Cryptography based on algebraic perpendicularities

TIMO TOSSAVAINEN

1. *Introduction*

Cryptography, the mathematics of protecting secret or sensitive information, is a continuously evolving area of interest. A concrete example of this is the fact that worldwide spending on information security and risk management technology and services has been estimated to reach over $150 billion in 2021. Modern cryptography is actually not a single separate domain of mathematics but an advanced encryption scheme can be based on applications of results in number theory, such as the Euler–Fermat Theorem, or involve the discrete logarithm problem using either a primitive root of a large prime or an element of an elliptic curve over a prime field. In the background, probability theory, statistics, studies on computational models and finite geometries, etc. play a major role. Recent research has considered even DNA-based molecular cryptography systems.

To get a grasp of how a modern encryption scheme works, let us revise an example from [1]. This example deals with a public-key cryptosystem called RSA and it exploits modular arithmetic. Another fundamental premise is the mathematical fact that it is easy to find two, say, 125-digit primes but it is much harder to factorise 250-digit numbers into prime numbers. In February 2020, a 250-digit product of two primes was successfully factorised and it was announced that the process took computing time equivalent to using 2700 computers continuously for a full year.

First, we need a modulus for the public and private keys: $m = pq$ where p and q are different primes such that $m \geqslant 26$, the number of letters in the english alphabet. Then we compute the totient $t = (p-1)(q-1)$ and choose a coding power $1 < c < t$ which has to be coprime with t. Following [1, p. 201], we take $m = 3 \cdot 11 = 33$ and $c = 3$. Next, we convert letters to numbers by setting $A = 01, B = 02, C = 03$ etc. Then, for instance,

HAVE A NICE DAY

becomes

08 01 22 05 01 14 09 03 05 04 01 25.

To encrypt this message with aid of the public keys c and m, we take the cubes of these numbers modulo 33:

17 01 22 26 01 05 03 27 26 31 01 16.

To decrypt this sequence, we need the private key d which is found by solving the congruence

$$cd \equiv 1 \pmod{t},$$

i.e. $3d \equiv 1 \pmod{20}$. Now $d = 7$. Taking the seventh powers modulo 33 of successive pairs of digits in the encrypted message gives us the original message in its numerical form because

$$x^{cd} \equiv x \pmod{m}$$

https://doi.org/10.1017/mag.2023.8 Published online by Cambridge University Press

for all x. The usefulness of RSA is based on the fact that, when m is a very large number, m and c can be made public so that anyone can send an encrypted message to the owner of these numbers but the owner is the only one who is able to decrypt the message as he or she is the only one who knows p and q which are needed for computing d.

However, a weakness in the above example and in many traditional encryption schemes is that each letter, number, or another symbol is always substituted by the same (obviously, other than the original) symbol. The invention of frequency analysis techniques sometime in the early ninth century made these schemes vulnerable. If the encrypted text is long enough, the cipher can be broken by comparing the letter frequencies of a natural language with the symbol frequencies in the cipher.

There is an encryption scheme which is suitable to be discussed at early undergraduate level and yet it does not have the above-mentioned weakness of the traditional schemes. It is based on using perpendicularity relations in an Abelian group and serves well as supplementary material in a first course in abstract algebra. Before looking at the scheme, let us record a few basic facts about such perpendicularities. To learn more about them, the reader can consult the articles [2, 3] listed at the end of this paper. The reference list also contains a couple of articles published in the *Gazette* which serve well as an introduction to the basics of cryptography [1, 4].

2. *Perpendicularities*

Let $G = (G, +)$ be an Abelian group. We call $\perp$ a *perpendicularity* in G if $\perp$ is a binary relation in G satisfying

(A1) $\forall a \in G : \exists b \in G : a \perp b$,

(A2) $\forall a \in G \setminus \{0\} : a \not\perp a$,

(A3) $\forall a, b \in G : a \perp b \Rightarrow b \perp a$,

(A4) $\forall a, b, c \in G : a \perp b \wedge a \perp c \Rightarrow a \perp (b + c)$,

(A5) $\forall a, b \in G : a \perp b \Rightarrow a \perp -b$.

It is easy to see that A1–A5 can be derived from the basic properties of an inner product given that the inner product of orthogonal vectors equals zero. The above definition would make sense also for a general group, but it is more convenient if $b + c = c + b$ for all $b, c \in G$. Anyway, $a \perp b \wedge a \perp c$ implies both $a \perp (b + c)$ and $a \perp (c + b)$.

For every Abelian group, there is the *trivial perpendicularity*

$$x \perp_0 y \Leftrightarrow x = 0 \vee y = 0.$$

Most Abelian groups have also non-trivial perpendicularities. For example, if we define $0 \perp 0$, 1, 2, 3, 4, 5 and $3 \perp 2$, 4, and vice versa, for the elements of the cyclic group $\mathbb{Z}_6$, then $\perp$ is a perpendicularity in $\mathbb{Z}_6$.

