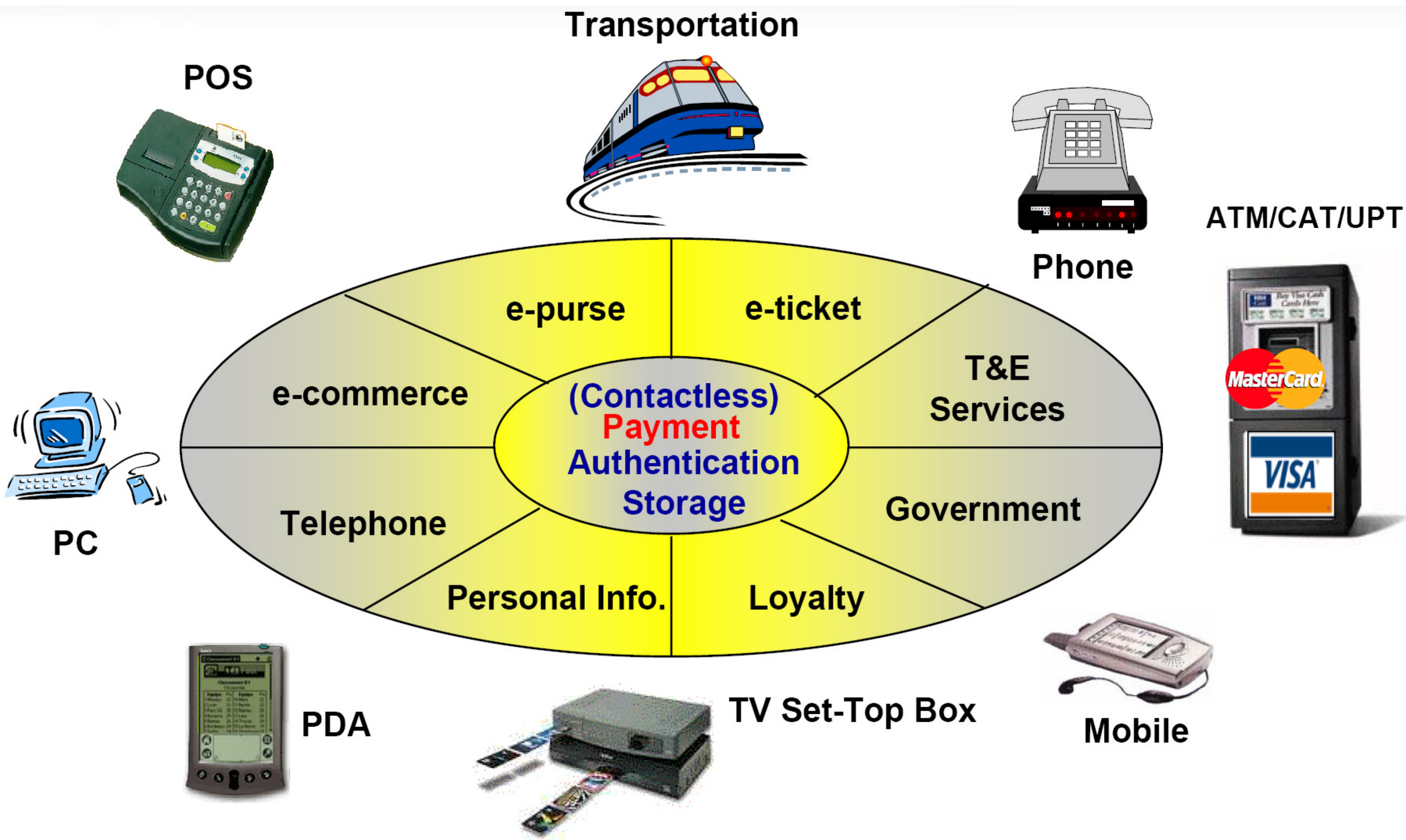
# Cryptology, homomorphisms and graph theory

