Cryptography

Academic Year 2024-2025

Homework 2

Michele Dinelli, ID 0001132338

November 18, 2024

Exercise 1.

Given G as a fixed pseudorandom generator with expansion factor ℓ and two algorithms Gen and Mac defined as:

- Gen on input 1^n outputs a binary string k drawn uniformly at random from $\{0,1\}^n$
- Mac on input $k \in \{0,1\}^n$ and $m \in \{0,1\}^{\ell(n)}$ draws at random $r \in \{0,1\}^{\ell(n)}$ and outputs the pair $\langle r, G(k) \oplus m \oplus r \rangle$

It is required to give a definition of the algorithm Vrfy such that MAC $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ is at least correct. It is also required to check if Π is eventually secure.

On the implementation and correctness of Vrfy.

- Vrfy is an algorithm that accepts three inputs: a key $k \in \{0,1\}^n$ a message $m \in \{0,1\}^{\ell(n)}$ and a tag t which consist of a pair namely $\langle r, G(k) \oplus m \oplus r \rangle$. It outputs a boolean b.

MAC Π is correct if and only if $Vrfy(k, m, Mac(k, m)) = 1$. Vrfy can be formalized as the following algorithm:

$\mathsf{Vrfy}(k, m, \langle r, t \rangle)$:

if $|m| \neq |r|$
  return 0
endif
$t' \leftarrow G(k) \oplus m \oplus r$;
return $t \stackrel{?}{=} t'$

Vrfy algorithm has to recompute the tag t and can't really use Mac algorithm because of the randomness of the variable r. We can say that using the Vrfy defined above will always return true for valid tags generated by Π so the resulting MAC $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ is correct.

On the security of MAC Π.

MAC Π is secure iif for every PPT adversary $\mathcal{A}$ exists a negligible function $\varepsilon \in \mathcal{NGL}$ such that

$$Pr(\mathsf{MacForge}_{\Pi,\mathcal{A}}(n) = 1) = \varepsilon(n) \tag{1}$$

where $\mathsf{MacForge}_{\Pi,\mathcal{A}}$ is defined and shown below:

$\mathsf{MacForge}_{\Pi,\mathcal{A}}(n)$:

$k \leftarrow Gen(1^n)$;
$(m, t) \leftarrow A(1^n, Mac_k(\cdot))$;
$\mathbb{Q} \leftarrow \{m \mid A \text{ queries } Mac_k(\cdot) \text{ on m}\}$;
return $(m \notin \mathbb{Q} \land Vrfy(k, m, t) = 1)$

MAC Π is not secure because it is possible to define and adversary $\mathcal{A}$ in the sense of the experiment $\mathsf{MacForge}$ which has non-negligible probability of success. The adversary $\mathcal{A}$ has given access to an oracle $\mathcal{O}$ for $Mac_k(\cdot)$ and can be built as follows:

$$
\begin{array}{l}
\underline{\mathcal{A}(1^n, Mac_k(\cdot)):} \\
m_0 \leftarrow \{0,1\}^{\ell(n)}; \\
\langle r, t \rangle \leftarrow Mac_k(m_0); \\
G(k) \leftarrow t \oplus m_0 \oplus r; \\
m_1 \leftarrow \{0,1\}^{\ell(n)}; \\
r' \leftarrow \{0,1\}^{\ell(n)}; \\
t' \leftarrow G(k) \oplus m_1 \oplus r'; \\
\textbf{return } \langle m_1, \langle r', t' \rangle \rangle
\end{array}
$$

$G(k)$ can be inferred and the random variable r does not introduce any randomness actually because it is an internal state of Mac but has to be exported in order to make Vrfy algorithm work. Given the adversary $\mathcal{A}$ it can be observed that

$$Pr(\mathsf{MacForge}_{\Pi,\mathcal{A}}(n) = 1) = 1 > \varepsilon \quad \forall \varepsilon \in \mathcal{NGL}$$

because the message m_1 has not been used by $\mathcal{A}$ for any oracle queries ($m_1 \notin \mathbb{Q} = \{m_0\}$) and $Vrfy(k, m, t) = Vrfy(k, m_1, \langle r, m_1, G(k) \oplus m_1 \oplus r) = 1$. So MAC Π can not be considered a secure authentication scheme.

