Introduction to Modern Cryptography - CheatSheet

WS20/21

Valentin Knappich Jan-Nicolai Geistler

February 20, 2021

Disclaimer: This document was created for exam preparation and has not been corrected by any of the lecturers/tutors. Therefore there is no guarantee for the correctness and/or comprehensiveness of the content.

Contents

1 Block-Cipher

1.1 Definition

Let $l : \mathbb{N} \to \mathbb{N}$ be a polynomial. Then the block-cipher is defined like this

$B = (\{0,1\}^l, Gen(1^\eta), E, D)$

Where $Gen(1^\eta)$ = probabilistic key generation

E = deterministic encrpytion

D = deterministic decryption

Only secure for scenario 2 (constant length no duplicated plaintext)

1.2 (Shortend) security game

$\mathbb{S}(1^\eta) : \{0,1\}$

1. *Choose real world or random world.*
   $b \xleftarrow{\$} \{0,1\}$
   if $b = 1$ then
   $k \xleftarrow{\$} \mathsf{Gen}(1^\eta)$ and $F = E(\cdot, k)$
   else
   $F \xleftarrow{\$} \mathcal{P}_{\{0,1\}^{l(\eta)}}$
2. *Guess phase.*
   $b' \xleftarrow{\$} U(1^\eta, F)$
3. *Output.*
   return b'.

1.3 Advantage

$$
\begin{aligned}
Adv_{U,B}(\eta) &= 2 \cdot (Pr[\mathbb{E}_U^B(1^\eta) = 1] - \frac{1}{2}) \\
&= Pr[\mathbb{S}_U^B\langle b = 1\rangle(1^\eta) = 1] - Pr[\mathbb{S}_U^B\langle b = 0\rangle(1^\eta) = 1]
\end{aligned}
$$

2 PRNG Game

2.1 Number Generator

Let $\eta \in \mathbb{N}$, p a polynomial and G a deterministic polynomial-time algorithm.

$G : (s : \{0,1\}^\eta) : \{0,1\}^{p(\eta)}$

p is expansion factor of G

2.2 PRNG-Distinguisher

Let $\eta \in \mathbb{N}$, p a polynomial and U is ppt algorithm.
$U(1^\eta, x : \{0,1\}^{p(\eta)}) : \{0,1\}$

2.3 (Shortend) Game

$\mathbb{S}_{U,G}^{PRNG}(1^\eta) : \{0,1\}$
1. *Choose real world or random world.*
   $b \xleftarrow{\$} \{0,1\}$
   if $b = 1$ then
     $s \xleftarrow{\$} \{0,1\}^\eta$ and $x = G(s)$
   else
     $x \xleftarrow{\$} \{0,1\}^{p(\eta)}$
2. *Guess phase.*
   $b' \xleftarrow{\$} U(1^\eta, x)$
3. *Output.*
   return b'.

2.4 Advantage

$$\begin{aligned} Adv_{U,G}(\eta) &= 2 \cdot (Pr[\mathbb{E}_{U,G}^{PRNG}(1^\eta) = 1] - \frac{1}{2}) \\ &= Pr[\mathbb{S}_{U,G}^{PRNG}\langle b = 1\rangle(1^\eta) = 1] - Pr[\mathbb{S}_{U,G}^{PRNG}\langle b = 0\rangle(1^\eta) = 1] \end{aligned}$$

3 CPA Security

3.1 Security game

$\mathbb{E}(1^\eta) : \{0,1\}$
1. *Choose cipher.*
   $k \xleftarrow{\$} \mathsf{Gen}(1^\eta)$; $H = E(\cdot, k)$
2. *Find phase.*
   $(z_0, z_1) \xleftarrow{\$} AF(1^\eta, H)$
3. *Selection.*
   $b \xleftarrow{\$} \{0,1\}$; $y \xleftarrow{\$} H(z_b)$
4. *Guess phase.*
   $b' \xleftarrow{\$} AG(1^\eta, H, y)$
5. *Evaluation.*
   if $b' = b$, return 1, otherwise 0.

