# A Zero-knowledge Undeniable Signature Scheme in Non-abelian Group Setting

Tony Thomas and Arbind Kumar Lal
*(Corresponding author: Tony Thomas)*

Department of Mathematics and Statistics, Indian Institute of Technology Kanpur
Kanpur, Uttar Pradesh, 208016 India (Email: tonythomasiit@yahoo.com)

*(Received May 4, 2006; revised and accepted May 25, 2006 & July 31, 2006)*

## Abstract

Recently non-abelian groups have attracted the attention of cryptographers for constructing public-key cryptographic protocols. In this paper we use the conjugacy problem in non-abelian groups to construct a zero-knowledge undeniable signature scheme.

*Keywords: Conjugacy problem, digital signature, non-abelian groups, undeniable signatures*

## 1 Introduction

Digital signatures bind signers to the contents of the document they sign. The ability for a third party to verify the validity of a signature is usually seen as the basis for the non-repudiation aspect of digital signatures. The authenticity of a digital signature can be verified by anyone having the public key of the signer. However, this universal verifiability property of digital signatures is not always a desirable property. Such is the case of a signature binding parties to a confidential agreement, or of a signature on documents carrying private or personal information.

Chaum and van Antwerpen [7] introduced the concept of *undeniable signatures* for limiting the ability of third parties to verify the validity of a signature. An undeniable signature, like digital signature depends on the signer's public key as well as on the message signed. Such signatures are characterized by the property that, verification can only be achieved by interacting with the legitimate signer through a *confirmation protocol.* On the other hand, the signer can prove a forgery by engaging in a *denial protocol.* If the signer does not succeed in denying (in particular, if it refuses to cooperate) then the signer remains legally bound to the signature. On the other hand the signer is protected by the fact that his signature cannot be verified by unauthorized third parties without his own cooperation.

Recently, several public key cryptographic protocols based on symbolic computations in non-abelian groups have been proposed [1, 2, 9, 11, 12, 15, 16, 19, 20, 21] as a more efficient alternative to well established numeric computations in abelian groups. Where as the security of cryptographic protocols based on number-theoretic abelian groups are based on the hardness of problems like integer factorization and discrete logarithm, the security of cryptographic protocols based on non-abelian groups are based on the hardness of problems like conjugacy search, decomposition and root problem. Almost all the undeniable signature schemes constructed so far have been based on the hardness of integer factorization [10] and discrete logarithm problems [6, 7]. In this paper, we present a zero-knowledge undeniable signature scheme based on the hardness of the conjugacy problem in non-abelian groups.

The outline of the paper is as follows. In Section 2, we describe the preliminaries needed for this paper. A zero-knowledge undeniable signature scheme is given in Section 3. We prove the completeness, soundness and zero-knowledgeness of the protocols also. In Section 4, we suggest some non-abelian groups for the implementation of the above signature scheme. The paper concludes with some general remarks in Section 5.

## 2 Preliminaries

In this section, we describe the initial system set up, intractability assumptions and the special notations used in this paper.

### 2.1 Intractability Assumptions

Let $G$ be a non-abelian group and $(x, \alpha) \in G \times G$, such that $x = a\alpha a^{-1}$, for some $a \in G$, then we say that $x$ and $\alpha$ are conjugates of each other. Given $x, y \in G$, the *conjugacy decision problem* (CDP) is to determine whether $x$ and $y$ are conjugates or not and the *conjugacy search problem* (CSP) is to find an $a \in G$ such that $x = aya^{-1}$ if $x$ and $y$ are known to be conjugates.

### 2.2 Initial Setup

A finite non-abelian group $G$ in which both CDP and CSP are hard is chosen. Let $A$ and $B$ be two mutually commuting subgroups of $G$, *i.e.* for $a \in A$ and $b \in B$, we have $ab = ba$. Let $H : \{0,1\}^* \to G$ and $h : G \to \{0,1\}^k$ be collision free hash functions.

### 2.3 Notations

We use the following notations through out this paper.

- By $a \in_r A$, we mean a random choice of an element $a$ from the set $A$.
- By $P \xrightarrow{Q} V$, we mean $P$ sends $Q$ to $V$.

## 3 A Zero-Knowledge Undeniable Signature Scheme

In this section, we describe our zero-knowledge undeniable signature scheme.

