Galois geometries contributing to cryptography

Leo Storme

Ghent University
Dept. of Mathematics
Krijgslaan 281 - S22
9000 Ghent
Belgium

Opatija, 2010

Outline

Outline

Cryptography

1. Transmit information in confidential way,
2. Split secret into shares,
3. Authentication.

OUTLINE

Secret sharing scheme

1. **Secret sharing scheme**: cryptographic equivalent of vault that needs several keys to be opened.
2. Secret S divided into *shares*.
3. *Authorised sets*: have access to secret S by putting their shares together.
4. *Unauthorised sets*: have no access to secret S by putting their shares together.

(n, k)-THRESHOLD SCHEME

1. n participants.
2. Each group of k participants can reconstruct secret S, but less than k participants have no way to learn anything about secret S.

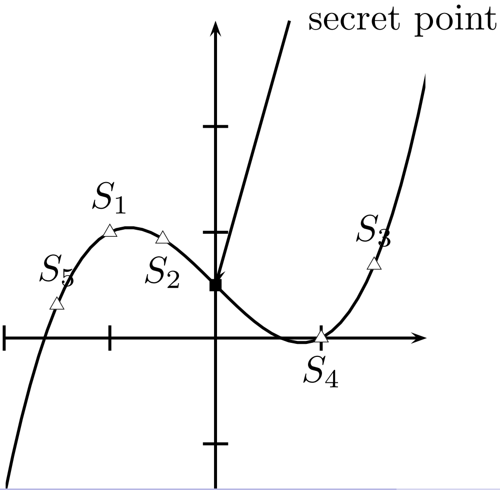
Shamir's k-out-of-n secret sharing scheme

1. $\mathbb{F}_q$ = finite field of order q.
2. Dealer chooses polynomial $f(X) = f_0 + f_1 X + \cdots + f_{k-1} X^{k-1} \in \mathbb{F}_q[X]$, and,
3. gives participant number i, point $(x_i, f(x_i))$ on graph of f $(x_i \neq 0)$.
4. Value $f(0) = f_0$ is secret S.

Shamir's k-out-of-n secret sharing scheme

1. Set of k participants can reconstruct $f(X) = f_0 + f_1 X + \cdots + f_{k-1} X^{k-1}$ by interpolating their shares $(x_i, f(x_i))$. Then they can compute secret $f(0)$.
2. If $k' < k$ persons try to reconstruct secret, for every $y \in \mathbb{F}_q$, there are exactly $|\mathbb{F}_q|^{k-k'-1}$ polynomials of degree at most $k-1$ which pass through their shares and the point $(0, y)$. Thus they gain no information about $f(0)$.

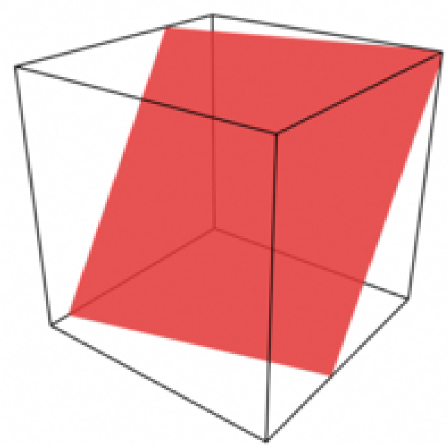
REALISATION OF SHAMIR'S k-OUT-OF-n SECRET SHARING SCHEME



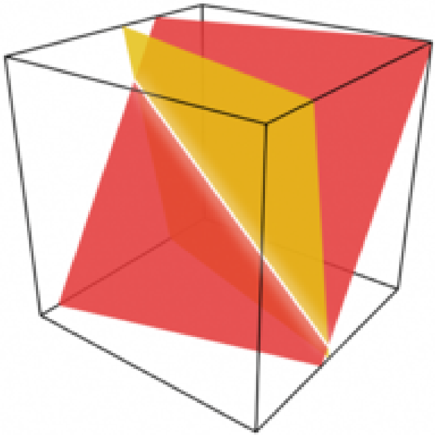
Geometrical realisation of Shamir's k-out-of-n secret sharing scheme (Blakley)

1. Secret S = point of $\mathrm{PG}(3, q)$.
2. Shares = planes of $\mathrm{PG}(3, q)$ such that exactly three of them only intersect in S.