3.2 Adversary

Let $\eta \in \mathbb{N}$ and the adversary being a ppt algorithm

$A(1^\eta, H : \{0,1\}^* \leftarrow \{0,1\}^*) : \{0,1\})$

$(AF(1^\eta, H), AG(1^\eta, H, y : \{0,1\}^*))$

AF = finder

AG = guesser

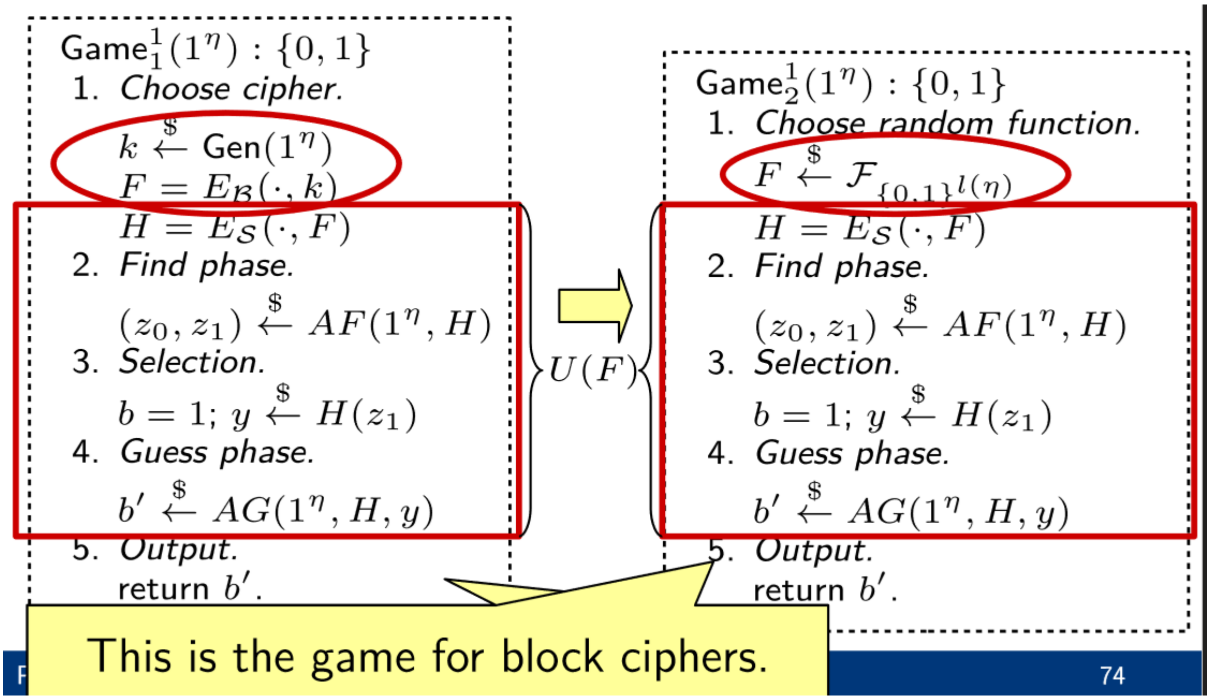
H = encryption oracle

3.3 Advantage

$$
\begin{aligned}
Adv_{U,G}(\eta) &= 2 \cdot (Pr[\mathbb{E}_{A,S}(1^{\eta}) = 1] - \frac{1}{2}) \\
&= Pr[\mathbb{S}_{A,S}\langle b = 1\rangle(1^{\eta}) = 1] - Pr[\mathbb{S}_{A,S}\langle b = 0\rangle(1^{\eta}) = 1]
\end{aligned}
$$

3.4 Proof

By security game switching from success game to failure game proofing that the distingusiher stays the same, but it is possible to change the Gen function to a pseudo-random function.



4 CCA

4.1 Security game

$\mathbb{E}(1^\eta) : \{0,1\}$

1. *Choose cipher.*
   $k \xleftarrow{\$} \mathsf{Gen}(1^\eta); H = E(\cdot, k); H^{-1} = D(\cdot, k)$
2. *Find phase.*
   $(z_0, z_1) \xleftarrow{\$} AF(1^\eta, H, H^{-1})$
3. *Selection.*
   $b \xleftarrow{\$} \{0,1\};\ y \xleftarrow{\$} H(z_b)$
4. *Guess phase.*
   $b' \xleftarrow{\$} AG(1^\eta, H, H^{-1}, y)$
5. *Evaluation.*
   if $b' = b$ and A did not request $H^{-1}(y)$ in AG,
   return 1, otherwise 0.

4.2 Adversary

Let $\eta \in \mathbb{N}$ and the adversary being a ppt algorithm

$A = (AF(1^\eta, H, H^{-1}), AG(1^\eta, H, y : \{0,1\}^*))$

AF = finder

AG = guesser

H = encryption oracle

H^{-1} = decryption oracle

4.3 Advantage

$$\begin{aligned} Adv_{U,G}(\eta) &= 2 \cdot (Pr[\mathbb{E}^{S-CCA}_{A,S}(1^\eta) = 1] - \frac{1}{2}) \\ &= Pr[\mathbb{S}^{S-CCA}_{A,S}\langle b = 1\rangle(1^\eta) = 1] - Pr[\mathbb{S}^{S-CCA}_{A,S}\langle b = 0\rangle(1^\eta) = 1] \end{aligned}$$