Rogla, May 2013

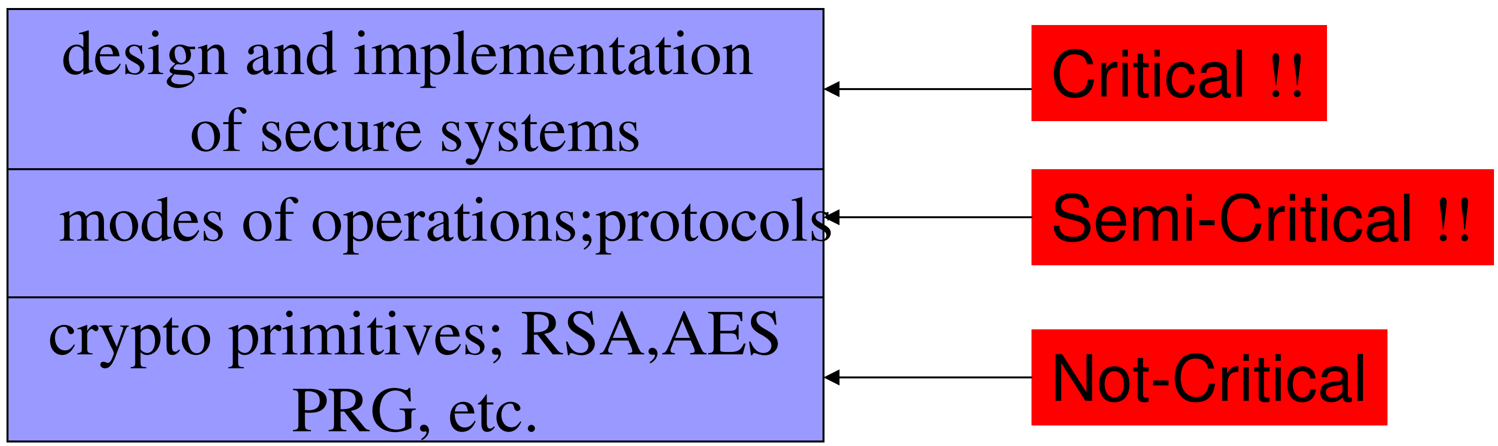
Enes Pasalic

# Applications of cryptography

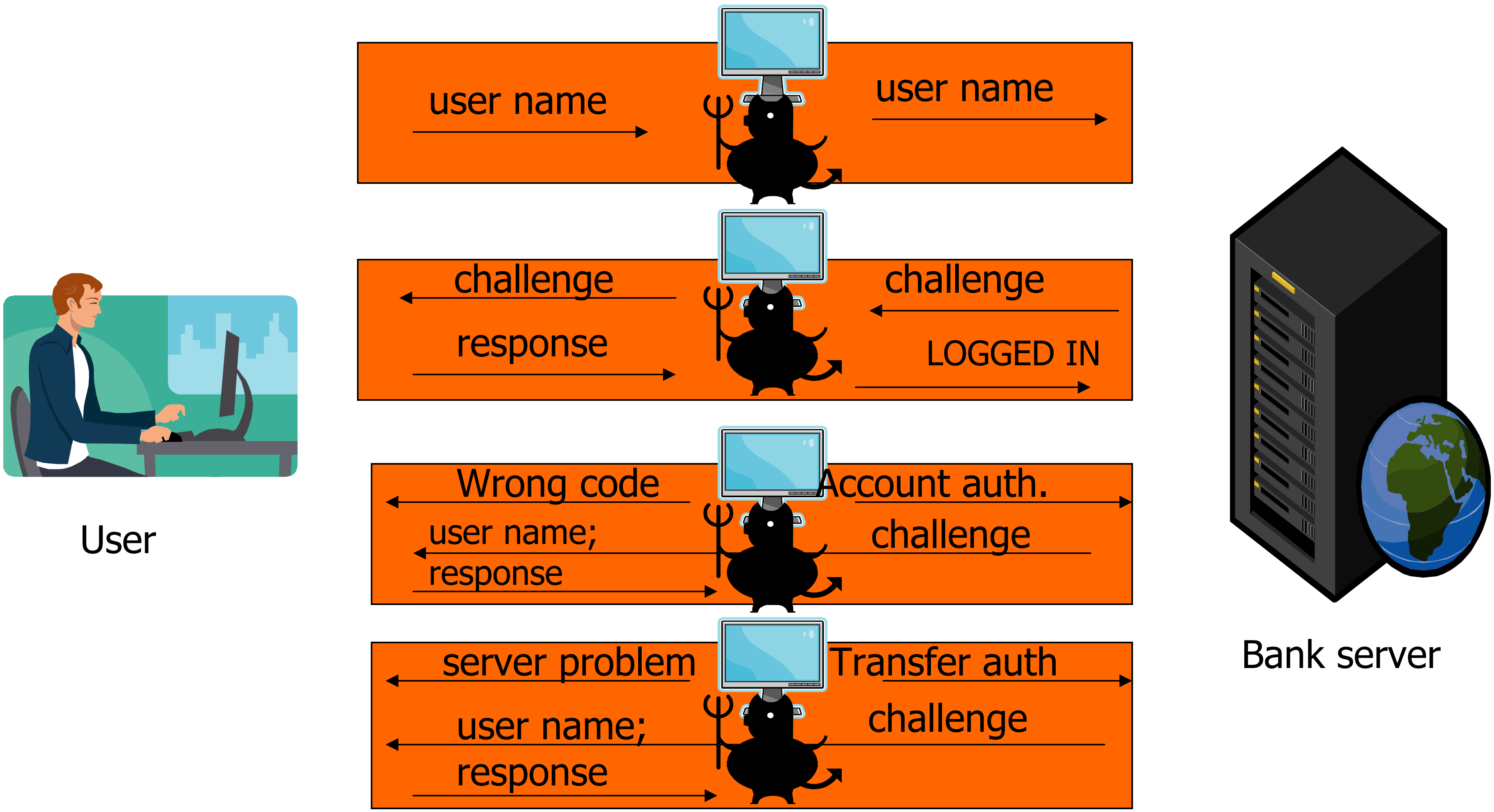


# Cryptography in a nutshell

- Talking about cryptography – not hacking !!



# Missusing protocols



# Why standard primitives are secure ?

- Because thousands of academics are designing and cryptanalyzing these primitives

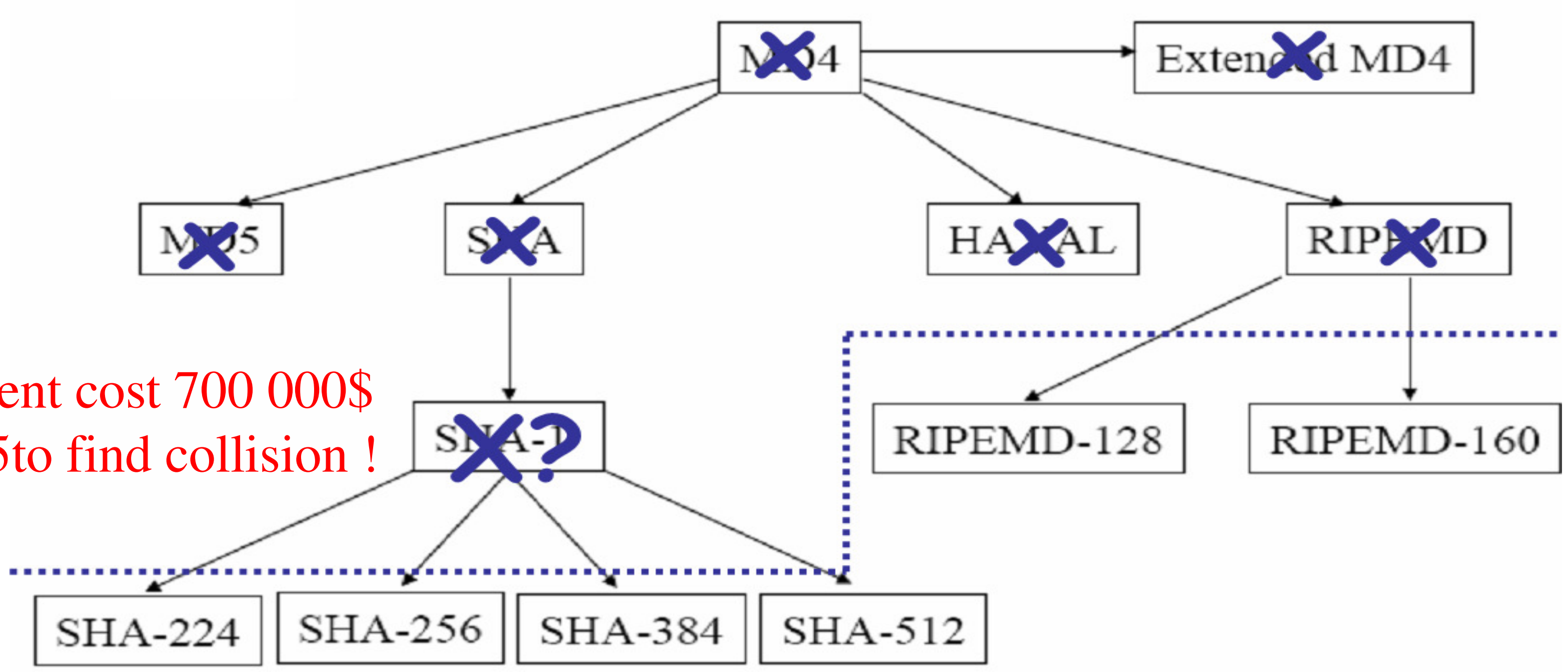
- Do you really care when using public key crypto based on :
  - **Factoring problem** – RSA
  - **Discrete log problem** – ElGammal . . .

  - or using finite **nonabelian (e.g. Braid)** groups, based on solving equations in **noncommutative** groups, **polycyclic** groups ...

- As long as the primitive has undergone public scrutiny you are doing fine

# BLAKE hash function YES or NOT ?

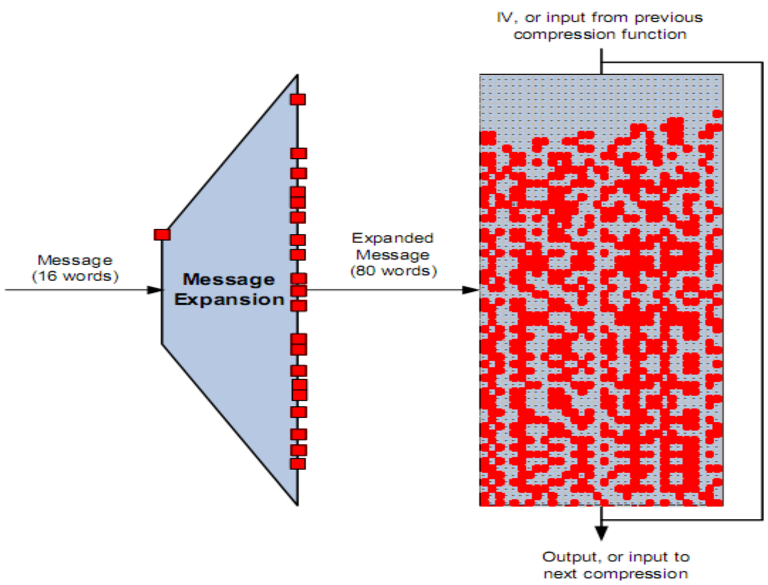
- **BLAKE** entered the final phase of NIST competition (5 left) – probably a hash standard




equipment cost 700 000$
in 2015to find collision !

# BLAKE is secure even though ...

- Janoš Vidali, Peter Nose, Enes Pasalic. *Collisions for variants of the BLAKE hash function*, *IPL, 2010*
- Attacks on simplified version, **BLAKE not compromized** !