We call $\perp$ *maximal* if it is not a subrelation of any other perpendicularity in G. Also a maximal perpendicularity exists for every Abelian group. This follows from Zorn's lemma; if $\perp_0 \subseteq \perp_1 \subseteq \ldots$ are

https://doi.org/10.1017/mag.2023.8 Published online by Cambridge University Press

perpendicularities in G, then $\cup_{i=0}^{\infty} \perp_i$ is a perpendicularity in g. An Abelian group may have more than one maximal perpendicularity. The perpendicularity in $\mathbb{Z}_6$ discussed above is maximal.

3. *The encryption scheme*

Now we are ready to discuss the encryption scheme. It has been shown that there are infinitely many Abelian groups that have infinitely many maximal perpendicularities [3]. For simplicity, we consider $G = (\mathbb{Z}^2, +)$.

Let $\mathbf{e}_1, \mathbf{e}_2 \in \mathbb{Z}^2$ so that

$$G = \langle \mathbf{e}_1 \rangle \oplus \langle \mathbf{e}_2 \rangle,$$

i.e. every element of G can be expressed as a direct sum $m\mathbf{e}_1 + n\mathbf{e}_2$, where $m, n \in \mathbb{Z}$. Observe that there are infinitely many ways to choose these 'base elements'. Then a maximal perpendicularity related to $\mathbf{e}_1$ and $\mathbf{e}_2$ can be defined via

$$x \perp y \Leftrightarrow ac + bd = 0, \tag{1}$$

where $x = a\mathbf{e}_1 + b\mathbf{e}_2$ and $y = c\mathbf{e}_1 + d\mathbf{e}_2$.

Let us choose, as an example, $\mathbf{e}_1 = (1, 0)$, $\mathbf{e}_2 = (-2, 1)$, $x = (-1, 1)$ and $y = (6, -2)$. Then $x \perp y$ because

$$x = 1\mathbf{e}_1 + 1\mathbf{e}_2, \qquad y = 2\mathbf{e}_1 - 2\mathbf{e}_2,$$

and $1 \cdot 2 + 1 \cdot -2 = 0$. On the other hand, if we had chosen, say, $\mathbf{e}_1 = (1, 0)$ and $\mathbf{e}_2 = (2, 1)$, then $x \not\perp y$ because now

$$x = -3\mathbf{e}_1 + 1\mathbf{e}_2, \qquad y = 10\mathbf{e}_1 - 2\mathbf{e}_2,$$

and $-3 \cdot 10 + 1 \cdot -2 \neq 0$. This example demonstrates that, except for certain obvious cases, it is impossible to say whether or not two non-zero elements x and y are perpendicular to one another if one does not know $\perp$. So, if $\perp$ is known only by a sender and a receiver, it works well with encrypting a bit by taking a pair of perpendicular elements for 0 or a pair (x, y) with $x \not\perp y$ for 1. There are infinitely many ways of doing this: for each $x \in \mathbb{Z}^2$, there are infinitely many $y, z \in \mathbb{Z}^2$ such that $x \perp y$ and $x \not\perp z$. For instance, if $x = a\mathbf{e}_1 + b\mathbf{e}_2$, then it suffices to select any $m \in \mathbb{Z}$ and set $c = mb$ and $d = -ma$ in $y = c\mathbf{e}_1 + d\mathbf{e}_2$ to have $x \perp y$. In other words, one never needs to use the same pair twice for encrypting the same bit. Moreover, choosing a pair does not depend on the selection of the previous pairs.

For those who know the correct $\perp$, the decrypting of

$$(x_1, y_1), \ldots . (x_k, y_k), \qquad x_i, y_i \in G \tag{2}$$

into a sequence of k bits is an easy procedure. It suffices to compute a_i, b_i, c_i and d_i for each (x_i, y_i) – which is also a simple problem of solving a system of linear equations – and then apply the rule

$$(x_i, y_i) \rightarrow \begin{cases} 0, & a_i c_i + b_i d_i = 0 \\ 1, & a_i c_i + b_i d_i \neq 0 \end{cases}.$$

https://doi.org/10.1017/mag.2023.8 Published online by Cambridge University Press

However, there is a vulnerability in the above scheme. If the encryption perpendicularity is constructed as above and an 'enemy' succeeds in finding out that $x \perp y$ for certain $x, y \in G \setminus \{0\}$, then he or she can – again by solving a system of linear equations – determine $\perp$ uniquely. In practice, given a sequence (2), the 'enemy' may simply start testing the pairs (x_i, y_i), $i = 1, 2, 3, \ldots$, assuming that $x_i \perp y_i$ holds and construct a corresponding perpendicularity $\perp_i$. Sooner or later he or she gets one right (otherwise, the secret message would consist only of 1's). On the other hand, if the encoding of the original message into a binary sequence does not follow any character encoding standard such as ASCII, it may be difficult for the 'enemy' to know which $\perp_i$ is the correct key; after all, each decrypting gives only a sequence of zeros and ones which are not yet meaningful messages as such.

