Formula sheet of Introduction to Cryptography

1 Mathematical concepts

1.1 Euler's totient function

Let $n > 1$ be an integer such that $n = \prod_i p_i^{k_i}$, where p_i are distinct prime numbers and $k_i > 0$. Then $\varphi(n)$ is computed as

$$\varphi(n) = \varphi\left(\prod_i p_i^{k_i}\right) = \prod_i \varphi(p_i^{k_i}) = \prod_i (p_i^{k_i} - p_i^{k_i-1}) = \prod_i (p_i - 1)p_i^{k_i-1}.$$

1.2 A left-to-right Square-and-multiply algorithm

Data: Integers a, d, n
Result: x with $x \equiv a^d \pmod n$
Write $d = (d_{k-1}d_{k-2}\cdots d_1d_0)_2$
$x \leftarrow 1$
for $i = k-1$ *to* 0 **do**
  $x \leftarrow x^2 \bmod n$
  if $d_i = 1$ **then**
    $x \leftarrow ax \bmod n$
  end
end
return x

1.3 CRT, specifically for RSA

Suppose that we want to solve a system of modular equations like

$$\begin{cases} x \equiv a_0 & \pmod p; \\ x \equiv a_1 & \pmod q. \end{cases}$$

A solution is $x = u_0a_0 + u_1a_1 \pmod n$, where $u_0 = (q^{-1} \bmod p) \cdot q$ and $u_1 = (p^{-1} \bmod q) \cdot p$.

Garner's method:

A solution is $x = a_1 + q \cdot ((a_0 - a_1 \bmod p) \cdot (q^{-1} \bmod p) \bmod p)$.

2 Security strength

Advantage:

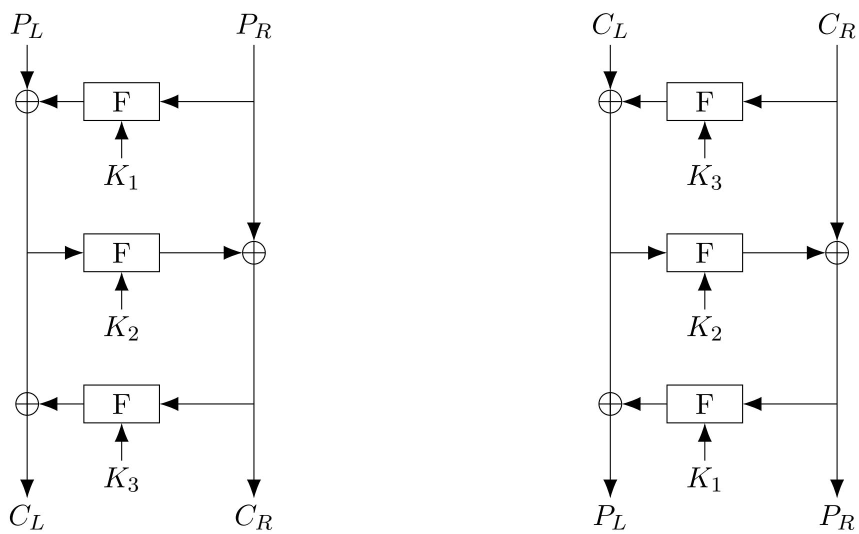
The advantage of distinguishing, e.g., a stream cipher SC with uniformly random key from a random oracle $\mathcal{RO}$ is given by: $\mathrm{Adv}_{\mathcal{A}} = |\Pr(\mathcal{A} = 1 \mid \mathrm{SC}_K) - \Pr(\mathcal{A} = 1 \mid \mathcal{RO})|$.

Security strength:

A cryptographic scheme offers security strength s if there are no attacks with $(M+N)/p < 2^s$ with N and M the adversary's (offline and online) resources and p the success probability, and there are no attacks with $(M+N)/\mathrm{Adv} < 2^s$ with N and M the adversary's (offline and online) resources and Adv the advantage of the adversary.

