Repetition: 9.4.1 GM Cryptosystem

i) $n = p \cdot q$, $p \neq q$ primes
ii) Choose $y \in \mathbb{Z}_n$ a QNR mod n and $\left(\frac{y}{n}\right) = 1$
iii) Public key (n, y) private key (p, q)

Encryption: $m = (m_1, \dots, m_t) \in \{0,1\}^t$

$$c_i = \begin{cases} y \cdot x_i^2 \bmod n, & \text{if } m_i = 1 \\ x_i^2 \bmod n, & \text{if } m_i = 0 \end{cases} \qquad i = 1, \dots, t$$

$x_1, \dots, x_t \in \mathbb{Z}_n^*$: random

~~cypert~~ cyphertext $c = (c_1, \dots, c_t)$

Decryption: Let $m_i = \begin{cases} 0 & \text{if } \left(\frac{c_i}{p}\right) = 1 \\ 1 & \text{otherwise} \end{cases} \qquad i = 1, \dots, t$

Security of the GM Cryptosystem

An opponent intercepts $c_i = \begin{cases} y \cdot x_i^2 \bmod n, & \text{if } m_i = 1 \\ x_i^2 \bmod n, & \text{if } m_i = 0 \end{cases}$

hence, a random QR or pseudosquare mod n.

„To decide whether $m_i = 0$ or 1, Oscar needs to solve $QRP(c_i, n)$" If QRP is comput. infeasible, then O can not do better than guessing m_i.

Remark 9.14 /

A major drawback of the GM cryptosystem ~~A~~ is the message expansion by a factor of $\log_2(n)$ bits. To assure security presently about 300 dec. digits of n is needed, which means an expansion by a factor ~ 1024.

9.4.2 Blum-Goldwasser Cryptosystem

- Key generation:

(i) $p \neq q$ primes $p, q \equiv 3 \pmod 4$, $n = p \cdot q$

(ii) Compute a, b with $a p + b \cdot q = 1$ // EEA

(iii) Public key n, private key (p, q, a, b)

- Encryption: Let $h \leq \lfloor \log_2 \lfloor \log_2 (n) \rfloor \rfloor$

Message $m = (m_1, \dots, m_t) \in \{0,1\}^{ht}$, each m_i is of size h (bits)

<u>Blum-Blum-Shub (BBS) generator</u> for generating pseudo random bits b_i

- Select a random QR mod n: x_0
  (Select randomly $s \in \mathbb{Z}_n^*$, let $x_0 \equiv s^2 \bmod n$)
- Iterate: $x_i = x_{i-1}^2 \bmod n$ $\quad i = 1, \dots, t+1$
  b_i denote the h least significant (last) bits of x_i

$c_i = m_i \oplus b_i$

Ciphertext: $c = (c_1, \dots, c_t, x_{t+1})$

- Decryption

$d_1 = \left(\frac{p+1}{4}\right)^{t+1} \bmod (p-1)$, $d_2 = \left(\frac{q+1}{4}\right)^{t+1} \bmod (q-1)$

$u = (x_{t+1})^{d_1} \bmod p$, $v = (x_{t+1})^{d_2} \bmod q$

$x_0 = (v \cdot a \cdot p + u \cdot b \cdot q) \bmod n$ $\quad m_i = c_i \oplus b_i$

<u>Prop. 9.15</u> The decryption of the BG Cryptosystem is correct

<u>Proof</u>: The only point remaining is to show that x_0 is correct. ~~$\forall i = 0$~~

$\forall i = 0, \dots, t+1$

x_i QR mod $n \overset{\text{Prop 9.7}}{\Rightarrow} x_i$ QR mod $p \overset{\text{Prop 9.2}}{\Rightarrow} x_i^{(p-1)/2} \equiv 1 \pmod p$

Hence,

$$x_{i+1}^{(p+1)/4} \equiv \left((x_i)^2\right)^{(p+1)/4} \equiv x_i^{(p+1)/2} \equiv x_i \, x_i^{(p-1)/2} \equiv x_i \pmod p \quad (*)$$

By induction it follows:

$$u = (x_{t+1})^{d_1} \overset{(**)}{\equiv} (x_{t+1})^{((p+1)/4)^{t+1}} \equiv \left[(x_{t+1})^{(p+1)/4}\right]^{((p+1)/4)^t}$$

$$\overset{(*)}{\equiv} (x_t)^{((p+1)/4)^t} \equiv \ldots \equiv x_1^{(p+1)/4} \overset{(*)}{\equiv} x_0 \pmod p$$

[In $(**)$: $d \equiv e \pmod{p-1} \Rightarrow x^d \equiv x^e \pmod p$

$\exists k : d = e + k(p-1)$

$$x^d \equiv x^e x^{k(p-1)} \equiv x^e \underbrace{\left(x^{p-1}\right)^k}_{\equiv 1 \text{ Fermat}} \equiv x^e \pmod p \,]$$