### 3.1 Public and Private Keys

The system is set up by the signer (Alice) in the following manner: Alice chooses $\alpha \in_r G$ and $a \in_r A$ and computes $x = a\alpha a^{-1}$. She sets her public key as $(\alpha, x)$ and the private key as $a$. It is assumed that CDP and CSP are hard for the pair $(\alpha, x)$. We shall denote by $PK$, the tuples $(\alpha, x)$ generated as above.

### 3.2 Signature Generation

Suppose that Alice wants to sign a message $m$. She computes $S_m = aya^{-1}$, where $y = H(m)$, giving the output pair $(m, S_m)$. We denote by $SIG(m)$, the set of valid signatures on $m$.

### 3.3 The Confirmation Protocol

Here we present a zero-knowledge confirmation protocol. It is carried out by two players, a prover $(P)$ and a verifier $(V)$. The public input to the protocol are the public key parameters, namely $(\alpha, x) \in PK$ and a pair $(m, \hat{S}_m)$. For the case that $\hat{S}_m$ is a valid signature of $m$, $P$ will be able to convince $V$ of this fact, while if the signature is invalid then no prover will be able to convince $V$ to the contrary except with a negligible probability.

**Signature Confirmation Protocol:**
Input:

Prover: Secret key $a \in A$.

Common: Public key $(\alpha, x) \in PK$, $y$ and alleged $\hat{S}_m$.

1) $V$ chooses $b \in_r B$, computes the challenge $Q = b(\hat{S}_m x)b^{-1}$ and $V \overset{Q}{\to} P$.
2) $P$ chooses $c, d \in_r G$, computes the response $R = d\alpha c(a^{-1}Qa)c^{-1}d^{-1}$, and $P \overset{R}{\to} V$.
3) $V \xrightarrow{b} P$.
4) $P$ verifies that $Q = b(\hat{S}_m x)b^{-1}$ and then $P \xrightarrow{(c,d)} V$.
5) $V$ verifies that $R = d\alpha cb(y\alpha)b^{-1}c^{-1}d^{-1}$. If it holds then $V$ accepts $\hat{S}_m$ as a valid signature of $P$.

### 3.4 Security Analysis of the Confirmation Protocol

**Completeness:** Let $S_m$ be a valid signature of $P$. As $a \in A$ and $b \in B$, we have $a^{-1}b = ba^{-1}$. Hence $P$ can compute,

$$\begin{aligned} R &= d\alpha c(a^{-1}Qa)c^{-1}d^{-1} = d\alpha c(a^{-1}b(S_m x)b^{-1}a)c^{-1}d^{-1} \\ &= d\alpha c(ba^{-1}(aya a^{-1})ab^{-1})c^{-1}d^{-1} \\ &= d\alpha cb(y\alpha)b^{-1}c^{-1}d^{-1}, \end{aligned}$$

which $V$ verifies after getting $(c, d)$ from $P$ and accepts the signature as valid. Hence if $S_m \in SIG(m)$ is a valid signature of $P$ and $P$ follows the *signature confirmation protocol*, then $V$ always accepts $S_m$ as a valid signature of $P$.

**Soundness:** Suppose that the signature does not belong to a cheating prover $P^*$. Since $R$ is committed by $P^*$ to $V$ before knowing $b$, the only strategy left for $P^*$ is to send a $(c, d)$ such that the equation $R = d\alpha cb(y\alpha)b^{-1}c^{-1}d^{-1}$ holds. But this amounts to solving the conjugacy problem for the pair $(R, \alpha cb(y\alpha)b^{-1}c^{-1})$. The other strategy for $P^*$ is to guess the value of $b$ after getting $Q$ from $V$ in Step 1. But this amounts not only to solving the conjugacy search problem for the pair $(Q, \hat{S}_m x)$ but also to distinguish between the different conjugators for the pair $(Q, \hat{S}_m x)$. In general out of the $k$ (say) conjugators for the pair $(Q, \hat{S}_m x)$ from $B$, only $l \leq k$ can give the same value for $R$ in step 5. In general $l$ will be much smaller than $k$. It is infeasible for a cheating prover $P^*$ to distinguish between these different values of $b$ even with infinite computing power. That is, suppose that $P^*$ gets a $b'$ such that the equation $Q = b'(\hat{S}_m x)(b')^{-1}$ holds. Then in Step 2, $P^*$ sends $R' = d\alpha cb'(y\alpha)(b')^{-1}c^{-1}d^{-1}$ to $V$. In Step 5, $V$ verifies whether the equation $d\alpha cb'(y\alpha)(b')^{-1}c^{-1}d^{-1} = d\alpha cb(y\alpha)b^{-1}c^{-1}d^{-1}$ holds or not. But in general this equation may not hold. Hence, a cheating prover $P^*$ even with infinite computing power cannot with probability exceeding $\frac{l}{k}$ provide a valid response to $V$ for an invalid signature. Thus the protocol is sound.