Flipping a single bit causes c.a. half of bits to change, etc.

# Loose ”Guidelines” - secure implementation

- Use well-analyzed primitives, AES, RSA, SHA - xx, unless you come from military (black box scenario :)
- Update your primitives, check if still using MD5 ☺ (even SHA-1 will need an update soon)
- Implement all the steps of protocols (try not to speed up algorithm by cheating !)
- How do you generate the keys ? Where do you store them ?
- Open source usage ? IV vector is reset to 0 when you lose elektricity ?

# Copyright,  PKC, homomorphic encryption ...

- Imagine that all encryption algorithms are **copyrighted**, I would be doing fine how about you ?
- Only possibility seems to be **pattent applications** (possibly on stand-alone basis or with some support) ...
- **Cloud computing and homomorphic encryption** seem to be very hot topic, though probably not for ARRS
- + 30 year open problem to embed **fully homomorphic encryption** scheme

# One-way functions

A **one-way function** $f : X \to Y$ has the properties that

- it is computationally "easy" to compute $f(x)$ for any $x \in X$.
- it is computationally "difficult" to invert $f$, i.e. given $y \in Y$, to find an $x \in X$ such that $f(x) = y$.

Of course, this is vague and needs to be more precisely defined, but the idea is to use such an $f$ as encryption function.

This makes life difficult for the Adversary, (GOOD)
but also for the intended receiver! (BAD)

# Trapdoor one-way function

A **trapdoor one-way function** is a one-way function $f$ with the further property that if you know some secret extra information, inverting $f$ becomes “easy”.

Refined idea: For encryption, we use a trapdoor one-way function for which only the receiver knows the secret (the trapdoor).

We need not only one trapdoor one-way function $E : \mathcal{M} \to \mathcal{C}$ but a whole family of such functions, indexed by keys.

- The public key cryptography realizes these ideas. Based on some **old number theoretical problems**.

# RSA – Public key cryptosystem

Key generation:

- Generate two large primes p and q of at least 512 bits.
- Compute $N = p \cdot q$ and $\phi(N) = (p-1)(q-1)$.
- Select a random integer $e$, $1 < e < \phi(N)$, such that

$$\gcd(e, (p-1)(q-1)) = 1.$$

- Using the XGCD compute the unique integer $d$, $1 < d < \phi(N)$ with

$$e \cdot d \equiv 1 \pmod{\phi(N)}.$$

Public key = $(N, e)$ which can be published.
Private key = $(d, p, q)$ which needs to be kept secret.

# RSA encryption/decryption

**RSA key setup**

Alice chooses secret primes $p$ and $q$, computes $N = pq$ and chooses an $e$ such that $\gcd(e, \Phi(N)) = 1$. She then computes $d = 1/e$ in $\mathbb{Z}^*_{\Phi(N)}$. Her public key is $(N, e)$ and her private key is $d$.

**RSA encryption**

Bob wants to encrypt $m \in \mathbb{Z}^*_N$ for Alice. He computes $C = m^e \bmod N$.

**RSA decryption**

Alice computes $m = C^d \bmod N$.

# Decryption - proof

Assume that $m \in \mathbb{Z}_N^*$. Alice computes

$$C^d \bmod N = m^{ed} \bmod N = m^{1+k\cdot\Phi(N)} = (m^{\Phi(N)})^k \cdot m = 1^k \cdot m = m.$$

What if $m \notin \mathbb{Z}_N^*$?

- This means that $m$ is a multiple of $p$ or $q$, a very unlikely case that can be ignored in practice.
- The equality $m^{ed} \bmod N = m$ holds also in this case, but requires another proof, based on the Chinese Remainder

# Proving that decryption works

- We have to show that $m^{ed}=m$. Recall that $\boxed{ed = 1 + k\,\phi(n) = 1 + k(p-1)(q-1)}$
  - If $\gcd(m,p)=1$:
    - By Fermat's Little Theorem we have $m^{p-1} \equiv 1 \pmod{p}$.
    - Taking $k(q-1)$-th power and multiplying with $m$ yields

$$m^{1+k(p-1)(q-1)} \equiv m \pmod{p} \qquad (*)$$

  - If $\gcd(m,p)=p$, then $m \equiv 0 \pmod{p}$ and $(*)$ is valid again.

Hence, in all cases $m^{e \cdot d} \equiv m \pmod{p}$ and by a similar argument we have $m^{e \cdot d} \equiv m \pmod{q}$.

Since $p$ and $q$ are distinct primes, the CRT leads to

$$c^d = (m^e)^d = m^{ed} = m^{k(p-1)(q-1)+1} = m \pmod{N}.$$

# Homomorphic property of RSA (multiplicative)

Essentially RSA is malleable owing to the homomorphic property.

Given the encryption of $m_1$ and $m_2$ we can determine the encryption $c_3$ of $m_1 \cdot m_2$.

Let $c_1 = m_1{}^e \pmod N$ and $c_2 = m_2{}^e \pmod N$

$$c_3 = c_1 \cdot c_2 = m_1{}^e \cdot m_2{}^e = (m_1 \cdot m_2)^e \pmod N.$$

We did this without knowing $m_1$ or $m_2$.

**Research problem:** To increase speed of encryption/decryption **binary weight of *e* and *d* should be small.** Can we derive a lower bound on wt( $e$ ) + wt( $d$ ) !

# Pallier E-voting – additive homomorphism

- Suppose Alice, Bob and Oscar are running in an election. **Only 6 people voted in the election.**

00 00 **01** = 1

00 **01** 00 = 4

00 **01** 00 = 4

00 00 **01** = 1

**01** 00 00 = 16

00 00 **01** = 1

| Vote | Oscar | Bob | Alice |
|---|---|---|---|
| 1 | | | ● |
| 2 | | ● | |
| 3 | | ● | |
| 4 | | | ● |
| 5 | ● | | |
| 6 | | | ● |

# Short mathematical description

- Decisional composite residuosity assumption
  - Given composite $n$ and integer $z$, it is hard to determine if $y$ exists such that

$$z \equiv y^n \pmod{n^2}$$

**Definition**

Pick two large primes $p$ and $q$ and let $n = pq$. Let $\lambda$ denote the Carmichael function, that is, $\lambda(n) = \text{lcm}(p-1, q-1)$. Pick random $g \in \mathbb{Z}^*_{n^2}$ such that $L(g^\lambda \bmod n^2)$ is invertible modulo $n$ (where $L(u) = \frac{u-1}{n}$). $n$ and $g$ are public; $p$ and $q$ (or $\lambda$) are private. For plaintext $x$ and resulting ciphertext $y$, select a random $r \in \mathbb{Z}^*_n$. Then,

$$e_K(x, r) = g^x r^n \bmod n^2$$

$$d_K(y) = \frac{L(y^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n$$

# Pallier voting - counting

- Let $p = 5$ and $q = 7$. Then $n = 35$, $n^2 = 1225$. $g$ is chosen to be 141 (so that $n \mid \mathrm{ord}(g)$ ). For the first vote x1 = 1, $r$ is **randomly chosen as 4**.
- Then,

$$\boxed{\mathbf{e_K\,(x1,r1) = e_K\,(1,\,4) = g^{x1} * r1^{n} = 141^{1} * 4^{35} = 359 \bmod 1225}}$$

| x1 | $r$ | $e_K\,(x1,r)$ |
|---|---|---|
| 1 | 4 | **359** |
| 4 | 17 | 173 |
| 4 | 26 | 486 |
| 1 | 12 | 1088 |
| 16 | 11 | 541 |
| 1 | 32 | 163 |