Analogously $v \equiv x_{t+1}^{d_2} \equiv x_0 \pmod q$

~~By CRT~~ : $x_0 \equiv v \cdot a \cdot p + u \cdot b \cdot q \pmod p$

$x_0 \equiv v \cdot a \cdot p + u \cdot b \cdot q \pmod q$

By Prop 8.1 : $v \cdot a \cdot p + u \cdot b \cdot q \equiv x_0 \pmod n$

Example 9.15 (with artificially small parameters)

Key generation: $p = 499$, $q = 547$ $\quad (\equiv 3 \pmod 4)$

$n = p \cdot q = 272953$

EEA: $a = -57$, $b = 52$ $\quad (ap + bq = 1)$

Encryption: $h = 4$, $t = 5$

$m = (m_1, \dots, m_5) = (1001 \,|\, 1100 \,|\, 0001 \,|\, 0000 \,|\, 1100)$

choose random $r \in \mathbb{Z}_n^*$: $r = 399$

$x_0 = 399^2 \pmod n = 159201$

i	$x_i = x_{i-1}^2 \bmod n$	b_i	$c_i = m_i \oplus b_i$
1	180539	1011	0010
2	193932	1100	0000
3	245613	1101	1100
4	130286	1110	1110
5	40632	1000	0100
6	139680	–	–

$c = (0010 \,|\, 0000 \,|\, 1100 \,|\, 1110 \,|\, 0100, 139680)$

Decryption

$$d_1 = \left(\frac{p+1}{4}\right)^{t+1} \bmod (p-1) = 463$$

$$d_2 = \left(\frac{q+1}{4}\right)^{t+1} \bmod (q-1) = 337$$

$$u = (x_{t+1})^{d_1} \bmod p = 20$$

$$v = (x_{t+1})^{d_2} \bmod q = 24$$

$$x_0 = v \cdot a \cdot p + u \cdot b \cdot q \bmod n = 159201$$

Security of the BG cryptosystem

a) An eavesdropper sees the QR x_{t+1}. To determine x_t means to solve $QRSP(x_{t+1}, n)$, which is considered comput. infeasible.

b) The BG cryptosystem is vulnerable to chosen ciphertext attacks.

Opponent access x_t (or x_0, then $x_t = x_0^{2^{t-1}} \bmod n$)

Opponent selects randomly $m \in \mathbb{Z}_n^*$, computes $x_{t+1} = m^2$

There are 4 solutions of $x_{t+1} = m^2 \pmod n$

If $x_t \neq \pm m$ then $\gcd(x_t - m, n) \in \{p, q\}$

If $x_t = \pm m$ then select a new random number m

This attack is analogous to the one against the Rabin cryptosystem.

Efficiency of the BG system

a) The message expansion is constant by $\lceil \log_2(n) \rceil$ bits, the representation of x_{t+1}

b) Computational effort is comparabel to RSA, both in the encryption and decryption step.

10. Cryptographic Hash Functions

One-way hash function: mapping messages of arbitrary length to a digest of fixed length n, typically $n = 64, 128, 160$ bits.

Applications:

- Signature Schemes, sign the hash of a document rather than a long document itself
- Data integrity, software protection, protection against viruses
  (MDC – Modification / Manipulation detection code
  MAC – Message authentication code)

Hash fcts are typically publicly known and involve no secret keys.

Formal description of hash functions:

$\mathcal{M}$: message space (e.g., $\mathcal{M} = \bigcup_{\ell=0}^{\infty} \{0,1\}^\ell = \{0,1\}^*$)

$\mathcal{Y}$: Finite set of possible hash values (digest, hash digest, authentication tags) (e.g. $\mathcal{Y} = \{0,1\}^{128}$)

$\mathcal{K}$: key space (finite set)

h : hash function $h : \mathcal{M} \times \mathcal{K} \to \mathcal{Y} : (m, k) \mapsto h(m, k)$

h is called unkeyed, if $|\mathcal{K}| = 1$ or $h : \mathcal{M} \to \mathcal{Y}$

$(m, h(m))$ is called a valid pair.

10.1 Security of hash functions

In the following we are considering unkeyed hash functions
„It is computationally infeasible to compute preimages or to generate a collision" leads to the following.

Basic properties of cryptographic hash functions $h: \mathcal{M} \to Y$

1. Given $m \in \mathcal{M}$, $h(m)$ is easy to compute

Further, the solution of the following problems is comput. infeasible

2. Given $y \in Y$, find $m \in \mathcal{M}$ such that $h(m) = y$.
   In this case, h is called one-way function or preimage resistant.
3. Given $m \in \mathcal{M}$, find $m' \neq m$ such that $h(m') = h(m)$
   In this case, h is called second preimage resistant
4. Find $m \neq m' \in \mathcal{M}$ such that $h(m) = h(m')$
   In this case h is called (strongly) collision free.

   Note: Both m and m' may be freely chosen.