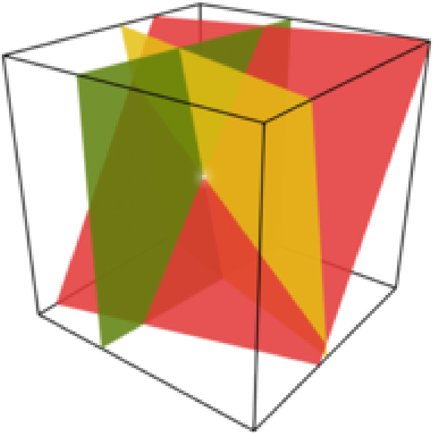
Geometrical realisation of Shamir's k-out-of-n secret sharing scheme (Blakley)



Geometrical realisation of Shamir's k-out-of-n secret sharing scheme



Geometrical realisation of Shamir's k-out-of-n secret sharing scheme



Coding-theoretical realisation of Shamir's k-out-of-n secret sharing scheme

(McEliece and Sarwate)

1. $C : [n+1, k, n-k+2]_q$ MDS code.
2. For secret $c_0 \in \mathbb{F}_q$, dealer creates codeword $c = (c_0, c_1, \ldots, c_n) \in C$. Share of participant number i is symbol c_i.
3. Since C is MDS code with minimum distance $n-k+2$, codeword c can be uniquely reconstructed if only k symbols are known.
4. So any set of k persons can compute secret c_0.
5. On the other hand, less than k persons do not learn anything about secret, since for any possible secret c', the same number of codewords that fit to secret c' and their shares exist.

More general secret sharing scheme

Definition

Support of $c = (c_1, \ldots, c_n) \in \mathbb{F}_q^n$:

$$\sup(c) = \{i \mid c_i \neq 0\}.$$

Let C be linear code. Nonzero codeword $c \in C$ is called *minimal* if

$$\forall c' \in C : \sup(c') \subseteq \sup(c) \implies c' \in \langle c \rangle \, .$$

More general secret sharing scheme

Lemma (Massey)

Let C be an $[n+1, k]_q$-code. Secret sharing scheme is constructed from C by choosing codeword $c = (c_0, \ldots, c_n)$. Secret is c_0 and shares of participants are coordinates c_i $(1 \leq i \leq n)$.
Minimal authorized sets of secret sharing scheme correspond to minimal codewords of $C^\perp$ with 0 in their supports.

MORE GENERAL SECRET SHARING SCHEME

Proof: Suppose set $\{1, \ldots, k\}$ is authorised set. This means that c_0 can be determined from $c_1, \ldots, c_k$, i.e. there exist constants $a_1, \ldots, a_k$, with

$$c_0 = a_1 c_1 + \cdots + a_k c_k, \tag{1}$$

which means that $(1, -a_1, \ldots, -a_k, 0, \ldots, 0)$ is codeword of $C^\perp$ with 0 in its support. $\square$