# Encryption/decryption

In order to sum the votes, we *multiply* the encrypted data modulo $n^2$:

$$359 \cdot 173 \cdot 486 \cdot 1088 \cdot 541 \cdot 163 \bmod 1225 = 983$$

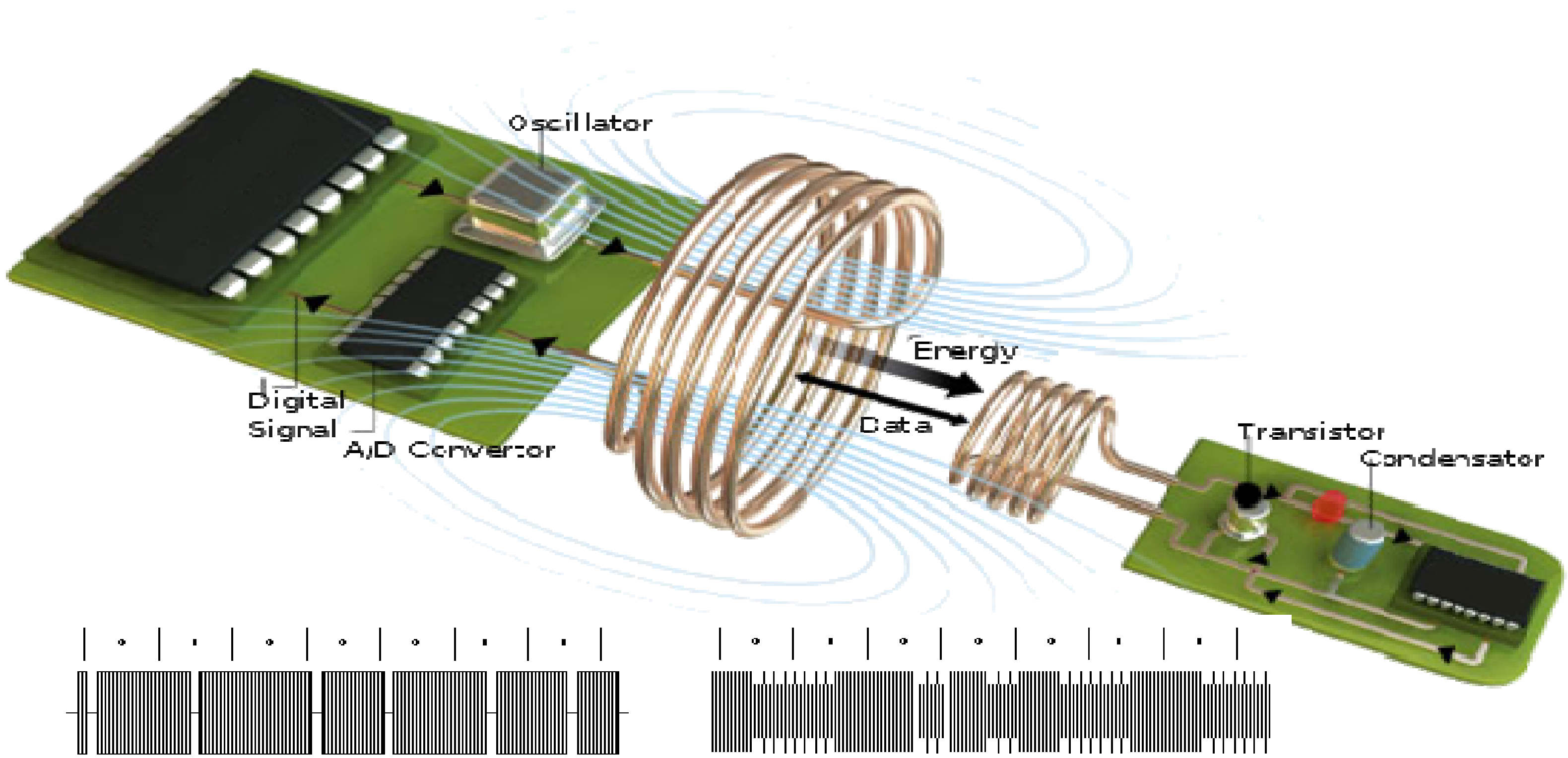
We then decrypt:

$$
\begin{aligned}
L(y^\lambda \bmod n^2) &= L(983^{12} \bmod 1225) = \frac{36-1}{35} = 1 \\
L(g^\lambda \bmod n^2) &= L(141^{12} \bmod 1225) = \frac{456-1}{35} = 13 \\
d_K(y) &= \left(L(y^\lambda \bmod n^2)\right)\left(L(g^\lambda \bmod n^2)\right)^{-1} \bmod n \\
&= 1 \cdot 13^{-1} \bmod 35 \\
&= 27
\end{aligned}
$$

We convert 27 to (01 02 03) for the final results.

# Cryptography and graph theory (a few words)

# RFID Technology

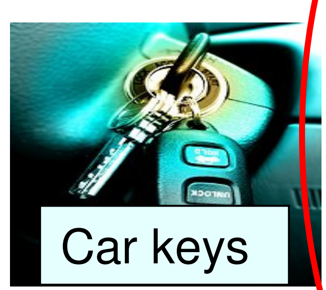



Reader to tag signal
• Dropping field
• Modified Miller Encoding

Tag to reader signal
• Modulating field
• Manchester Encoding

# RFID Applications

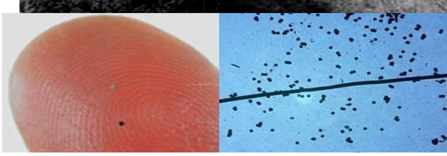
Identify friend or foe (1942)



Car keys


Public transport ticketing


Electronic passport

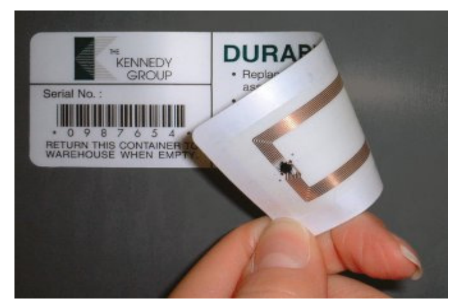


RFID Powder

Access control

Anti-theft


Event ticketing




Supply chain management

# MIFARE

**MIFARE product family from NXP**

- Ultralight
- Classic or Standard (320B, 1KB and 4KB)
- DESFire
- SmartMX

**MIFARE dominance**

- Over 1 billion MIFARE cards sold
- Over 200 million MIFARE Classic cards in use covering 85% of the contactless smart card market

