{2\sqrt{\pi e\sqrt{k}}}.$$

For security analysis of cryptosystems based on an endomorphism semiring it is important that one can estimate its size. Given a random lattice L we believe that it is very hard to find the size $|\text{End}(L, \vee)|$ exactly. Even giving some (tight) lower or upper bounds seems to be a nontrivial task.

Furthermore, one has to provide algorithms for random drawing of endomorphisms from $\text{End}(L, \vee)$.

One approach to tackle these questions practically is by a Monte-Carlo algorithm, given we are able to solve the following challenge: Is there a superset $S \supset \text{End}(L, \vee)$ such that

- (1) $|S|$ is computable,
- (2) uniform random drawing from S is feasible,
- (3) $|S|/|\text{End}(L, \vee)|$ is not too large?

For example, the set $S = L^L$ of all functions $L \rightarrow L$ would be a superset of $\text{End}(L, \vee)$ satisfying (1) and (2), but not (3).

We mention that one can probably exploit particular properties of lattice classes, like distributivity, to tackle these questions.

4.2.1 Cryptosystems using simple semirings

Lattices L of moderate size lead to large simple semirings $\text{End}(L, \vee)$ which can be used for new and interesting semigroup actions for cryptography. We illustrate this by an example.

Let L_A and L_B be lattices and L be the composed lattice $\frac{L_A}{L_B}$ (we identify the greatest element of L_B and the least element of L_A), see Figure 4.2. Consider the simple semiring $R = \text{End}(L, \vee)$.

On $A = R \times R$ define a semigroup operation $(f, g) \cdot (h, k) := (f \circ h, k \circ g)$. Then let A act on $X = R$ by the two-sided composition

$$A \times X \rightarrow X, \quad ((f, g), x) \mapsto f \circ x \circ g.$$

Now let R_A be the subsemiring of R consisting of all endomorphisms of L acting only on L_A and leaving L_B fixed. Define R_B similarly. Then it is easy to see that $f \circ g = g \circ f$ for all $f \in R_A$ and $g \in R_B$.

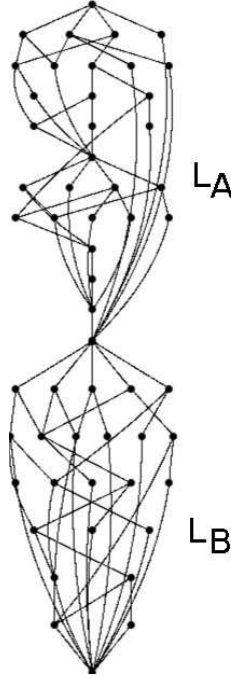


Figure 4.2: A decomposable lattice.

We define $C_A = R_A \times R_B$ and $C_B = R_B \times R_A$, which will be mutually commuting subsets of G . Then we can set up the key exchange protocol.

- Alice and Bob choose publicly $x \in X$.
- Alice privately chooses $f_A \in R_A$ and $g_A \in R_B$. She publishes $f_A \circ x \circ g_A$.
- Bob privately chooses $f_B \in R_A$ and $g_B \in R_B$. He publishes $g_B \circ x \circ f_B$.
- They both can compute their shared key $k = f_A \circ (g_B \circ x \circ f_B) \circ g_A = g_B \circ (f_A \circ x \circ g_A) \circ f_B$.

We have to investigate the security of this cryptosystem for different choices of the lattices L_A and L_B . It is also important that the endomorphism $x \in X = \text{End}(L, \vee)$ is chosen in such a way that a maximal “mixing” of elements in the upper half and the lower half of the lattice is provided.

Even though this example might already lead to a practical cryptosystem, we note that it is of rather preliminary nature. Indeed, the addition of the semiring R can be of significant benefit when looking for commuting elements. For this notice that if we have elements $a_i, b_j \in R$ such that $a_i b_j = b_j a_i$ for all i, j , then also

$$\left(\sum_i a_i \right) \left(\sum_j b_j \right) = \left(\sum_j b_j \right) \left(\sum_i a_i \right).$$

In fact, at this point it appears to be open if there are methods which are applicable to attack public-key cryptosystems involving both operations of a simple semiring.

To mention a final research problem at the end of this dissertation, there may well be other kinds of interesting lattices and methods to create mutually commuting subsets of endomorphisms, thus leading to new public-key cryptosystems. It is advisable to study several tools from semiring theory and lattice theory in detail to progress in this direction.

