# Research Internship
# Project Plan

Joan Daemen U167219

Tim van Dijk S4477073

September 2018

## 1 Practical information

The research internship takes place at the Digital Security Group of the Radboud University. The intern, Tim van Dijk, is supervised by Joan Daemen, who is a member of this group. The first couple of weeks, Joan Daemen also takes on the task of daily supervisor, although this task may be delegated to a PhD student in the future.

The research internship is done during the first semester of academic year 2018-2019. This semester is from September 3, 2018 until Februari 1, 2019. In this period of 22 weeks, the intern works half-time for about 2.5 days a week for a total of 420 hours (which equals 15 EC). Weekly, the intern will spend at least one day working physically at the Digital Security Group. The remaining hours can be worked from home.

During the research internship, the intern will assist in carrying out research activities at the Digital Security Group, primarily regarding symmetric cryptography over fields of odd characteristic. The research will be in the following domains:

- Efficient bit-sliced implementations of addition, multiplication and power functions in these fields.
- Development of tools for cryptanalysis and determining the strength of ciphers against attacks. This includes efficient generation of difference distribution tables and correlation matrices using multidimensional Fourier transforms.

## 2 Background information

Modern symmetric cryptographic functions operate on strings of binary-valued symbols called *bits*. In particular, for encrypting, hashing or computing a MAC over a message with modern crypto, one must code it as a bitstring. The result,

a ciphertext, digest or MAC is also a bitstring. This makes sense as in modern IT systems all data are encoded as bit strings.

Typically the cryptographic operations are performed with schemes that consist of a *mode of use* of some cryptographic *primitive* that operates on bitstrings of fixed length. The best known example of such primitives are block ciphers such as DES and AES, but more recently also permutations have become popular primitives. Cryptographic permutations define invertible transformations on the space of $b$-bit strings, for some fixed value of $b$. Well-known permutations are the ones underlying Salsa and ChaCha, both with width $b = 512$ and the 7 Keccak-f permutations of width 25, 50, 100, 200, 400, 800 and 1600 bits. These block ciphers and permutations operate in an iterated way: they consist of the repeated application of a relatively simple round function to a state, alternated with the (binary bitwise) addition of secret round keys (in the case of a block cipher) or round constants (in the case of a permutation). This state consists of an array of $b$ bits.

The round functions as described above typically have three layers:

- Non-linear layer: bits are combined in a non-linear way
- Mixing layer: bits are combined in a linear way
- Transposition layer: bits are moved away from each other

These layers are chosen and combined in such a way that there is no exploitable differential propagation from input to output or exploitable correlations between input and output. This is called the *wide trail strategy*. The choice of the operations is guided by a design strategy and knowledge of (multi-variate) functions in the finite field GF(2) or extension fields thereof such as GF($2^8$). For example AES makes use of the multiplicative inverse in GF($2^8$) as non-linear layer and of MDS matrices over GF($2^8$) for the mixing layer. The relevant properties of these functions have been studied extensively by an expert community of mathematicians, leading to solid designs. Other operations that have been used in symmetric crypto are the binary Golay code and so-called almost perfect nonlinear functions operating on arrays of bits.

Now, this community investigating functions over GF(2) and GF($2^n$) do not stop at the binary case but also study similar functions over GF(q) with $q$ a prime or a prime power. Often, results can be generalized and this may even give new insights. On the other hand, there are also interesting differences between the binary case and the odd-prime case and to a certain extent the fields of odd characteristic are richer in functionality than binary fields.

The idea of this work is to try to capitalize on this rich field of functions over GF($q$) by defining permutations operating on arrays of elements of GF($q$). We will simply apply the wide trail strategy and select interesting non-linear and mixing layers from the rich literature. This comes with lots of subtleties.

For example, in the GF(2) case, addition and subtraction are the same, in GF($q$) this is no longer the case. In GF($2^n$) squaring is a linear operation. In GF($q$) squaring is in a certain sense an optimally nonlinear operation. In GF(2)

correlations between input and output bits have values that are rational and range from $-1$ to $+1$, in GF($q$) correlations are complex numbers that are in the unit disk. It will be interesting to see how this affects the description of differential propagation and correlations.

Another thing is that in practice of course the data we work on will continue to be represented as bit strings. This is not incompatible with modes where the input is *absorbed* into a state and output is *squeezed* from the state, such as in the sponge construction. We must investigate how to absorb binary data into a state consisting of elements of GF($q$) in a sound way. This seems not so difficult. The operation of squeezing seems more difficult: we must squeeze a sequence of bits, preferably balanced from a state of elements of GF($q$). A first analysis shows that here we can make use of techniques from the field of error-correcting codes.