# MIFARE Classic

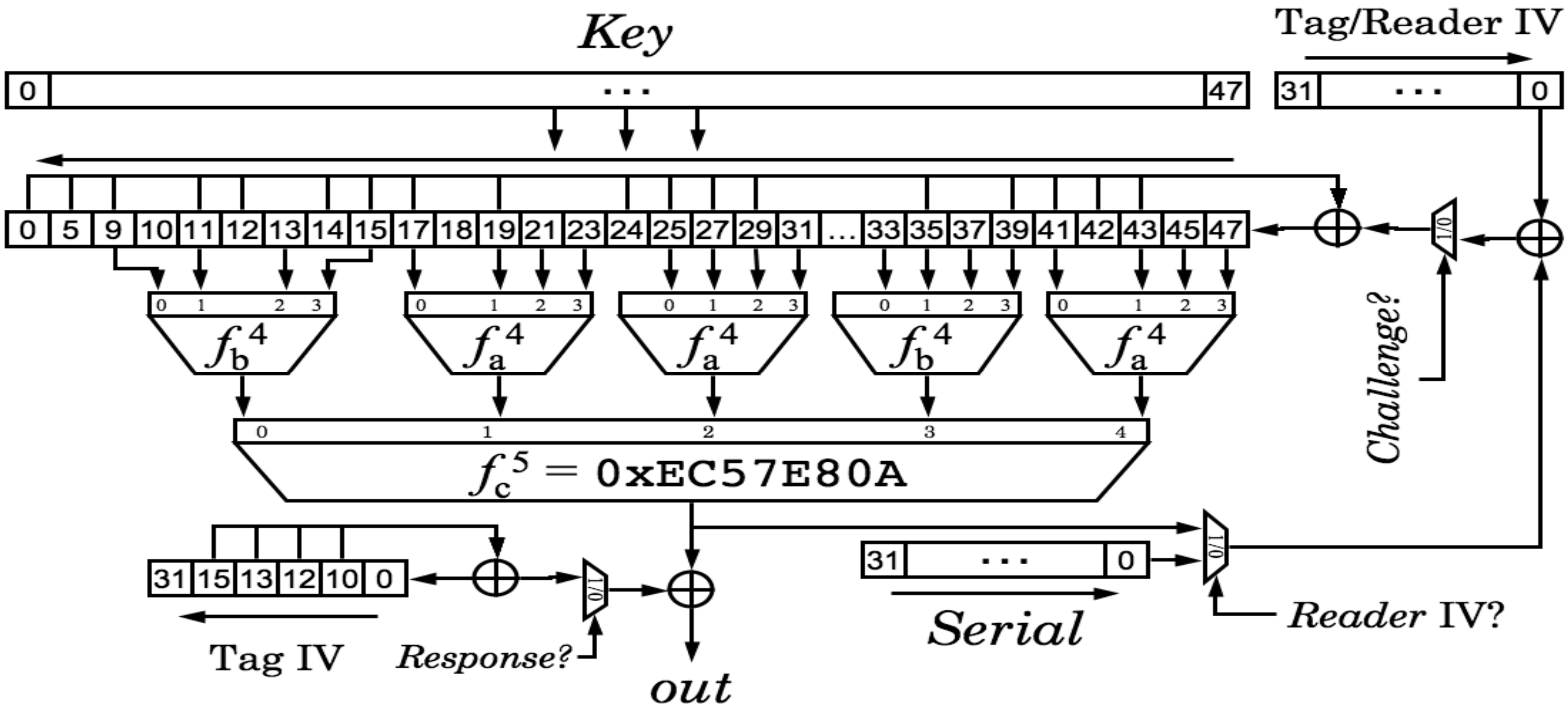
- Used in many office and official buildings
- Public transport systems
  - OV-Chipkaart (Netherlands)
  - Oyster card (London)
  - Smartrider (Australia)
  - EMT (Malaga) ☺
- Personnel entrance to Schiphol Airport (Amsterdam)
- Access to Dutch military bases
- Popular payment system in Asia

# Manufacturer response- freedom of publishing ?

**Timeline**

| | |
|---|---|
| Dec 2007 | CCC presentation by Nohl and Plotz |
| March 2008 | We recover CRYPTO1 and found attacks. |
| March 2008 | We notified the manufacturer and other stakeholders (without disclosure). |
| Jun 2008 | NXP tries to stop “irresponsible” publication, via injunction (court order). |
| July 2008 | Judge refuses to prohibit, basically on freedom of expression. Also: |
| | “University acted with due care, warning stakeholders early on” |
| | “Damage is not result of publication, but of apparent deficiencies in the cards” |
| | NXP did not appeal |

# Crypto1 Cipher



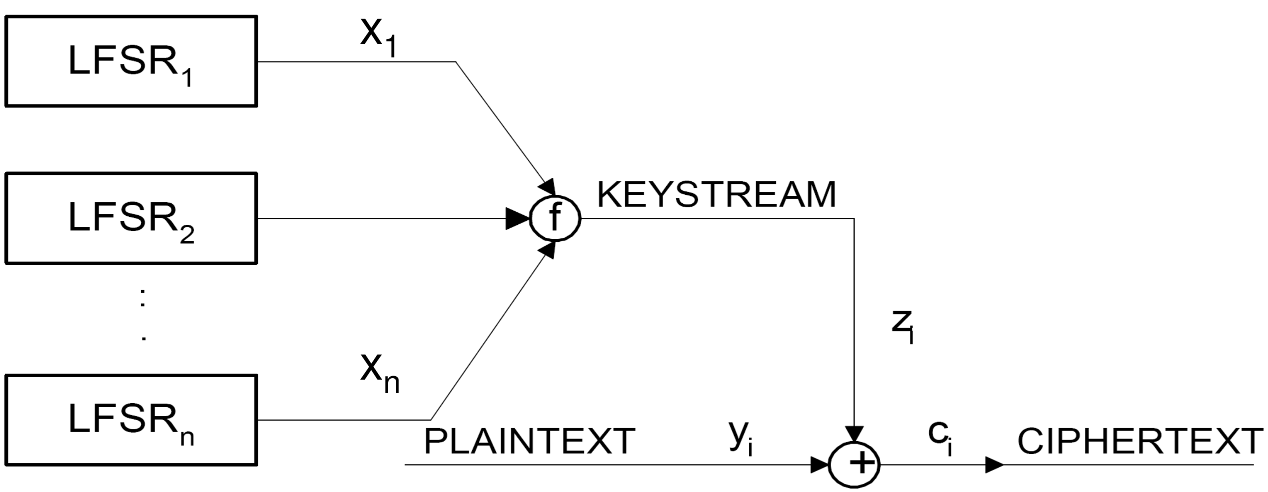

$$f_a^4 = \mathtt{0x9E98} = (a+b)(c+1)(a+d)+(b+1)c+a$$

$$f_b^4 = \mathtt{0xB48E} = (a+c)(a+b+d)+(a+b)cd+b$$

Tag IV ⊕ Serial is loaded first, then Reader IV ⊕ NFSR

- Attacking MIFARE (2 seconds on a laptop)