Bibliography

- [AAG99] Iris Anshel, Michael Anshel, and Dorian Goldfeld, *An algebraic method for public-key cryptography*, Math. Res. Lett. **6** (1999), no. 3-4, 287–291.
- [AB09] Sanjeev Arora and Boaz Barak, *Complexity theory: A modern approach*, Cambridge University Press, Cambridge, 2009, To appear.
- [BG99] Simon R. Blackburn and Steven D. Galbraith, *Cryptanalysis of two cryptosystems based on group actions*, Advances in cryptology—ASIACRYPT 1999, Lecture Notes in Comput. Sci., vol. 1716, Springer, Berlin, 1999, pp. 52–61.
- [Bir67] Garrett Birkhoff, *Lattice theory*, Third edition. American Mathematical Society Colloquium Publications, Vol. XXV, American Mathematical Society, Providence, R.I., 1967.
- [Bon98] Dan Boneh, *The decision Diffie-Hellman problem*, Algorithmic number theory—ANTS-III, Lecture Notes in Comput. Sci., vol. 1423, Springer, Berlin, 1998, pp. 48–63.
- [Bou51] Samuel Bourne, *The Jacobson radical of a semiring*, Proc. Nat. Acad. Sci. U. S. A. **37** (1951), 163–170.
- [Bro93] William C. Brown, *Matrices over commutative rings*, Monographs and Textbooks in Pure and Applied Mathematics, vol. 169, Marcel Dekker Inc., New York, 1993.
- [BWJM97] Simon Blake-Wilson, Don Johnson, and Alfred Menezes, *Key agreement protocols and their security analysis*, Cryptography and coding: 6th IMA international conference, Lecture Notes in Comput. Sci., vol. 1355, Springer, Berlin, 1997, pp. 30–45.
- [BZ57] Samuel Bourne and Hans Zassenhaus, *On a Wedderburn-Artin structure theory of a potent semiring*, Proc. Nat. Acad. Sci. U.S.A. **43** (1957), 613–615.

- [Cou06] Jean-Marc Couveignes, *Hard homogeneous spaces*, Cryptology ePrint Archive, Report 2006/291, 2006, <http://eprint.iacr.org/>.
- [CS03] Ronald Cramer and Victor Shoup, *Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack*, SIAM J. Comput. **33** (2003), no. 1, 167–226 (electronic).
- [Deh04] Patrick Dehornoy, *Braid-based cryptography*, Group theory, statistics, and cryptography, Contemp. Math., vol. 360, Amer. Math. Soc., Providence, RI, 2004, pp. 5–33.
- [DH76] Whitfield Diffie and Martin E. Hellman, *New directions in cryptography*, IEEE Trans. Information Theory **IT-22** (1976), no. 6, 644–654.
- [EHJK01] Robert ElBashir, Jan Hurt, Antonín Jančařík, and Tomáš Kepka, *Simple commutative semirings*, J. Algebra **236** (2001), no. 1, 277–306.
- [ElG85] Taher ElGamal, *A public key cryptosystem and a signature scheme based on discrete logarithms*, Advances in cryptology—CRYPTO 1984, Lecture Notes in Comput. Sci., vol. 196, Springer, Berlin, 1985, pp. 10–18.
- [FS87] Amos Fiat and Adi Shamir, *How to prove yourself: practical solutions to identification and signature problems*, Advances in cryptology—CRYPTO 1986, Lecture Notes in Comput. Sci., vol. 263, Springer, Berlin, 1987, pp. 186–194.
- [Geb06] Volker Gebhardt, *Conjugacy search in braid groups: from a braid-based cryptography point of view*, Appl. Algebra Engrg. Comm. Comput. **17** (2006), no. 3-4, 219–238.
- [Gol99] Jonathan S. Golan, *Semirings and their applications*, Kluwer Academic Publishers, Dordrecht, 1999.
- [Gol01] Oded Goldreich, *Foundations of cryptography. basic tools*, Cambridge University Press, Cambridge, 2001.
- [Gol04] ———, *Foundations of cryptography. II. basic applications*, Cambridge University Press, Cambridge, 2004.
- [Gol08] ———, *Computational complexity. a conceptual perspective*, Cambridge University Press, Cambridge, 2008.
- [Her68] Israel N. Herstein, *Noncommutative rings*, The Carus Mathematical Monographs, No. 15, Published by The Mathematical Association of America, 1968.