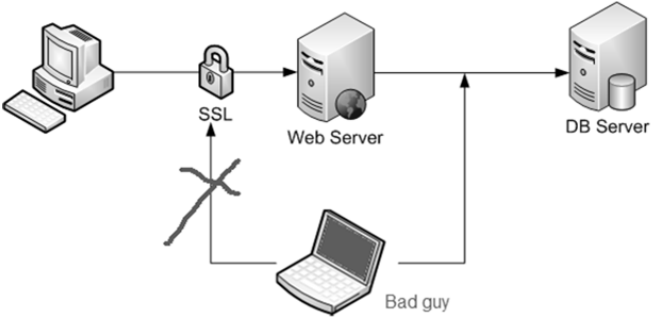
Outline

1. Cryptography
2. Secret sharing scheme
3. Message Authentication code (MAC)
4. Linear MDS code in AES

Problem of authentication

1. Problem: Alice wants to send Bob a message m.
2. Attacker intercepts m and sends alternated message m' to Bob.

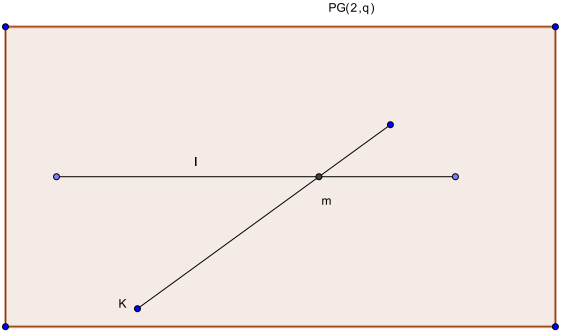
Problem of authentication



How can Bob be sure that message he gets is correct?
Introduce *authentication*!

Example of message authentication code

1. l = line of $\mathrm{PG}(2, q)$.
2. Message m = point of l.
3. Authentication key K = point in $\mathrm{PG}(2, q) \setminus l$.
4. Authentication tag = line through message m and key K.


PG(2,q)
l
m
K

Example of authentication code

1. If attacker wants to create message (m, K) without knowing key K, he must guess an affine line through m. There are q possibilities, i.e. the chance for correct attack is $\frac{1}{q}$.
2. If attacker already knows authenticated message (m, K), he knows that key K must lie on the line mK. But for every of q affine points on line mK, there exists line through m. So he cannot do better than guess the key which gives probability of $\frac{1}{q}$ for successful attack.

Security of authentication code

1. p_i = probability of attacker to construct pair (m, K) without knowledge of key K, if he only knows i different pairs (m_j, K_j).
2. Smallest value r for which $p_{r+1} = 1$ is called *order* of authentication code.
3. For $r = 1$, p_0 = probability of *impersonation attack* and probability p_1 = probability of *substitution attack*.

Theorem

If MAC has attack probabilities $p_i = 1/n_i$ $(0 \leq i \leq r)$, then $|\mathcal{K}| \geq n_0 \cdots n_r$.

MAC that satisfies this theorem with equality is called *perfect*.

Geometrical construction of perfect MAC

Definition

Generalised dual arc $\mathcal{D}$ of order l with dimensions $d_1 > d_2 > \cdots > d_{l+1}$ of $\mathrm{PG}(n, q)$ is set of subspaces of dimension d_1 such that:

1. each j subspaces intersect in subspace of dimension d_j, $1 \leq j \leq l+1$,
2. each $l+2$ subspaces have no common intersection.

$(n, d_1, \ldots, d_{l+1})$ = *parameters* of dual arc.

Generalised dual arcs

Theorem

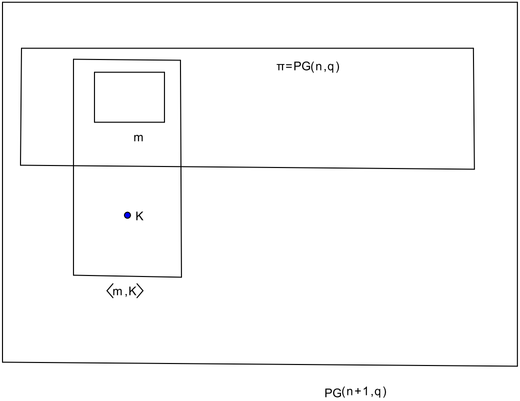
There exists generalised dual arc in $PG(\binom{n+d+1}{d+1} - 1, q)$*, with dimensions* $d_i = \binom{n+d+1-i}{d+1-i} - 1, i = 0, \ldots, d+1.$

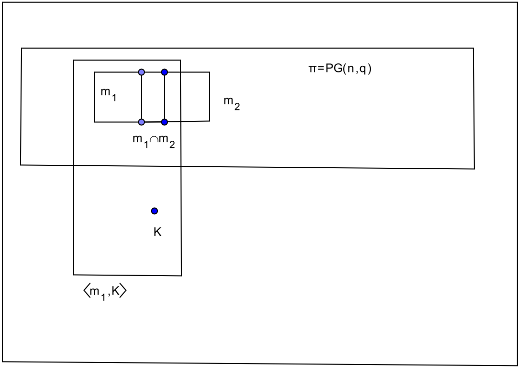
1. Spaces have dimension $d_1 = \binom{n+d}{d} - 1$.
2. Two spaces intersect in space of dimension $d_2 = \binom{n+d-1}{d-1} - 1$.
3. Three spaces intersect in space of dimension $d_3 = \binom{n+d-2}{d-2} - 1$.
4. $\ldots$