# Nonlinear combiner (RFID applications)



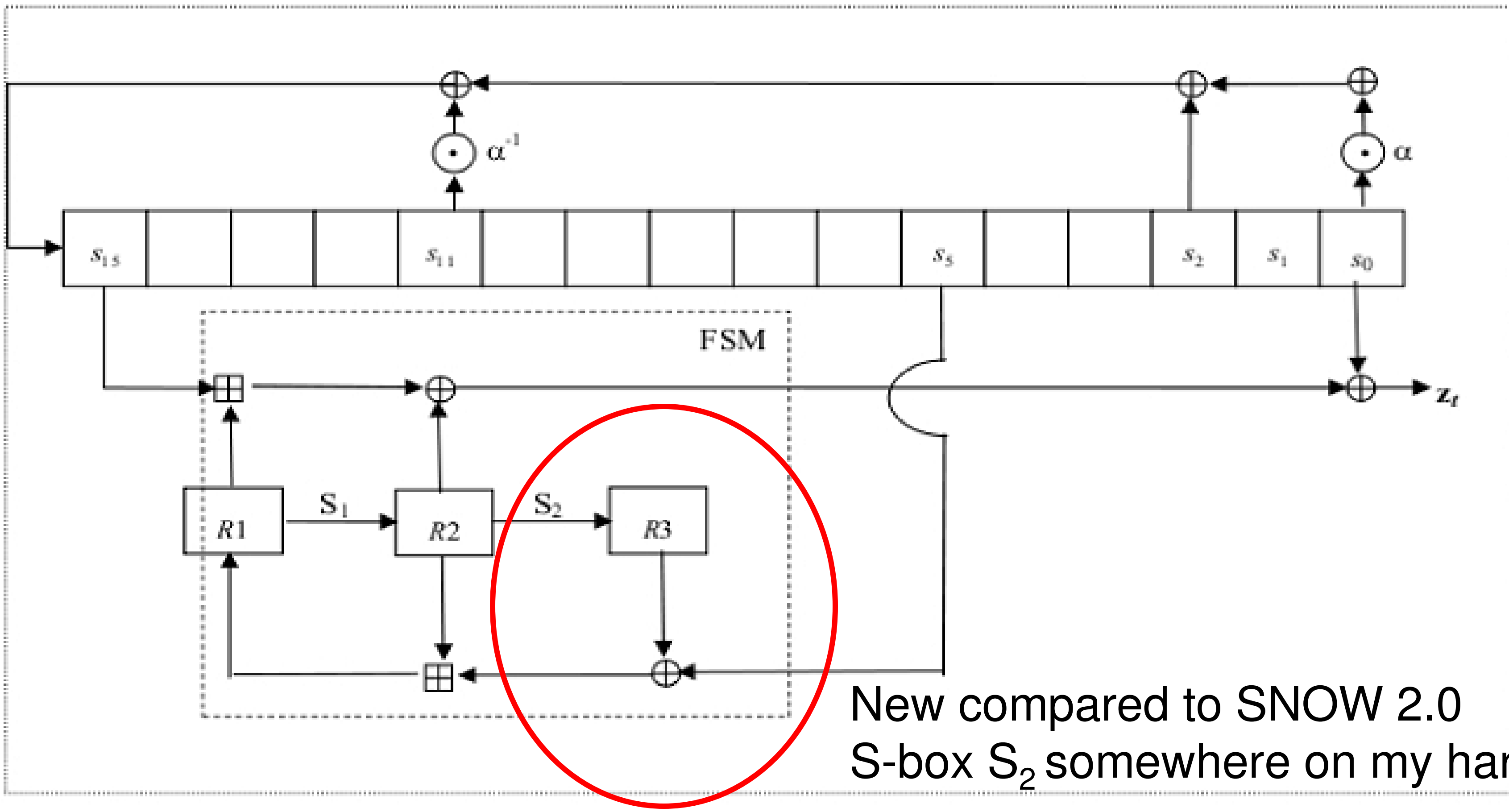

- Period of length $\prod_{i=1}^{n}(2^{L_i} - 1)$.
- Linear complexity is evaluation of Boolean function over integers !

**Problem** : Design secure Boolean function *f* !

# ZUC algorithm – SNOW variant

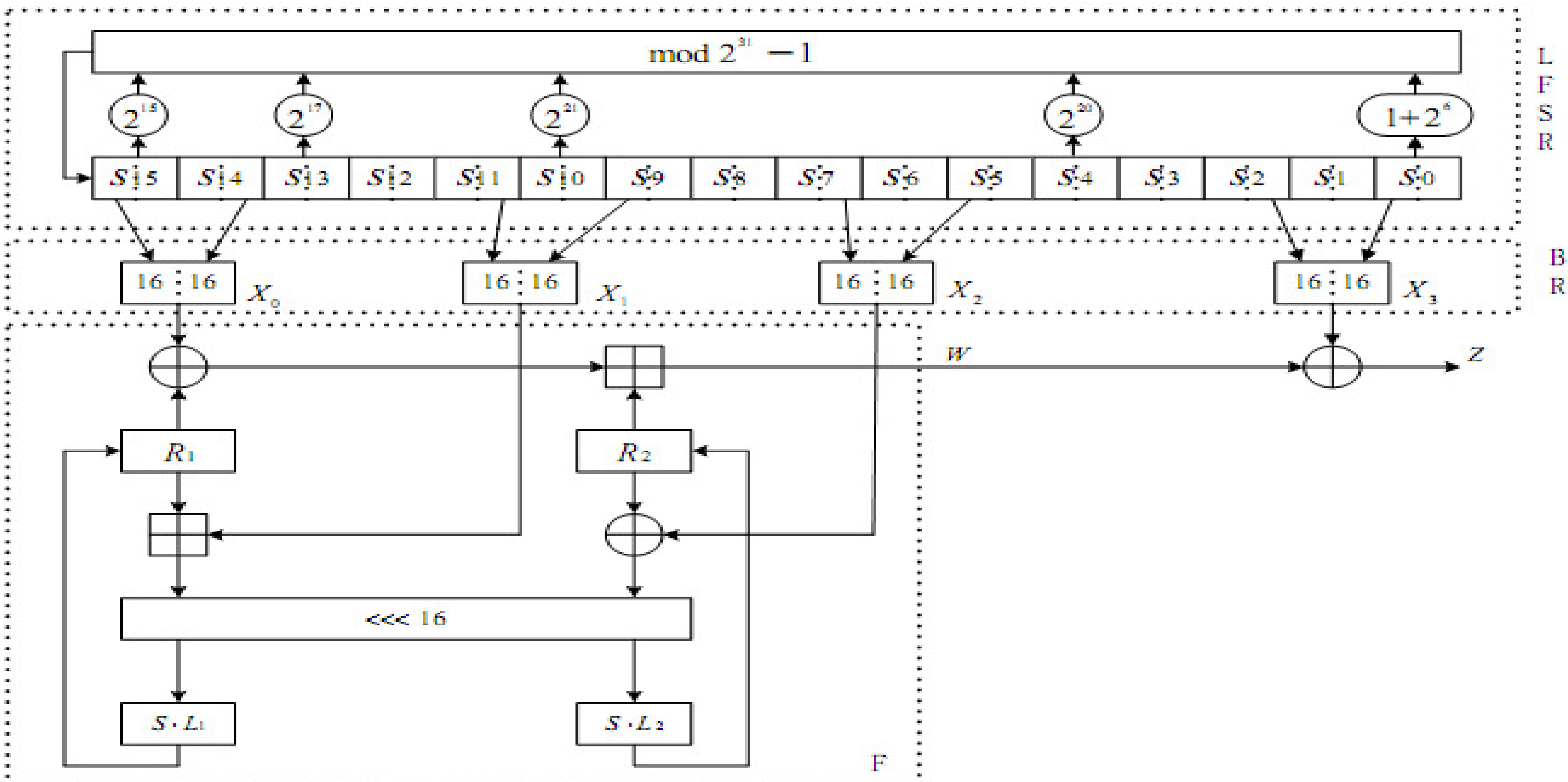
- SNOW 1.0 and 2.0 were developed in Lund in early 2000 (while I was developing better primitives Thomas and Patrik were designing a cipher ☺ )
- SNOW 3.0 was developed for 3G using some nonlinear ”secure” permutations over GF(2^8) of mine (resistant to algebraic attacks)
- After a few more modifications SNOW 3.0 became ZUC – very strong design comprehending all inteligent design strategies developed last 30 years

# SNOW 3G - design




New compared to SNOW 2.0
S-box $S_2$ somewhere on my hard disc

# ZUC algorithm




⊕ exclusive-OR

⊞ module $2^{32}$ addition

\+ the addition

<<<k the k-bit cyclic shift

# Useful transforms for cryptography

- Main tool is **Walsh-Hadamard spectra (graphs)**

$f(x) = 1 \oplus x_1 \oplus x_3 \oplus x_2 x_3$ ANF

| $x_3$ | $x_2$ | $x_1$ | $f$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

$$\begin{array}{ccccc} \boldsymbol{M} & & f & & W_f \\ \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \end{bmatrix} & & \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} & = & \begin{bmatrix} 4 \\ 2 \\ 0 \\ -2 \\ 0 \\ 2 \\ 0 \\ 2 \end{bmatrix} \end{array}$$

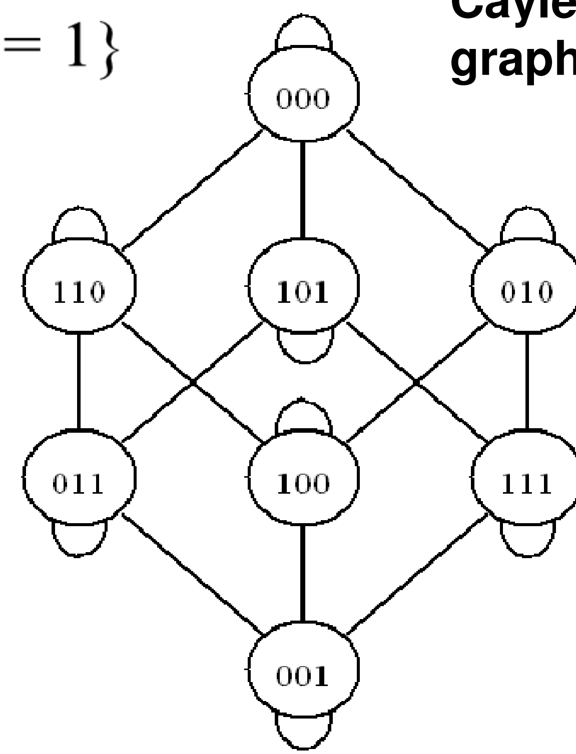
$V = GF(2)^n$ $\quad W_f(y) = \sum_{x \in V} f(x)(-1)^{x \bullet y}$ $- Walsh - Hadamard\ transform$