Not all maximal perpendicularities in $\mathbb{Z}^2$ arise by only choosing $\mathbf{e}_1 = \pm(1, 0)$ and $\mathbf{e}_2 = \pm(z, 1)$, $z \in \mathbb{Z}$ but there are maximal perpendicularities of another kind, too [4]. The existence of various types of maximal perpendicularities makes the scheme more secure because the finding of the correct perpendicularity becomes more difficult for the 'enemy'.

Decrypting can be made harder for the 'enemy', without making the encrypting of the original binary data significantly more complicated, also by increasing the number of applied perpendicularities. As the number of maximal perpendiculaties in $\mathbb{Z}^2$ is infinite, we may select a unique perpendicularity for each bit, or take $m > 1$ perpendicularities $\perp_1, \ldots \perp_m$ and decide that the i th pair in (2) is en- and decrypted by using $\perp_j$, where $j \equiv i + n \pmod{m}$ and $n \in \{0, \ldots, m - 1\}$. The complexity of decrypting increases radically as k and m increase. Another way to confuse the 'enemy' is to embed $\mathbb{Z}^2$ (more generally, the Abelian group in use) in a larger mathematical structure.

It almost seems to be 'a law of nature' that every encryption scheme has some vulnerabilities or is immeasurably complicated to use. An apparent weakness of the above scheme is that the applied perpendicularities cannot be made public. If the 'enemy' succeeds to find them out, information on a new set of perpendicularities can no longer be distributed via this scheme. Another fragility is that all users of the scheme must know all actual perpendicularities. This means that the scheme does not allow private communication within the group of the users like RSA does. Therefore it is most suitable for encrypting information that is intended only for a small and internally open community and needs to be kept safe only temporarily.

4. *Discussion and a task*

The reader may ask why the above encryption scheme has been introduced by speaking of Abelian groups; it could also have been done by speaking of the bases of $\mathbb{R}^2$ and the orthogonality of the plane vectors. The answer is: for the sake of practicality. Concerning applications of mathematics in the real world, we have to acknowledge the fact that

https://doi.org/10.1017/mag.2023.8 Published online by Cambridge University Press

computers ultimately do not operate on real numbers. Finite Abelian groups are more compatible with the real computing systems than real numbers.

Actually, the real usefulness of the above scheme depends on how rich perpendicularity structures a finite Abelian group can have. We know [3, Proposition 1] that, if G is cyclic, then it has a unique maximal perpendicularity. Consequently, such a group is not useful for protecting information. But a non-cyclic finite Abelian group may have an impressive number of maximal perpendicularities – and other non-trivial perpendicularities too. Already the smallest non-cyclic group, the Klein four-group, has three non-trivial perpendicularities and all of them are maximal [2, Example 7]. Another motivation for speaking of Abelian groups in this context is the fact that an Abelian group underlies many fundamental algebraic structures, and also $\mathbb{R}^2$. Thus the above scheme is compatible with all of them.

To demonstrate the fascination (and, perhaps, occasionally occurring frustration) with decrypting secret messages, I share a code that represents one of my favourite numbers as binary number.

$$\big((5,\ 2)\,(0,\ 2)\big),\ \big((5,\ 3)\,(-5,\ -2)\big),\ \big((-1,\ -3)\,(4,\ 2)\big),\ \big((-1,\ \ 3)\,(-17,\ 7)\big).$$

Can you find out what it is? How much would it help if I revealed that it is even?

References

1. J. B. Reade, Modular arithmetic and cryptography, *Math. Gaz.* **72** (October 1988) pp. 198-202.
2. P. Haukkanen, M. Mattila, J.K. Merikoski, and T. Tossavainen, Perpendicularity in an Abelian group, *Internat. J. Math. Math. Sci.* (2013), Article 983607.
3. M. Mattila, P. Haukkanen, J. K. Merikoski, and T. Tossavainen, Maximal perpendicularity in certain Abelian groups, *Acta Univ. Sapientiae, Math.* **9** (2017) pp. 235-247.
4. R. E. Lewand, The perfect cipher, *Math. Gaz.* **94** (November 2010), pp. 401-411.

TIMO TOSSAVAINEN
Department of Health,
Education and Technology,
Lulea University of Technology, 97187 Lulea, Sweden
e-mail: *timo.tossavainen@ltu.se*

10.1017/mag.2023.8