Link between MAC and generalised dual arc

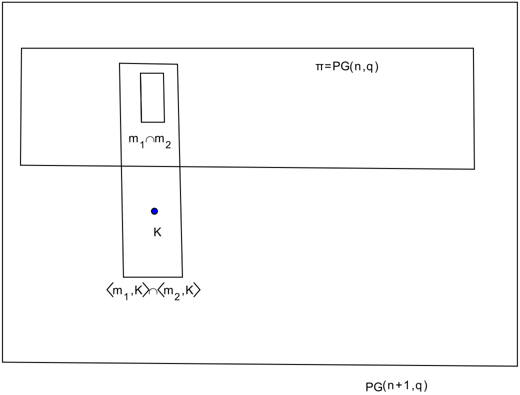
1. π = hyperplane of PG$(n+1, q)$ and $\mathcal{D}$ = generalised dual arc of order l in π with parameters $(n, d_1, \ldots, d_{l+1})$.
2. message m = element of $\mathcal{D}$.
3. key K = point of $PG(n+1, q)$ not in π.
4. Authentication tag that belongs to message m and key K is generated $(d_1 + 1)$-dimensional subspace.
5. Perfect MAC of order $r = l + 1$ with attack probabilities

$$p_i = q^{d_{i+1} - d_i}.$$


π=PG(n,q)
m
K
⟨m,K⟩
PG(n+1,q)



PG(n+1,q)


π=PG(n,q)
$m_1 \cap m_2$
K
$\langle m_1, K\rangle \cap \langle m_2, K\rangle$
PG(n+1,q)

OUTLINE

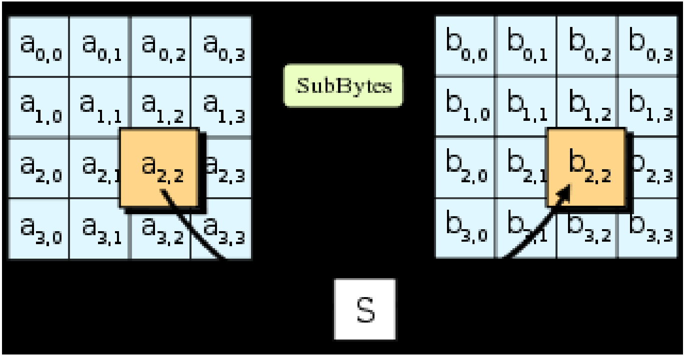
Advanced Encryption Standard (AES)

1. In 1997, American National Institute of Standards and Technology started competition to design a successor for Data Encryption Standard (DES).
2. In 2000, proposal of J. Daemen and V. Rijmen was selected as new Advanced Encryption Standard (AES).

Short description of AES

1. Clear text: 4×4 matrix over $\mathbb{F}_{256}$.
2. 10 rounds of *SubBytes*, *ShiftRows*, *MixColumns*, and *AddRoundKey*.

SubBytes

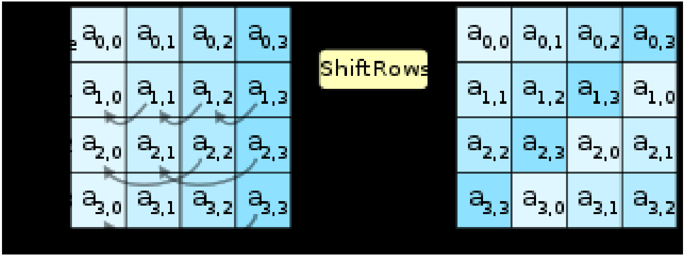

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
SubBytes
S
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$