- [HS02] Dennis Hofheinz and Rainer Steinwandt, *A practical attack on some braid group based cryptographic primitives*, Public key cryptography—PKC 2003, Lecture Notes in Comput. Sci., vol. 2567, Springer, Berlin, 2002, pp. 187–198.
- [HW93] Udo Hebisch and Hanns Joachim Weinert, *Halbringe. Algebraische Theorie und Anwendungen in der Informatik*, Teubner Studienbücher Mathematik, B. G. Teubner, Stuttgart, 1993.
- [HW98] ———, *Semirings: algebraic theory and applications in computer science*, Series in Algebra, vol. 5, World Scientific Publishing Co. Inc., River Edge, NJ, 1998.
- [Iiz59] Kenzo Iizuka, *On the Jacobson radical of a semiring*, Tôhoku Math. J. (2) **11** (1959), 409–421.
- [KL08] Jonathan Katz and Yehuda Lindell, *Introduction to modern cryptography*, Chapman & Hall/CRC Cryptography and Network Security, Chapman & Hall/CRC, Boca Raton, FL, 2008.
- [KLC⁺00] Ki Hyoung Ko, Sang Jin Lee, Jung Hee Cheon, Jae Woo Han, Ju-sung Kang, and Choonsik Park, *New public-key cryptosystem using braid groups*, Advances in cryptology—CRYPTO 2000, Lecture Notes in Comput. Sci., vol. 1880, Springer, Berlin, 2000, pp. 166–183.
- [Lan03] Edmund Landau, *Über die Maximalordnung der Permutationen gegebenen Grades*, Arch. Math. Phys. (3) **5** (1903), 92–103.
- [Maz03] Gérard Maze, *Algebraic methods for constructing one-way trapdoor functions*, Ph.D. thesis, University of Notre Dame, 2003, available at <http://www.math.uzh.ch/rosen>.
- [MF88] Sidney S. Mitchell and Paul B. Fenoglio, *Congruence-free commutative semirings*, Semigroup Forum **37** (1988), no. 1, 79–91.
- [MMR07] Gérard Maze, Chris Monico, and Joachim Rosenthal, *Public key cryptography based on semigroup actions*, Adv. Math. Commun. **1** (2007), no. 4, 489–507.
- [Mon02] Chris Monico, *Semirings and semigroup actions in public-key cryptography*, Ph.D. thesis, University of Notre Dame, 2002, available at <http://www.math.uzh.ch/rosen>.
- [Mon04] ———, *On finite congruence-simple semirings*, J. Algebra **271** (2004), no. 2, 846–854.

- [MSU05] Alexei Myasnikov, Vladimir Shpilrain, and Alexander Ushakov, *A practical attack on a braid group based cryptographic protocol*, Advances in cryptology—CRYPTO 2005, Lecture Notes in Comput. Sci., vol. 3621, Springer, Berlin, 2005, pp. 86–96.
- [MSU08] ———, *Group-based cryptography*, Advanced Courses in Mathematics. CRM Barcelona, Birkhäuser Verlag, Basel, 2008.
- [MvOV97] Alfred Menezes, Paul van Oorschot, and Scott Vanstone, *Handbook of applied cryptography*, CRC Press Series on Discrete Mathematics and its Applications, CRC Press, Boca Raton, FL, 1997.
- [Pap94] Christos H. Papadimitriou, *Computational complexity*, Addison-Wesley Publishing Company, Reading, MA, 1994.
- [PHK⁺01] Seong-Hun Paeng, Kil-Chan Ha, Jae Heon Kim, Seongtaek Chee, and Choonsik Park, *New public key cryptosystem using finite nonabelian groups*, Advances in cryptology—CRYPTO 2001, Lecture Notes in Comput. Sci., vol. 2139, Springer, Berlin, 2001, pp. 470–485.
- [PKHK01] Seong-Hun Paeng, Daesung Kwon, Kil-Chan Ha, and Jae Heon Kim, *Improved public key cryptosystem using finite nonabelian groups*, Cryptology ePrint Archive, Report 2001/066, 2001, <http://eprint.iacr.org/>.
- [PS00] David Pointcheval and Jacques Stern, *Security arguments for digital signatures and blind signatures*, J. Cryptology **13** (2000), no. 4, 361–396.
- [Rot73] Joseph J. Rotman, *The theory of groups. An introduction*, second ed., Allyn and Bacon Inc., Boston, Mass., 1973, Allyn and Bacon Series in Advanced Mathematics.
- [RS84] Christophe Reutenauer and Howard Straubing, *Inversion of matrices over a commutative semiring*, J. Algebra **88** (1984), no. 2, 350–360.
- [RST07] Dima Ruinskiy, Adi Shamir, and Boaz Tsaban, *Length-based cryptanalysis: the case of thompson’s group*, J. Math. Cryptol. **1** (2007), no. 4, 359–372.
- [Sch90] Claus Peter Schnorr, *Efficient identification and signatures for smart cards*, Advances in cryptology—CRYPTO 1989, Lecture Notes in Comput. Sci., vol. 435, Springer, New York, 1990, pp. 239–252.