**Zero-knowledgeness:** Consider the probabilistic Turing machine $M$ defined as follows: it chooses random braids $c, d \in G$ using the same drawing as the honest prover $P$ and outputs the instances $(c, d, d\alpha cQc^{-1}d^{-1})$. Then the instances generated by this simulator follows the same probability distribution as the interactive pair $(P, V)$ as $G = \{ca : c \in G\}$. Thus on input of a message and its valid signature, any (possibly cheating) verifier $V^*$ interacting with the prover $P$ does not learn any information aside from the validity of the signature. Hence the protocol is zero-knowledge.

### 3.5 The Denial Protocol

The public input to the protocol are the public key parameters, namely $(\alpha, x) \in PK$ and a pair $(m, \hat{S}_m)$. In the case that $\hat{S}_m$ is not a valid signature of $P$, he will be able to convince $V$ of this fact, while if $\hat{S}_m$ is a valid signature of $P$, he will be not able to convince $V$ that the signature is invalid except with negligible probability.

The public input to the protocol are the public key parameters, namely $(\alpha, x) \in PK$ and a pair $(m, \hat{S}_m)$.

In this protocol, we use a zero-knowledge commitment function called *blob*. $blob(r, t)$ perfectly hides the value of $t$ as long as $r$ is secret and once the value of $r$ is revealed one can open the *blob* and get the value of $t$.

**Signature Denial Protocol:**
Input:

Prover: Secret key $a \in A$.

Common: Public key $(\alpha, x) \in PK$, $y$ and alleged $\hat{S}_m$.

1) $P$ computes $k$ such that $k + 1 = \min\{l : \hat{S}_m^l = ay^la^{-1}\}$, and $P \xrightarrow{k} V$.

2) $V$ chooses $b \in_r B$ and $t \in_r \{1, 2, \ldots, k\}$, computes $Q = (y^tb\alpha b^{-1}, \hat{S}_m^tbxb^{-1}) = (Q_1, Q_2)$ and $V \xrightarrow{Q} P$.

3) $P$ computes $t$ by trial and error using, $Q_2(aQ_1a^{-1})^{-1} = \hat{S}_m^t(aya^{-1})^{-t}$.

   Also, $P$ chooses $r$ randomly and $P \xrightarrow{blob(r,t)} V$.

4) $V \xrightarrow{b} P$.

5) $P$ checks the value of $Q$ using $b$ and then $P \xrightarrow{r} V$.

6) $V$ opens the *blob* using the value of $r$ and checks the value of $t$. If the value of $t$ committed by $P$ is correct, then $V$ accepts that $\hat{S}_m$ is not a valid signature of $P$.

### 3.6 Security Analysis of the Denial Protocol

**Completeness:** Suppose that $\hat{S}_m$ is not the signature of $P$. Hence $\hat{S}_m \neq aya^{-1}$. Also, $\hat{S}_m^t \neq (aya^{-1})^t$. Now, upon receiving $Q$ from $V$, $P$ computes

$$\begin{aligned} Q_2(aQ_1a^{-1})^{-1} &= (\hat{S}_m^tbxb^{-1})((ay^ta^{-1})(ab\alpha b^{-1}a^{-1}))^{-1} \\ &= (\hat{S}_m^tbxb^{-1})((ay^ta^{-1})(bxb^{-1}))^{-1} \\ &= \hat{S}_m^t(aya^{-1})^{-t} \neq Identity. \end{aligned}$$

Since $P$ knows $Q_2(aQ_1a^{-1})^{-1}$, $\hat{S}_m$ and $aya^{-1}$, $P$ can compute the value of $t$ by trial and error. Hence if $\hat{S}_m$ is not a signature of $P$ and $P$ and $V$ follow the protocol, then $V$ always accepts that $\hat{S}_m$ is not a valid signature of $P$.