5 Asymmetric encryption scheme

5.1 Definition

A asymmetric encryption scheme $\mathcal{S}$ is a tuple $\mathcal{S} = (X, \mathsf{Gen}(1^\eta), E, D)$ with security parameter $\eta \in \mathbb{N}$,

- $\mathsf{Gen}(1^\eta)$ is a ppt algorithm that outputs a pair of keys $(k, \hat{k})$. We call $\mathsf{Gen}(1^\eta)$ key generation algorithm.
  k is called public key, $\hat{k}$ is called private key. We denote the range of $\mathsf{Gen}(1^\eta)$ by K. We define $K_{pub} := \{k \mid (k, \hat{k}) \in K\}$ to be the set of all public keys, and $K_{priv} := \{\hat{k} \mid (k, \hat{k}) \in K\}$ to be the set of all private keys.
- $X = (X_k)_{k \in K_{pub}}$ a family of plaintext sets.
- a ppt encryption algorithm $E(x : \{0,1\}^*, k : K_{pub}) : \{0,1\}^*$,
- a deterministic polynomial-time decryption algorithm $D(y : \{0,1\}^*, \hat{k} : K_{priv}) : \{0,1\}^*$.

5.2 Security game

$\mathbb{S}(1^\eta) : \{0,1\}$

1. *Generate keys.*
   $(k, \hat{k}) \stackrel{\$}{\leftarrow} \mathsf{Gen}(1^\eta)$
2. *Find phase.*
   $(z_0, z_1) \stackrel{\$}{\leftarrow} AF(1^\eta, k)$
3. *Selection.*
   $b \stackrel{\$}{\leftarrow} \{0,1\}$; $y \stackrel{\$}{\leftarrow} E(z_b, k)$
4. *Guess phase.*
   $b' \stackrel{\$}{\leftarrow} AG(1^\eta, k, y)$
5. *Output.*
   return b'.

5.3 Advantage

$$
\begin{aligned}
Adv_{U,G}^{A-CPA}(\eta) &= 2 \cdot (Pr[\mathbb{E}_{A,S}^{A-CPA}(1^\eta) = 1] - \frac{1}{2}) \\
&= Pr[\mathbb{S}_{A,S}^{A-CPA}\langle b = 1\rangle(1^\eta) = 1] - Pr[\mathbb{S}_{A,S}^{S-CCA}\langle b = 0\rangle(1^\eta) = 1]
\end{aligned}
$$

6 RSA

6.1 Definition

Let $\eta \in \mathbb{N}$. The RSA (asymmetric) encryption scheme is the tuple

$$\mathcal{S}_{RSA} = (X, \mathsf{Gen}(1^\eta), E, D),$$

where

- $\mathsf{Gen}(1^\eta)$ selects two randomly chosen primes $p \neq q, p > 2, q > 2, |p|_2 = |q|_2 = \eta$, sets $n := p \cdot q$, $m := (p-1) \cdot (q-1)$ $[= \Phi(n)]$, chooses an element $e \in \mathbb{Z}_m^*$, computes $d = e^{-1} \mod m$, and outputs $((n,e),(n,d))$.
  We denote the range of $\mathsf{Gen}(1^\eta)$ by $K = \{((n,e),(n,d)) : n = p \cdot q$ for primes $p \neq q$, $e \cdot d \mod m = 1, m = \Phi(n)\}$.
- $X = \{X_{(n,e)}\}_{(n,e) \in K_{pub}}$ where $X_{(n,e)} = \mathbb{Z}_n$
- $E(x,(n,e)) = x^e \mod n, \quad (n,e) \in K_{pub}, x \in \mathbb{Z}_n$
- $D(y,(n,d)) = y^d \mod n, \quad (n,d) \in K_{priv}, y \in \mathbb{Z}_n$ $\diamond$

6.2 Security game

$\mathbb{E}(1^\eta) : \{0,1\}$

1. *Generate keys.*

   $((n,e),(n,d)) \xleftarrow{\$} \mathsf{Gen}(1^\eta)$
2. *Message selection.*

   $x \xleftarrow{\$} \mathbb{Z}_n$

   $y = x^e \mod n$
3. *Guess phase.*

   $x' \xleftarrow{\$} I(1^\eta,(n,e),y)$
4. *Evaluation.*

   if $x' = x$, return 1, otherwise 0.

6.3 Advantage / RSA-Assumption

$|Adv_{I,S}^{RSA}(\eta)| = Pr[\mathbb{E}_{I,S}^{RSA}(1^{\eta}) = 1]$

The function of the RSA encryption considered not invertable, because it is a trap-door function

7 ElGamal

7.1 Definition

ElGamal is CPA secure.