# Cayley graph representation

- Set of vertices $V$ – set of points

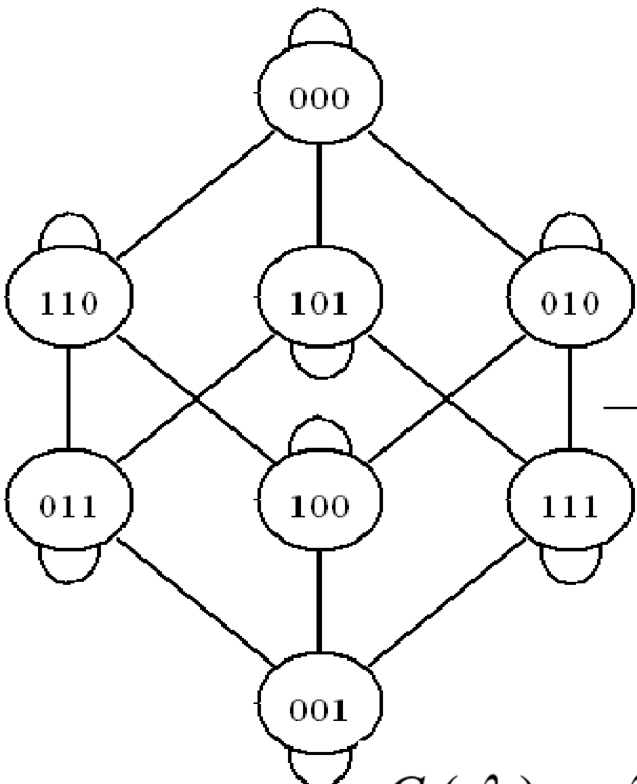
$$E = \{(m_i\ m_j) \in B^n \times B^n \mid f(m_i \oplus m_j) = 1\}$$

| $x_3$ | $x_2$ | $x_1$ | $f$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |




**Cayley graph**

# Cayley graph - eigenvalues




$$A = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$C(\lambda) = \lambda^8 - 8\lambda^7 + 16\lambda^6 + 16\lambda^5 - 80\lambda^4 + 64\lambda^3$$

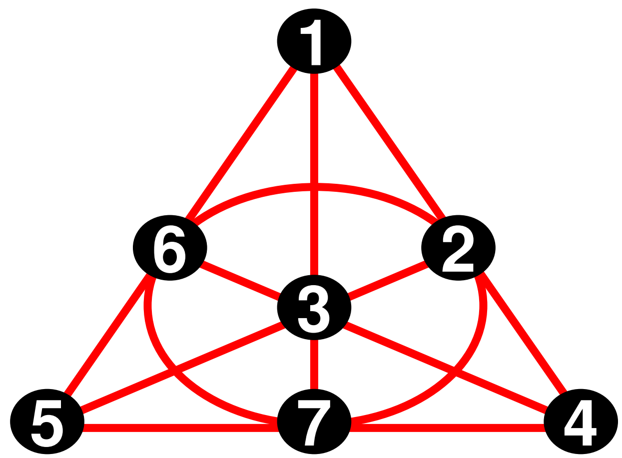
Find the roots – (4,2,0-2,0,2,0,2)

# Some open problems

- How to find "good" functions through Cayley graphs ?
- What are "good" functions ?
  - **high degree**
  - **algebraic immunity** (no low degree function *g* such that *fg* =0)
  - **large distance to affine functions** and other cryptographic criteria
- Algebraic representation currently seems to be more suitable than graph theoretical tools or ...
- **Research problem**: What is graph like if *f* is constant or affine on some $k-$dimensional flat ($k-$ *normality) ? What is the graph of linear combinations of several functions ? .....*

# Hypergraphs

**Hypergraph**: A set (called “vertices”) and a set of sets of vertices (called “edges” or sometimes “hyperedges”).




- **Example of a 3-uniform hypergraph**: The “Fano Plane”, V = {1,2,3,4,5,6,7} and
- E = {{1,2,4},{2,3,5},{3,4,6},{4,5,7},{5,6,1},{6,7,2},{7,1,3}}.

# Transversals and *annihilators*

- Algebraic attacks commonly use annihilators of $f$ i.e. existence of low degree $g$ s.t. $fg$ =0. (more variants)
- In 2008, Zhang, Pieprzyk and Zhang showed that transversal $T$ - subset of $V$ of a "Boolean hypergraph"

$$T \cap e_j \neq \emptyset \quad \forall e_j \in E$$

correspond to **annihilator** of $f$ !

- **Problem : Transversals found by greedy algorithm not optimal (lowest degree) and**
- **No connection to** $fg = h$ **for low degree** $g$, $h$.

# Bent functions - as a special class

- Favourite combinatorial objects (difference sets, coding, CDMA, ...).
- Fix a basis of $GF(2^n)$ to get isomorphism $GF(2^n) \cong GF(2)^n$ and define for $f : GF(2^n) \to GF(2)$,

$$W_f(a) = \sum_{x \in GF(2^n)} (-1)^{f(x) + Tr(ax)},$$

  If $|W_f(a)| = 2^{n/2}$ for all $a \in GF(2^n)$ then $f$ is **bent**.
- Maximum distance (uniform) to affine functions, $n$ even !!
- Many known classes, potentially for $n = 2k$ one may consider:

$$f : GF(2^n) \to GF(2)$$

$$f(x) = Tr(ax^{2^k - 1} + bx^{r(2^k - 1)}); \quad a, b \in GF(2^n), r \in \mathbb{N}.$$

# Multiple output bent and hyperbent functions

- Then you might get a BENT FUNCTION for some $a, b \in GF(2^n)$ and $r$ positive number ... Take $a = 1$, $r = 3$ and find $b$ by computer ...

- Nyberg proved in 1992 that the maximum output bent space is $n/2$ in binary case !

- Meaning: One can find $f_1, \ldots, f_k$, $f_i : GF(2)^n \to GF(2)$ (multiple bent $F : GF(2)^n \to GF(2)^k$) such that

$$a_1 f_1 + \ldots + a_k f_k \qquad \text{is bent } \forall a \in GF(2)^n \setminus \{0\}.$$

- Furthermore, define HYPERBENT function so that $f(x^i)$ is bent for any $i$ s.t. $\gcd(i, 2^n - 1) = 1$.

# Finding bent++ functions

- How to find such classes ?
- Instead of absolute trace use relative trace:

$$Tr_k^n(x) = x + x^2 + x^{2^2} + \ldots + x^{2^{n-k}},$$

  a function from $GF(2^n) \to GF(2^k)$.

- Consider instead

$$F(x) = Tr_k^n(ax^{2^k-1} + bx^{r(2^k-1)})$$

- Our class with explicit calculation of $a, b, r$ (Pasalic *et al.* 2012, 2013) is both bent, multiple bent, multiple hyperbent - it cannot be more bent than that :)

# All credits go to Dillon !

- The exponent $2^k - 1$ is known as Dillon's exponent, and for $n = 2k$ we have:

$$2^n - 1 = (2^k - 1)(2^k + 1).$$

- Note that $\#GF(2^k) \setminus 0 = 2^k - 1$, and there is a cyclic group $U$ of $(2^k + 1)$th roots of unity of size $2^k + 1$ !!