- [SDG02] Hervé Sibert, Patrick Dehornoy, and Marc Girault, *Entity authentication schemes using braid word reduction*, Cryptology ePrint Archive, Report 2002/187, 2002, <http://eprint.iacr.org/>.
- [Sha49] Claude E. Shannon, *Communication theory of secrecy systems*, Bell System Tech. J. **28** (1949), 656–715.
- [Sho97] Victor Shoup, *Lower bounds for discrete logarithms and related problems*, Advances in cryptology—EUROCRYPT 1997, Lecture Notes in Comput. Sci., vol. 1233, Springer, Berlin, 1997, pp. 256–266.
- [Slo09] Neil J. A. Sloane, *The on-line encyclopedia of integer sequences*, no. A002720, 2009, <http://www.research.att.com/~njas/sequences/>.
- [SU06] Vladimir Shpilrain and Alexander Ushakov, *A new key exchange protocol based on the decomposition problem*, Algebraic methods in cryptography, Contemp. Math., vol. 418, Amer. Math. Soc., Providence, RI, 2006, pp. 161–167.
- [SZ06] Vladimir Shpilrain and Gabriel Zapata, *Combinatorial group theory and public key cryptography*, Appl. Algebra Engrg. Comm. Comput. **17** (2006), no. 3-4, 291–302.
- [Tob02] Christian Tobias, *Security analysis of the MOR cryptosystem*, Public key cryptography—PKC 2003, Lecture Notes in Comput. Sci., vol. 2567, Springer, Berlin, 2002, pp. 175–186.
- [Van34] Harry S. Vandiver, *Note on a simple type of algebra in which the cancellation law of addition does not hold*, Bull. Amer. Math. Soc. **40** (1934), no. 12, 914–920.
- [Vau06] Serge Vaudenay, *A classical introduction to cryptography: Applications for communications security*, Springer, New York, 2006.
- [Win93] Reinhard Winkler, *On maximal abelian groups of maps*, J. Austral. Math. Soc. Ser. A **55** (1993), no. 3, 414–420.
- [Yam98] Akihiro Yamamura, *Public-key cryptosystems using the modular group*, Public key cryptography—PKC 1998, Lecture Notes in Comput. Sci., vol. 1431, Springer, Berlin, 1998, pp. 203–216.
- [Yam99] ———, *A functional cryptosystem using a group action*, ACISP 1999: 4th Australasian conference on information security and privacy, Lecture Notes in Comput. Sci., vol. 1587, Springer, Berlin, 1999, pp. 314–325.

- [Zum08] Jens Zumbärgel, *Classification of finite congruence-simple semi-rings with zero*, J. Algebra Appl. **7** (2008), no. 3, 363–377.