Let $\eta \in \mathbb{N}$. The ElGamal (asymmetric) encryption scheme based on GroupGen is the tuple $\mathcal{S}_{ElGamal} = (X, \mathsf{Gen}(1^\eta), E, D)$, where

Public key

- $\mathsf{Gen}(1^\eta)$ executes $\mathsf{GroupGen}(1^\eta)$ and obtains $(\mathcal{G}, n, g)$. Then, $\mathsf{Gen}(1^\eta)$ chooses $b \in \{0, \ldots, n-1\}$ uniformly at random and outputs $((\mathcal{G}, n, g, g^b), (\mathcal{G}, n, g, b))$. Private key
- $X = \{\mathcal{G}\}_{(\mathcal{G},n,g,h) \in K_{pub}}$. That is, the plaintexts are interpreted as elements of the group $\mathcal{G}$.
- $E(x : \mathcal{G}, (\mathcal{G}, n, g, h) : K_{pub})$:
  $a \xleftarrow{\$} \{0, \ldots, n-1\}$
  return $(g^a, x \cdot h^a)$.
- $D((y_0, y_1) : \mathcal{G} \times \mathcal{G}, (\mathcal{G}, n, g, b) : K_{priv})$:
  return $y_1 \cdot \left((y_0)^b\right)^{-1}$. ◇

inverse in $\mathcal{G}$

7.2 Advantage

TODO (DH-Assumption)

8 Hashes

8.1 Definition

Let $\eta \in \mathbb{N}$, $l : \mathbb{N} \to \mathbb{N}$ be a polynomial. A (cryptographic) hash function is a pair of the form $\mathcal{H} = (\mathsf{Gen}(1^\eta), h)$ where

- $\mathsf{Gen}(1^\eta)$ is a ppt algorithm that outputs a key k. We call $\mathsf{Gen}(1^\eta)$ key generation algorithm. We denote the range of $\mathsf{Gen}(1^\eta)$ by K. We also write $h_k(x)$.
- $h(k : K, x : \{0,1\}^*) : \{0,1\}^{l(\eta)}$ is a ppt algorithm.

The output of h is called a hash or a hash value of x.

Let $l' : \mathbb{N} \to \mathbb{N}$ be such that $l'(\eta) > l(\eta)$. If h_k is defined only for inputs $x \in \{0,1\}^{l'(\eta)}$, then we call $\mathcal{H} = (\mathsf{Gen}(1^\eta), h)$ a compression function. ◇

8.2 Security Game

$\mathbb{E}(1^\eta) : \{0,1\}$

1. *Choose index.*
   $k \xleftarrow{\$} \text{Gen}(1^\eta)$
2. *Find collision.*
   $(x_0, x_1) \xleftarrow{\$} A(1^\eta, k)$
3. *Evaluation.*
   if $h_k(x_0) = h_k(x_1)$ and $x_0 \neq x_1$
   return 1, otherwise 0.

8.3 Advantage

$Adv^{Coll}_{A,\mathbb{H}} = Pr[\mathbb{E}^{Coll}_{A,\mathbb{H}}(1^\eta) = 1]$.

9 MAC

9.1 Definition

Let $\eta \in \mathbb{N}$ and $l : \mathbb{N} \to \mathbb{N}$ be a polynomial.
A message authentication code (MAC) is a tuple of the form $\mathcal{M} = (\text{Gen}(1^\eta), T, V)$, where

- $\text{Gen}(1^\eta)$ is a ppt algorithm that outputs a key k. We call $\text{Gen}(1^\eta)$ key generation algorithm. We denote the range of $\text{Gen}(1^\eta)$ by K.
- T is a deterministic polynomial time tag-generation algorithm of the form $T(x : \{0,1\}^*, k : K) : \{0,1\}^{l(\eta)}$ that outputs a tag $t \in \{0,1\}^{l(\eta)}$.
- V is a polynomial time verification algorithm of the form $V(x \in \{0,1\}^*, t \in \{0,1\}^{l(\eta)}, k : K) : \{\texttt{valid}, \texttt{invalid}\}$

such that the following holds true:

$\forall x \in X, k \in K : V(x, T(x,k), k) = \texttt{valid}$.

one can generalize this definition to probabilistic tag-generation algorithms.
However, MACs are usually deterministic

9.2 Security game

$\mathbb{E}(1^\eta) : \{0,1\}$

1. *Choose key.*
   $k \xleftarrow{\$} \mathsf{Gen}(1^\eta)$ and $F = T(\cdot, k)$
2. *Compute a message-tag pair.*
   $(x,t) \xleftarrow{\$} A(1^\eta, F)$
3. *Evaluation.*
   if $V(x,t,k) = \texttt{valid}$ and A did not query x, then return 1, otherwise 0.

9.3 Advantage

$Adv^{MAC}_{A,\mathcal{M}}(\eta) = Pr[\mathbb{E}^{MAC}_{A,\mathcal{M}}(1^\eta) = 1]$