**Soundness:** Assume that a cheating prover $P^*$ has signed the message. There fore $\hat{S}_m = aya^{-1}$. Also, since $(\hat{S}_m)^t = (aya^{-1})^t$, $P^*$ can not compute the value of $t$ by trial and error in Step 3. Now, since $b$ hides $t$ in the challenge $Q$ and the value committed by the blob cannot be changed, $P^*$'s best strategy is to guess the value of $t$, and there are $k$ choices for $t$. Hence, even with infinite computing power, a cheating prover $P^*$ cannot with probability exceeding $\frac{1}{k}$ provide a valid response for an invalid signature. Hence the protocol is sound.

**Remark 1.** *We have given the denial protocol in a non zero-knowledge fashion to make the protocol simpler. The protocol can be made zero-knowledge by avoiding Step 1 and asking V to guess a k. If this guess is not good (which happens when $\hat{S}_m^t = (aya^{-1})^t$, when $\hat{S}_m \neq (aya^{-1})$ ), then the protocol needs to be repeated with a smaller k. One good choice may be $k = 2$. In this case the protocol needs to be repeated several rounds, say l, so that the probability for a cheating prover $P^*$ to provide a valid response for an invalid signature $(= \frac{1}{2^l})$ is negligible.*

*An interaction in which V sends the correct b is trivially simulated. Any V not supplying an acceptable b only receives a blob and so the type of zero-knowledge depends on the type of blob. By assumption the blob used in Step 3 is a zero-knowledge commitment function which perfectly hides t.*

*Hence, the modified protocol is zero-knowledge, namely, on input of a message and a non valid signature, any (possibly cheating) verifier $V^*$ interacting with the prover P does not learn any information about the secret key of P or his commitment aside from the fact that $\hat{S}_m$ is in fact not a valid signature for the message m.*

## 4 Some Non-Abelian Groups for Implementation

In this section, we suggest some non-abelian groups as possible platforms for the implementation of the above undeniable signature scheme.

### 4.1 Braid Groups

Recently braid groups have been suggested as an alternate platform for doing public-key cryptography. The birthdate of braid group based cryptography can be traced back to the pioneering work of Anshel *et al.* in 1999 [1] and Ko *et al.* in 2000 [12]. Since then, braid groups attracted the attention of many cryptographers due to the fact that, they provide a rich collection of hard problems like the *conjugacy problem, braid decomposition problem* and *root problem* and there are efficient algorithms for parameter generation and group operation [5].

A braid group $B_n$ for $n \geq 2$ is an infinite noncommutative group. Any member of $B_n$ is called an $n$-braid. For each integer $n \geq 2$, the $n$-braid group $B_n$ has the Artin presentation by generators $\sigma_1, \sigma_2, \ldots, \sigma_{n-1}$ with relations:

$$\begin{array}{l} \sigma_i\sigma_j = \sigma_j\sigma_i, \text{ where } |i-j| \geq 2, \text{ and} \\ \sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1}, \text{ for } 1 \leq i \leq n-2. \end{array} \qquad (1)$$

Braids have the following geometric interpretation: an $n$-braid (where $n \in \mathbb{N}$) is a set of disjoint $n$ strands all of which are attached to two horizontal bars at the top and bottom such that each strand always heads downwards as one moves along the strand from top to bottom. Two braids are equivalent if one can be deformed to the other continuously in the set of braids. Then the *Artin generators* $\sigma_i$, corresponds to the crossing of the $i^{th}$ strand under the $(i+1)^{th}$ strand.

Let $LB_n$ and $RB_n$ be two subgroups of $B_n$ consisting of braids obtained by braiding left $\lfloor \frac{n}{2} \rfloor$ strands and right $n - \lfloor \frac{n}{2} \rfloor$ strands, respectively. That is,

$$LB_n = \langle \sigma_1, \ldots, \sigma_{\lfloor \frac{n}{2} \rfloor - 1} \rangle, \text{ and } RB_n = \langle \sigma_{\lfloor \frac{n}{2} \rfloor + 1}, \ldots, \sigma_{n-1} \rangle.$$

Then we have the commutativity property that for any $\alpha \in LB_n$ and $\beta \in RB_n$, $\alpha\beta = \beta\alpha$.