- Simply take a primitive $\alpha \in GF(2^n)$ and consider:

$$\{\alpha^{(2^k-1)i} : i = 0, \dots 2^k\} = U.$$

- **Meaning:**

$$GF(2^n)^* = \cup_{u \in U} uGF(2^k)^*$$

# Application of the unity circle

- We were interested in the functions of type

$$f_{a,r}(x) = Tr(x^{2^k-1} + ax^{r(2^k-1)})$$

Then, since $x \in GF(2^n)$ can be written (uniquely) as $x = uy$ for $u \in U$, $y \in GF(2^k)$,

$$\begin{aligned} f_{a,r}(x) &= f_{a,r}(yu) \\ &= Tr_1^n(u^{2^k-1}y^{2^k-1} + au^{(2^k-1)r}y^{(2^k-1)r}) \\ &= Tr_1^n(u^{2^k-1} + au^{(2^k-1)r}) \\ &= f_{a,r}(u). \end{aligned}$$

## Application of the unity circle II

- Thus, when computing

$$W_f(a) = \sum_{x \in GF(2^n)} (-1)^{f(x) + Tr(ax)},$$

we end up with something like

$$\begin{aligned} W_f(\lambda) &= \sum_{x \in \mathbb{F}_{2^n}} (-1)^{f_{a,r}(x) + Tr_1^n(\lambda x)} \\ &= 1 + \sum_{u \in U} \sum_{y \in \mathbb{F}_{2^k}^*} (-1)^{f_{a,r}(yu) + Tr_1^n(\lambda y u)} \\ &= 1 + \sum_{u \in U} (-1)^{f_{a,r}(u)} \sum_{y \in \mathbb{F}_{2^k}^*} (-1)^{Tr_1^n(\lambda y u)} = \dots \end{aligned}$$

# Planar mappings

- From quadratic planar mappings you get commutative semifields (not associative) and affine/projective planes !
- Definition:
$$F(x + a) - F(x),$$
a permutation for any nonzero $a \in \mathbb{F}_q$ and $F : \mathbb{F}_q \to \mathbb{F}_q$ !
- **Example :**$F(x) = x^2$ is planar **over any field of odd characteristic.**

- **PROOF:** $F(x + a) - F(x) = x^2 + 2ax + a^2 - x^2 = 2ax + a^2$, permutation since any linear polynomial is permutation !
- What if the characteristic of $\mathbb{F}_q$ is $p = 2$ ?

- NO planar mappings over $GF(2^n)$ since for any $b$ if $x_0$ is a solution to $F(x + a) + F(x) = b$ so is $x_0 + a$

# Everything can be extended - Part II

- But planar functions only exist for $p \neq 2$. Well, define (extend):

$$\mathcal{F}_f(a) = \sum_{x \in \mathbb{F}_p^n} \omega^{f(x)+a\cdot x}, \quad \omega = e^{\frac{2\pi i}{p}}. \tag{2}$$

- Then $f : GF(p)^n \to GF(p)$ is bent iff $|\mathcal{F}_f(a)| = p^{n/2}$ for any $a \in GF(p)^n$.

- What this got to do with planar mappings ?

- $F : GF(p^n) \to GF(p^n)$ is planar iff

$$s_1 f_1 + \ldots + s_n f_n \quad ,$$

is **bent** for all $(s_1, \ldots, s_n) \in GF(p)^{n*}$ !!!

# Some final comments

- Lots of quadratic planar mappings

$$F(x) = \sum_{0 \le k,j < n} \lambda_{k,j} x^{p^k + p^j}, \quad \lambda_{k,j} \in \mathbb{F}_{p^n},$$

added an affine function $A(x) = \sum_{0 \le i < n} a_i x^{p^i}$

- Derivatives are linearized polynomials, easy to handle !

- Nontrivial interesting class of planar mappings is:

$$F(x) = x^{\frac{3^t+1}{2}}$$

over $\mathbb{F}_{3^n}$, where $t$ is odd and $\gcd(t, n) = 1$.

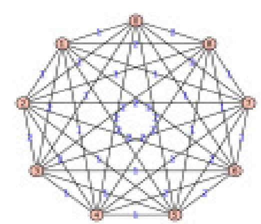
- The only example of nonquadratic planar mappings - hard to find !!!

# Bent functions over GF(p)

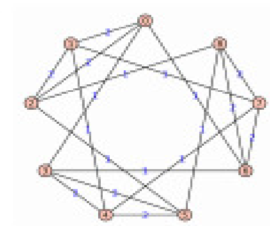
There are exactly 18 even bent functions $GF(3)^2 \longrightarrow GF(3)$ sending 0 to 0.

| $GF(3)^2$ | (0, 0) | (1, 0) | (2, 0) | (0, 1) | (1, 1) | (2, 1) | (0, 2) | (1, 2) | (2, 2) |
|---|---|---|---|---|---|---|---|---|---|
| $b_1$ | 0 | 1 | 1 | 1 | 2 | 2 | 1 | 2 | 2 |
| $b_2$ | 0 | 2 | 2 | 1 | 0 | 0 | 1 | 0 | 0 |
| $b_3$ | 0 | 1 | 1 | 2 | 0 | 0 | 2 | 0 | 0 |
| $b_4$ | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 1 |
| $b_5$ | 0 | 0 | 0 | 2 | 1 | 0 | 2 | 0 | 1 |
| $b_6$ | 0 | 1 | 1 | 0 | 2 | 0 | 0 | 0 | 2 |
| $b_7$ | 0 | 0 | 0 | 1 | 2 | 0 | 1 | 0 | 2 |
| $b_8$ | 0 | 2 | 2 | 0 | 0 | 1 | 0 | 1 | 0 |
| $b_9$ | 0 | 0 | 0 | 2 | 0 | 1 | 2 | 1 | 0 |
| $b_{10}$ | 0 | 2 | 2 | 2 | 1 | 1 | 2 | 1 | 1 |
| $b_{11}$ | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 1 | 2 |
| $b_{12}$ | 0 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | 2 |
| $b_{13}$ | 0 | 1 | 1 | 2 | 2 | 1 | 2 | 1 | 2 |
| $b_{14}$ | 0 | 1 | 1 | 0 | 0 | 2 | 0 | 2 | 0 |
| $b_{15}$ | 0 | 0 | 0 | 1 | 0 | 2 | 1 | 2 | 0 |
| $b_{16}$ | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 2 | 1 |
| $b_{17}$ | 0 | 2 | 2 | 1 | 1 | 2 | 1 | 2 | 1 |
| $b_{18}$ | 0 | 1 | 1 | 2 | 1 | 2 | 2 | 2 | 1 |

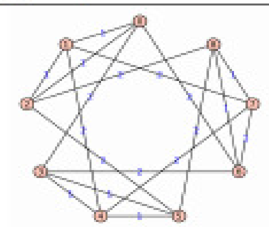
# Corresponding graphs



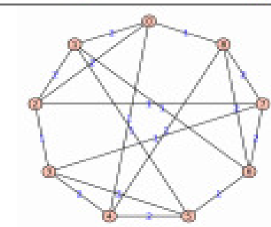
The Cayley graph for $b_1 = x_0^2 + x_1^2$



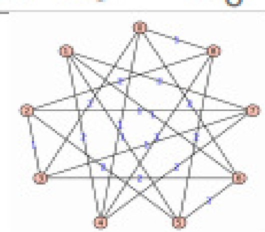
... for $b_2 = -x_0^2 + x_1^2$



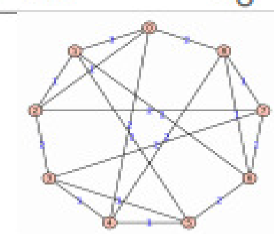
... for $b_3 = x_0^2 - x_1^2$



... for $b_4 = -x_0^2 - x_0 x_1$



... for $b_5 = -x_0 x_1 - x_1^2$



... for $b_6 = x_0^2 + x_0 x_1$

**Thanks for your patience !**