Index

- A-set, 29
- \mathcal{C} , cipher space, 2
- \mathcal{K} , key space, 2
- \mathcal{M} , message space, 2
- \mathcal{S} , signature space, 3
- \mathcal{X} , probability space, 2
- adaptive attack, 13
- algorithm, 7
 - efficient, 8
 - probabilistic, 8
- Artin group, 56
 - of extra large type, 56
- asymptotic approach, 7
- attack model, 5
- authentication, 1
- bi-ideal, 60
- Boolean semifield, 59
- braid group, 52
- CDH problem, 22
- chosen ciphertext attack, 11
- chosen message attack, 13
- chosen plaintext attack, 11
- Church-Turing thesis, 8
- ciphertext only attack, 11
- cipher space, 2
- collection
 - of one-way functions, 16
 - of one-way trapdoor functions, 17
- compatible key generators, 41
- completeness, 25
- computational Diffie-Hellman problem, 22
- computational model, 7
- computational security, 6
- congruence, 59, 62
- congruence-simple, 62
- conjugacy problem, 51
- conjugator search problem, 51
- cryptography, 1
- cryptosystem, 4
 - efficient, 9
- DDH problem, 22
- decision Diffie-Hellman problem, 22
- decomposition problem, 54
- decryption function, 2
- dense subsemiring, 64
- Diffie-Hellman key agreement, 21
- Diffie-Hellman problem
 - computational, 22
 - decision, 22
- digital signature scheme, 4
 - public-key, 10
 - symmetric, 4
- discrete logarithm problem, 20
- DL problem, 20
- domain, 18
- dual semigroup, 31
- efficient algorithm, 8
- efficient cryptosystem, 9
- encryption function, 2
- encryption scheme, 2
 - deterministic, 2
 - probabilistic, 2
 - public-key, 10
 - symmetric, 2
- endomorphism semiring, 64
- existential forgery, 14

- extended semigroup action, 37
- family of semigroup actions, 40
- forgery
 - existential, 14
 - selective, 14
 - universal, 14
- function problem, 18
- fundamental domain, 50
- generator
 - of an A set, 30
- group action, 30
- group family, 19
- group instance, 19
- GSCDH problem, 35
- homomorphism of semirings, 59
- ideal, 60
- identification protocol, 24
- indistinguishability, 6
 - polynomial, 12
- instance, 7, 18
- instance generator, 19, 40
- integrity, 1
- intractable, 19
- invertible matrix, 81
- k -ideal, 60
- Kerckhoffs' principle, 2
- key agreement protocol, 10
- key generator, 4
 - scalable, 9
- key pair, 2
- key space, 2
- known plaintext attack, 11
- Landau's function, 84
- left congruence, 33
- malleability, 13
- matrix product, 78
- matrix semiring, 78
- message-specification function, 24
- message authentication scheme, 4
- message space, 2
- modular group, 49
- monogenic A set, 30
- multiple conjugator search problem, 53
- negligible function, 12
- non-adaptive attack, 13
- non-malleability, 13
- one, 58
- one-time pad, 3
- one-way function, 14
- one-way functions, 16
- one-way trapdoor functions, 17
- order, 58, 84
- polynomially bounded, 18
- polynomial indistinguishability, 12
- primitives, 14
- probabilistic algorithm, 8
- problem, 7
- problem instance, 7
- proof of knowledge, 25
- proper ideal, 60
- protocol, 24
- Rabin function, 17
- reduction, 8
- right semigroup action, 31
- ring, 58
- RSA function, 17
- running time, 7
- SCDH problem, 33
- Schnorr identification, 26
- SDDH assumption, 42
- SDDH problem, 34
- SDL problem, 33
- search problem, 18
- secrecy, 1
- secret (proof of knowledge), 24
- security
 - indistinguishable, 6
 - perfect, 5

- polynomial indistinguishable, 12
- semantic, 13
- unconditional, 6
- security parameter, 9
- security goal, 5
- selective forgery, 14
- semifield, 58
- semigroup action, 29
 - Diffie-Hellman function, 35
 - Diffie-Hellman key agreement, 43
 - ElGamal encryption, 44
 - instance, 40
 - Schnorr signature, 49
 - ZK-based identification, 46
- semimodule, 61
- semimodule congruence, 62
- semiring, 58
- semiring congruence, 59
- semitransitivity, 30
- signature space, 3
- signing function, 3
- simple, 62
- simple transitivity, 30
- solution, 18
- solve, 18
- soundness, 25
- special conjugacy problem, 54
- SSCDH problem, 35
- standard fundamental domain, 50
- strong Church-Turing thesis, 8
- subsemimodule, 62
- subsemiring, 58

- transitivity, 30
- trapdoor, 17
- two-sided semigroup action, 32

- unforgeability
 - existential, 14
- universal forgery, 14
- verification function, 4
- word problem, 51
- zero, 58
- zero-knowledge, 25
- ZK proof of knowledge, 25