One disadvantage of using braid groups is that $B_n$ is that it an infinite group and does not have any finite non trivial subgroup. Hence for implementation purpose we have to take $G$, $A$ and $B$ as a some finite subsets of $B_n$, $LB_n$ and $RB_n$ respectively. More about the implementation aspects of braid group based cryptographic schemes can be found in [5].

However, many attacks in recent years have considerably reduced the security of braid group based cryptographic protocols. At present it is not very clear whether any trusted cryptographic protocols can be developed on braid groups. For more information about braid groups refer to [3]. Survey on braid cryptography can be found in [8, 13].

### 4.2 Polycyclic Groups

At present polycyclic groups appear as a very promising platform for developing non-abelian cryptography. Polycyclic groups were first suggested for cryptographic applications by B. Eick and D. Kahrobaei [9] in 2004. In the polycyclic groups, the word problem can be solved efficiently, where as the conjugacy problem does not have an efficient solution.

Polycyclic groups are natural generalizations of cyclic groups, but they are much more complex in their structure than cyclic groups. A polycyclic group has a finite presentation of the following form,

$$\begin{aligned} \langle a_1, \ldots, a_n | a_i^{-1}a_ja_i &= w_{ij}, a_ia_ja_i^{-1} \\ &= v_{ij}, r_k^{-1}a_kr_k = u_{kk} \\ &\quad \text{for } 1 \leq i < j \leq n \text{ and } k \in I \rangle, \end{aligned}$$

where $I \subset \{1, \ldots, n\}$ and $r_i \in \mathbb{N}$ if $i \in I$ and the right sides $w_{ij}, v_{ij}, u_{ij}$ of the relations are words in the generators $a_{j+1}, \ldots, a_n$. More about polycyclic groups can be found in [18].

There are both finite and infinite polycyclic groups. One can work with either finite polycyclic groups or finite subgroups of polycyclic groups for cryptographic applications.

### 4.3 Thompson's Group

Thompson's group is well known in many areas of mathematics like algebra, geometry and analysis. Thompson's group was first suggested for cryptographic applications by V. Shpilrain and A. Ushakov [9] in 2005. It has the attractive feature that the word problem is solvable in almost linear time. Thompson's group is an infinite non-abelian group having the following infinite presentation,

$$\langle x_0, x_1, x_2, \ldots | x_i^{-1}x_kx_i = x_{k+1}, \text{ where } k > i \rangle.$$

More about Thompson's groups can be found in [4].

### 4.4 Other Groups

There are many other non-abelian groups which are promising candidates for developing public-key cryptography like matrix groups over finite commutative rings suggested by D. Grigoriev, I. Ponomarenko in 2005 [11] and finitely presented non-abelian nilpotent group of class 2 suggested by A. Mahalanobis in March, 2006 [14].

## 5 Concluding Remarks

In this paper, we described a zero-knowledge undeniable signature scheme in the frame work of a general non-abelian group. In Section 5, we suggested some potential non-abelian groups in which the above digital signature scheme can be implemented. The security of our scheme is depending upon the conjugacy problem in non-abelian groups. It is worth reformulating these protocols by employing other hard problems in non-abelian groups like the decomposition problem.

There are many desirable features for a good undeniable signatures like *convertibility* (the possibility to transform undeniable signatures into regular ones), *delegation* (enabling selected third parties to confirm/deny signatures but not to sign). We have not considered these problems in this paper. It is worth constructing protocols for these cases in the non-abelian group settings.

## Acknowledgements

Research supported by CSIR, New Delhi, under award number: 2-31/97(11)-E.U.II

# References

[1] I. Anshel, M. Anshel, and D. Goldfeld, “An algebraic method for public-key cryptography,” *Mathematical Research Letters*, vol. 6, pp. 287-291, 1999.

[2] I. Anshel, M. Anshel, B. Fisher, and D. Goldfeld, “New key agreement protocols in braid group cryptography,” in *Cryptology- CT-RSA'01*, LNCS 2020, pp. 144-1561, Springer-Verlag, 2001.

[3] J. S. Birman, *Braids, Links and Mapping Class Groups*, Annals of Math. Study 82, Princeton University Press, 1974.

