CODING THEORY

EDWARD O'CALLAGHAN

Contents

1. Prelude

TODO: Fix notation here...

2. Introduction

In this course we builds up the rudiments of the theoretical framework that unpins many modern day coding methodology. Applications such as ISBN, RSA, bar codes and error correcting with polynomials over finite groups, typically used in things such as wifi IEEE 802.11*. Historical context is given where possible to provide the motivation behind the developments in cryptography.

3. Arithmetic

3.1. **Modular arithmetic.**

Modular arithmetic is the arithmetic of forgetting. This notion of forgetful arithmetic will become apparent though the following examples given with integers. Note however, that we do not necessarily have to ”read modulo” an integer (e.g., read modulo a polynomial) and we shall explore this in more depth later.

Here are some motivating examples:

Example 3.1.

$$6 \equiv 1 \pmod 5 \tag{1}$$

$$7 \equiv 2 \pmod 5 \tag{2}$$

$$13 \equiv 3 \pmod{10} \tag{3}$$

TODO...

3.2. **Primality testing.**

TODO.

3.3. **Prime number generation.**

TODO.

3.4. **Random number generation.**

TODO.

3.5. **Factorising.**

TODO.