3 Symmetric cryptography

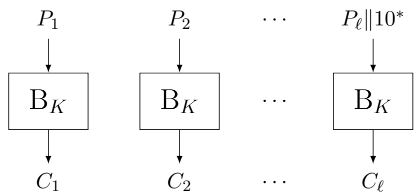
3.1 Feistel structure



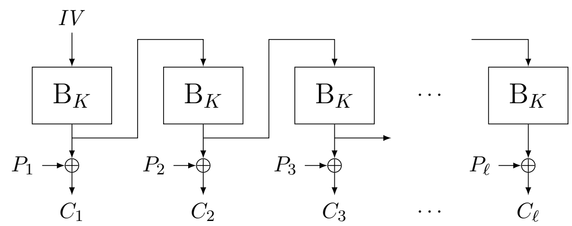
(a) Encryption (b) Decryption

Figure 1: Three-round Feistel structure.

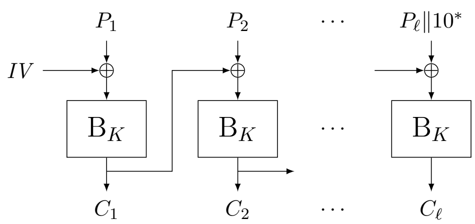
3.2 Block cipher modes



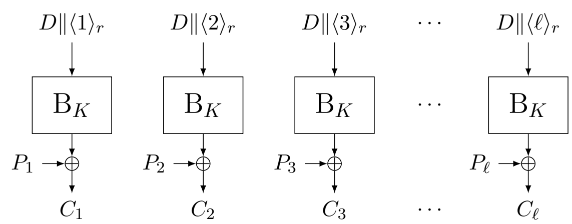
(a) Electronic codebook mode



(b) Output Feedback mode



(a) Cipher block chaining mode



(b) Counter mode

3.3 Hash function constructions

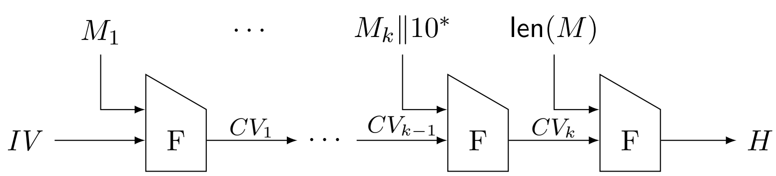


Figure 4: Merkle-Damgård construction for hash functions.

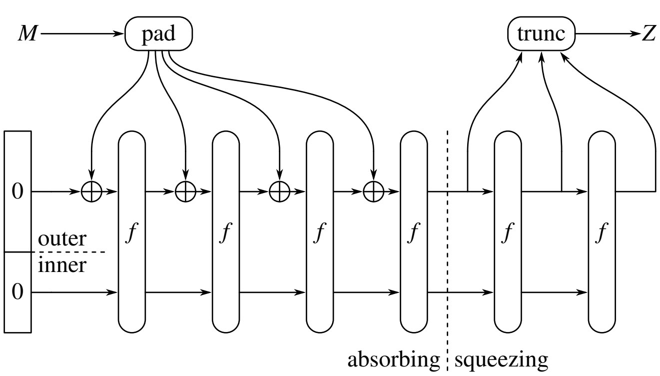


Figure 5: Sponge function.

4 Public-key cryptography

4.1 Key agreement schemes

4.1.1 Textbook (Merkle-)Diffie-Hellman key agreement

Alice		Bob
p, g, q		p, g, q
$a \xleftarrow{\$} \mathbb{Z}/q\mathbb{Z}$		$b \xleftarrow{\$} \mathbb{Z}/q\mathbb{Z}$
$A \leftarrow g^a$		$B \leftarrow g^b$
	$\xrightarrow{\text{Alice}, A}$	
	$\xleftarrow{\text{Bob}, B}$	
$K_{A,B} \leftarrow B^a$		$K_{B,A} \leftarrow A^b$

4.2 Encryption schemes

4.2.1 ElGamal encryption scheme

Alice		Bob
$p, g, (q), B$		$p, g, (q), b, B(= g^b)$
$a \xleftarrow{\$} \mathbb{Z}/q\mathbb{Z}$		
$A \leftarrow g^a$		
$C \leftarrow M \times B^a$	$\xrightarrow{(C,A)}$	$M \leftarrow C \times A^{q-b}$

4.2.2 Textbook RSA encryption scheme

Bob		Alice
Alice's public key (n, e)		Alice's private key (n, d)
$c \leftarrow m^e \bmod n$	$\xrightarrow{c}$	$m \leftarrow c^d \bmod n$