SubBytes

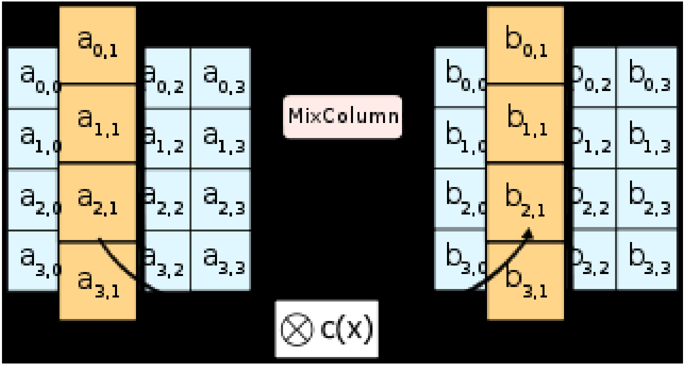
1. First
$$\mathbb{F}_{256} \longrightarrow \mathbb{F}_{256} : x \mapsto x^{-1},$$
($x = 0$ is mapped onto itself).
2. Secondly (represent $x \in \mathbb{F}_{256}$ by its 8 bits in additive notation)

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

SHIFTROWS


$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
ShiftRows
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,1}$ $a_{1,2}$ $a_{1,3}$ $a_{1,0}$
$a_{2,2}$ $a_{2,3}$ $a_{2,0}$ $a_{2,1}$
$a_{3,3}$ $a_{3,0}$ $a_{3,1}$ $a_{3,2}$

MixColumns


$a_{0,1}$
$a_{0,0}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
MixColumn
$\otimes c(x)$
$b_{0,1}$
$b_{0,0}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$

Diffusion in cryptography

1. Small change in clear text must imply large change for cipher text.
2. Small change in cipher text must arise from large change in clear text.

Question: how to realize diffusion?

MixColumns

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} \alpha & \alpha+1 & 1 & 1 \\ 1 & \alpha & \alpha+1 & 1 \\ 1 & 1 & \alpha & \alpha+1 \\ \alpha+1 & 1 & 1 & \alpha \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix},$$

where $\alpha^8 + \alpha^4 + \alpha^3 + \alpha + 1 = 0$.

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \alpha & \alpha+1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & \alpha & \alpha+1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & \alpha & \alpha+1 \\ 0 & 0 & 0 & 1 & \alpha+1 & 1 & 1 & \alpha \end{pmatrix}$$

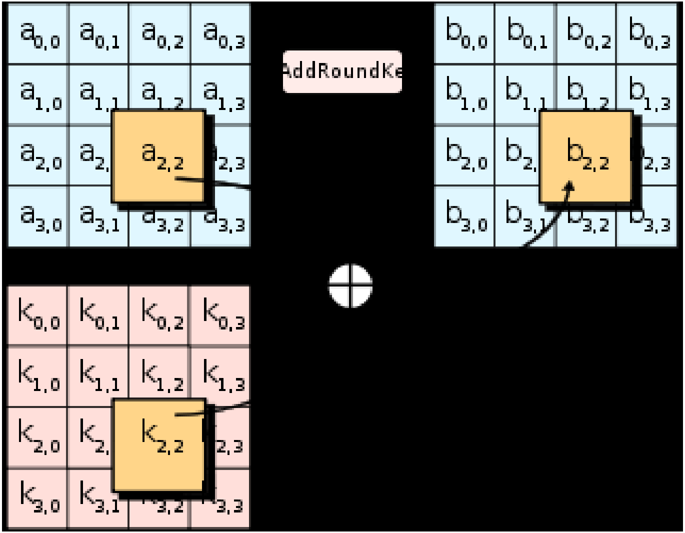
is generator matrix of an $[8,4,5]$-MDS code over $\mathbb{F}_{256}$.

DIFFUSION IN AES

Bytes changed in input	Bytes changed in output
1	4
2	≥ 3
3	≥ 2
4	≥ 1

Bytes changed in output	Bytes changed in input
1	4
2	≥ 3
3	≥ 2
4	≥ 1

AddRoundKey



Thank you very much for your attention!