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 $a \cdot p + b \cdot q = 1$

(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)

Blum-Blum-Shub (BBS) generator for generating pseudo random bits b_i

- Select a random QR mod n: x_0
  (Select randomly $r \in \mathbb{Z}_n^*$, let $x_0 \equiv r^2 \bmod n$)
- Iterate: $x_i = x_{i-1}^2 \bmod n$ $\quad i = 1, \dots, t+1$
  b_i denotes 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:

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

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

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

Iterate: $x_i = x_{i-1}^2 \bmod n$ $\quad i = 1, \dots, t(+1)$

$m_i = c_i \oplus b_i$

Prop. 9.15 The decryption of the BG cryptosystem is correct.

Proof: The only remaining point is to show that x_0 is correct.

$\forall i = 0, \dots, t-1$: x_i QR mod $n \overset{\text{Prop 9.7}}{\Rightarrow}$ x_i QR mod $p \overset{\text{Prop 9.2}}{\Rightarrow} x_i^{\frac{(p-1)}{2}} \equiv 1 \pmod{p}$

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

By induction it follows:

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

$$\overset{(*)}{\equiv} (x_t)^{\left(\frac{p+1}{4}\right)^t} \equiv \dots \equiv x_1^{\frac{p+1}{4}} \overset{(*)}{\equiv} x_0 \pmod{p}$$

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

$\exists k \in \mathbb{Z} : d = e + k(p-1)$

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

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

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

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

Example 9.15 (with artificially small parameters)

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

$n = p \cdot q = 272\,953$, $\log_2(\log_2(n)) \approx 4.175 \geq h$

EEA: $a \cdot p + b \cdot q = 1 = \gcd(p, q)$ $a = -57$, $b = 52$

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

$m = (m_1, \dots, m_5) = (1001 | 1100 \; 1000 1 | 0000 | 1100)$

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

$x_0 = 399^2 \pmod n = 159\,201$

i	$x_i = x_{i-1}^2 \bmod n$	b_i	$c_i = m_i \oplus b_i$
1	180 539	1011	0010
2	193 932	1100	0000
3	245 613	1101	1100
4	130 286	1110	1110
5	40 632	1000	0100
$6 = t+1$	139 680		

$C = (0010 | 0000 | 1100 | 1110 | 0100, 139\,680)$

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 computationally infeasible.

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

Opponent access x_t.

Opponent selects a random message $m \in \mathbb{Z}_n^*$, computes

$x_{t+1} = m^2 \mod n$

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

If $x_t \not\equiv \pm m$ then $\gcd(x_t - m, n) \in \{p, q\}$

If $x_t = \pm m$ then select an new random message m.

This attack is analogously to the one in the Rabin's 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 comparable to RSA, both in the encryption and decryption.

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 functions are typically publicly known and involve no secret keys.

Formal description of hash functions:

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

Y : Finite set of possible hash values (digest, hash digest, authentication tags) (e.g., $Y = \{0,1\}^n$; $n \in \{64; 128; 160\}$)

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

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

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

$(m, h(m))$ 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 collisions." leads to the following:

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

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

Further, the solution of the following problems is computationally 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 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.