4.3 Key encapsulation mechanisms (KEM)

4.3.1 KEM from ElGamal

Alice		Bob
$p, g, (q), B$		$p, g, (q), b, B(= g^b)$
$a \xleftarrow{\$} \mathbb{Z}/q\mathbb{Z}$		
$A \leftarrow g^a$		
$K \leftarrow \mathrm{h}(\text{“KDF”}; B^a)$		
$CT \leftarrow \mathrm{Enc}_K(m)$	$\xrightarrow{(A,CT)}$	$K \leftarrow \mathrm{h}(\text{“KDF”}; A^b)$
		$m \leftarrow \mathrm{Dec}_K(CT)$

4.3.2 KEM from RSA

Bob has Alice's public key (n, e)		Alice with private key (n, d)
$r \xleftarrow{\$} \mathbb{Z}/n\mathbb{Z}$		
$c \leftarrow r^e \bmod n$		
$K \leftarrow \mathrm{h}(\text{“KDF”}; r)$		
$CT \leftarrow \mathrm{Enc}_K(m)$	$\xrightarrow{(c, CT)}$	$r \leftarrow c^d \bmod n$
		$K \leftarrow \mathrm{h}(\text{“KDF”}; r)$
		$m \leftarrow \mathrm{Dec}_K(CT)$

4.4 Authentication protocols

4.4.1 Chaum-Evertse-van de Graaf (CEG) protocol

Alice		Bob
p, g, q, A, a		p, g, q (Alice: A)
$v \xleftarrow{\$} \mathbb{Z}/q\mathbb{Z}$		
$V \leftarrow g^v$	$\xrightarrow{\text{Alice}, V}$	$c \xleftarrow{\$} \{0, 1\}$
	$\xleftarrow{c}$	
$r \leftarrow v - ca$	$\xrightarrow{r}$	$V \stackrel{?}{=} g^r A^c$

4.4.2 Schnorr's authentication protocol

Alice		Bob
p, g, q, A, a		p, g, q (Alice: A)
$v \xleftarrow{\$} \mathbb{Z}/q\mathbb{Z}$		
$V \leftarrow g^v$	$\xrightarrow{\text{Alice}, V}$	$c \xleftarrow{\$} \mathbb{Z}/q\mathbb{Z}$
	$\xleftarrow{c}$	
$r \leftarrow v - ca$	$\xrightarrow{r}$	$V \stackrel{?}{=} g^r A^c$

4.5 Signature schemes

4.5.1 Schnorr's signature scheme

Alice		Bob
p, g, q, A, a		p, g, q (Alice: A)
$v \xleftarrow{\$} \mathbb{Z}/q\mathbb{Z}$		
$V \leftarrow g^v$		
$c \leftarrow \mathrm{h}(p; g; A; V; m)$		
$r \leftarrow v - ca$	$\xrightarrow{\text{Alice}, m, (r, V)}$	$c \leftarrow \mathrm{h}(p; g; A; V; m)$
		$V \stackrel{?}{=} g^r A^c$

4.5.2 Full-domain hash RSA signatures

Alice with private key (n, d)		Bob with Alice's public key (n, e)
$H \leftarrow \mathrm{h}(m)$		
$s \leftarrow H^d \bmod n$	$\xrightarrow{\text{Alice},m,s}$	$H \leftarrow \mathrm{h}(m)$
		$H \stackrel{?}{=} s^e \bmod n$

4.5.3 Security notions

Discrete log (DL) problem:
Let $a \stackrel{\$}{\leftarrow} \mathbb{Z}/q\mathbb{Z}$ and $A \leftarrow g^a$. Given $\langle g \rangle$ and A, determine a.

Computational Diffie-Hellman (CDH) problem:
Let $a, b \stackrel{\$}{\leftarrow} \mathbb{Z}/q\mathbb{Z}$, $A \leftarrow g^a$ and $B \leftarrow g^b$. Given $\langle g \rangle$ and A, B, determine g^{ab}.

Decisional Diffie-Hellman (DDH) problem:
Let $a, b, c \stackrel{\$}{\leftarrow} \mathbb{Z}/q\mathbb{Z}$, and $A \leftarrow g^a$, and $B \leftarrow g^b$. With probability $\frac{1}{2}$, set $C \leftarrow g^c$, and otherwise $C \leftarrow g^{ab}$. Given $\langle g \rangle$ and A, B, C, determine whether $C = g^{ab}$ holds.