[4] J. W. Cannon, W. J. Floyd, and W. R. Parry,“Introductory notes on Richard Thompson's groups,”*L'Enseignement Mathematique*, vol. 2, no. 42, pp. 215-256, 1996

[5] J. C. Cha, K. H. Ko, S. J. Lee, J. W. Han, and J. H. Cheon, “An efficient implementation of braid groups,” in *Cryptology: Proceedings of Asiacrypt'01*, LNCS 2248, pp. 144-156, Springer-Verlag, 2001.

[6] D. Chaum, “Zero-knowledge undeniable signatures,” in *Cryptology: Proceedings of Eurocrypt'90*, LNCS 473, pp. 458-464, Springer-Verlag, 1990.

[7] D. Chaum and H. V. Antwerpen, “Undeniable signatures,” *Advances in Cryptology: Proceedings of Crypto'89*, LNCS 435, Springer-Verlag, pp. 212-217, 1990.

[8] P. Dehornoy, “Braid-based cryptography,” *Contemporary Mathematics*, vol. 360, pp. 5-33, 2004.

[9] B. Eick and D. Kahrobaei, *Polycyclic Groups: A New Platform for Cryptology.* (http://www-public.tu-bs.de:8080/ beick/publ/crypto.ps)

[10] R. Gennaro, H. Krawczyk, and T. Rabin, “RSA-based undeniable signatures,” in *Cryptology: Proceedings of Crypto'97*, LNCS 1294, pp. 132-149, Springer-Verlag, 1997.

[11] D. Grigoriev and I. Ponomarenko, *Constructions in Public-Key Cryptography Over Matrix Groups.* (http://arxiv.org/abs/math/0506180)

[12] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. S. Kang, and C. S. Park, “New public-key cryptosystem using braid groups,” in *Cryptology: Proceedings of Crypto'00*, LNCS 1880, pp. 166-183, Springer-Verlag, 2000.

[13] E. Lee, “Braid groups in cryptology,”*IEICE Tarnsactions on Fundamentals*, vol. E87-A, no. 5, pp. 986-992, 2004.

[14] A. Mahalanobis, *Diffie-Hellman Key Exchange Protocol and Non-Abelian Nilpotent Groups.* (http://arxiv.org/abs/math.GR/0602282)

[15] S. H. Paeng, K. C. Ha, J. H. Kim, S. Chee, and C. Park, “New public key cryptosystem using finite non abelian groups,” *advances in Cryptology: Proceedings of Crypto'01*, LNCS 2139, pp. 470-485, Springer-Verlag, 2001.

[16] V. Shpilrain and A. Ushakov, *Thompson's Group and Public Key Cryptography.* (http://arxiv.org/abs/ math.GR/0505487)

[17] H. Sibert, P. Dehornoy, and M. Girault, *Entity Authentication Schemes Using Braid Word Reduction*, 2002. (http://eprint.iacr.org/2002/187)

[18] C. C. Sims, *Computations with Finitely Presented Groups*, Encyclopedia of Mathematics and its Applications, vol. 48, Cambridge University Press, 1994.

[19] R. Steinwandt, *Non-Abelian Groups in Public Key Cryptography*, 2004. (http://www.cms.math.ca/ Events/winter04/abs/Ple n.html)

[20] E. Stickel, “A new public-key cryptosystem in non abelian groups,” in *Proceedings of the Thirteenth International Conference on Information Systems Development*, pp. 70-80, Vilnius Technika, 2004.

[21] T. Thomas, *On Public-Key Cryptography Using Hard Problems in Braid Groups*, Ph.D Thesis, Indian Institute of Technology Kanpur, Kanpur, India, Sep. 2005.

**Tony Thomas** completed his Masters and Ph.D from the Indian Institute of Technology Kanpur, India. He will be joining as a Postdoctoral Researcher at KAIST, Daejeon, South Korea in August 2006. His research interests include Cryptography and Network Security.

**Arbind Kumar Lal** completed his Masters in 1988 and Ph.D. in 1993 from the Indian Statistical Institute, Delhi Centre. He is presently working as an Associate Professor at Indian Institute of Technology Kanpur, India. His research interests include applications of Linear Algebra to Combinatorics, Coding Theory, Cryptography and Graph Theory.