Advantage:

The advantage of an adversary on the decisional Diffie-Hellman problem is given by:

$$\mathrm{Adv}_{\mathcal{A}} = |\Pr(\mathcal{A} = 1 \mid C = g^{ab}) - \Pr(\mathcal{A} = 1 \mid C = g^c)|$$

IND-CPA security:

Challenger		Adversary
Domain parameters (if any)		Domain parameters (if any)
randomly generate (PrK, PK)	$\xrightarrow{PK}$	Repeat: $\mathrm{Enc}_{PK}(M)$
	$\xleftarrow{M_0, M_1}$	M_0, M_1 messages
$i \stackrel{\$}{\leftarrow} \{0, 1\}$		
$CT \leftarrow \mathrm{Enc}_{PK}(M_i)$	$\xrightarrow{CT}$	Repeat: $\mathrm{Enc}_{PK}(M)$

4.6 Elliptic curves

4.6.1 Addition formulas for Weierstrass curves over prime fields

An elliptic curve (in short Weierstrass form) is the set of points in $\mathbb{F}_p^2$ that satisfy

$$\mathcal{E}\colon y^2 = x^3 + ax + b, \qquad (a, b \in \mathbb{F}_p)$$

together with the point at infinity $\mathcal{O}$.

If points $P = (x_1, y_1)$, $Q = (x_2, y_2)$ are on curve $\mathcal{E}$, then we can compute their sum, $R = (x_3, y_3)$, algebraically as follows:

	$P = -Q$	$P \neq \pm Q$	$P = Q$
$R =$	$\mathcal{O}$	$\lambda = \frac{y_1 - y_2}{x_1 - x_2}$ $x_3 = \lambda^2 - x_1 - x_2$ $y_3 = -y_1 + \lambda(x_1 - x_3)$	$\lambda = \frac{3x_1^2 + a}{2y_1}$ $x_3 = \lambda^2 - 2x_1$ $y_3 = -y_1 + \lambda(x_1 - x_3)$

For a point $P = (x, y)$ on the curve $\mathcal{E}$, the inverse of P is the point $-P = (x, -y)$.

4.6.2 Projective coordinates

We can convert any point $(X : Y : Z)$ with $Z \neq 0$ to affine coordinates, as (XZ^{-1}, YZ^{-1}). The homogeneous elliptic curve has the form

$$Y^2 Z = X^3 + aXZ^2 + bZ^3.$$

The curve's point at infinity is $\mathcal{O} = (0 : 1 : 0)$.

4.7 Attacks on the discrete logarithm problem

We use multiplicative notation in the following. In additive notation, multiplications are replaced by additions and exponentiations by scalar multiplications.

4.7.1 Baby-step giant-step algorithm

Data: Group elements g, h and table size m
Result: Integer a such that $h = g^a$
$q \leftarrow \#\langle g \rangle$
$L \leftarrow [\ ]$
for $i = 0$ *to* m **do**
    $b_i \leftarrow g^i$
    Append(L, b_i)
end
$j \leftarrow 0$
repeat $c_j \leftarrow h \cdot g^{-m \cdot j}$
until $\exists i : c_j = L[i]$
then $i_0 \leftarrow i$
return $i_0 + m \cdot j$

4.7.2 Example of how to execute Pollard's ρ algorithm

Let p be a prime number such that $g \in (\mathbb{Z}/p\mathbb{Z})^*$ has order q. We want to solve the DL problem given $\langle g \rangle$ and h with $h = g^a$, to determine a.

We take as starting point $(g, 1, 0)$ and as our function:

$$(a_{i+1}, b_{i+1}, c_{i+1}) = \begin{cases} (a_i \cdot g, b_i + 1, c_i) & \text{if } a_i \equiv 1 \pmod 3; \\ (a_i \cdot h, b_i, c_i + 1) & \text{if } a_i \equiv 2 \pmod 3; \\ (a_i^2, 2b_i, 2c_i) & \text{if } a_i \equiv 0 \pmod 3. \end{cases}$$

When we find $i \neq j$ with $a_i = a_j$, then we have

$$g^{b_i} h^{c_i} \equiv g^{b_j} h^{c_j} \pmod p,$$

so we get

$$g^{b_i - b_j} \equiv h^{c_j - c_i} \equiv g^{x(c_j - c_i)} \pmod p.$$

We then find x by solving $b_i - b_j \equiv x(c_j - c_i)$ modulo q.