Exercise 2.

Given Gen defined as above and F as a pseudorandom function it is required to consider the three following functions $H1, H2$ and $H3$ and to verify which one among $(Gen, H1)$, $(Gen, H2)$, $(\mathsf{Gen}, H3)$ are collision resistant hash-functions [1].

$$H_1^s(x \cdot y) = x \oplus y \oplus s \qquad H_2^s(x \cdot y) = F_s(x \oplus y) \qquad H_3^s(x \cdot y) = F_s(x) \oplus y$$

A hash function $\Pi = (Gen, H)$ is collision-resistant if and only if for every PPT adversary $\mathcal{A}$ exists a negligible function $\varepsilon \in \mathcal{NGL}$ such that

$$Pr(\mathsf{HashColl}_{\Pi,\mathcal{A}}(n) = 1) \leq \varepsilon(n) \tag{2}$$

where $\mathsf{HashColl}_{\mathcal{A},\Pi}$ is defined as follows

$$
\begin{array}{ll}
\multicolumn{2}{l}{\underline{\mathsf{HashColl}_{\Pi,\mathcal{A}}(n):}} \\
1: & s \leftarrow Gen(1^n); \\
2: & (x, y) \leftarrow A(s); \\
3: & \textbf{return } (x \neq y) \wedge (H(x) = H(y))
\end{array}
$$

- H_1^s is not a collision-resistant hash function because it is possible to define and adversary $\mathcal{A}$ in the sense of experiment $\mathsf{HashColl}$ with non-negligible probability of success.

$$
\begin{array}{l}
\underline{\mathcal{A}(s):} \\
x \leftarrow \{0,1\}^{|s|}; \\
y \leftarrow \{0,1\}^{|s|}; \quad /\!/ \text{ such that } x \neq y \\
\textbf{return } \langle (x \cdot y), (y \cdot x) \rangle
\end{array}
$$

[1] Here $x \cdot y$ is the concatenation of x and y

Given the definition of $\mathcal{A}$ and considering $\Pi = (Gen, H_1)$ it can be observed that

$$Pr(\mathsf{HashColl}_{\Pi,\mathcal{A}} = 1) = 1 > \varepsilon \quad \forall \varepsilon \in \mathcal{NGL}$$

because the two messages namely $m_1 = x \cdot y$ and $m_2 = y \cdot x$ have the same resulting hash $H_1^s(m_1) = H_1^s(m_2)$ but $m_1 \neq m_2$. More in general for any pair (x, y) and (x', y') if $x \oplus y = x' \oplus y'$ then $H_1^s(x \cdot y) = H_1^s(x' \cdot y')$.

- H_2^s is not a collision-resistant hash function because it is possible to define and adversary $\mathcal{A}$ in the sense of experiment $\mathsf{HashColl}$ with non-negligible probability of success.

$\mathcal{A}(s)$:

$x \leftarrow \{0,1\}^{|s|}$;
$y \leftarrow \{0,1\}^{|s|}$; // such that $x \neq y$
return $\langle (x \cdot y), (y \cdot x) \rangle$

Given the definition of $\mathcal{A}$ and considering $\Pi = (Gen, H_2)$ it can be observed that

$$Pr(\mathsf{HashColl}_{\Pi,\mathcal{A}} = 1) = 1 > \varepsilon \quad \forall \varepsilon \in \mathcal{NGL}$$

because the two messages namely $m_1 = x \cdot y$ and $m_2 = y \cdot x$ have the same resulting hash $H_2^s(m_1) = H_2^s(m_2)$ but $m_1 \neq m_2$. Although F is a pseudorandom function it still has to produce the same output for the same input if used with the same key. Exploiting the fact that $x \oplus 0 = x$ it is possible to produce two different messages that result in the same input for F. More in general for any pair (x, y) and (x', y') if $x \oplus y = x' \oplus y'$ then $H_2^s(x \cdot y) = H_2^s(x' \cdot y')$.

- H_3^s is not a collision-resistant hash function because it is possible to define and adversary $\mathcal{A}$ in the sense of experiment $\mathsf{HashColl}$ with non-negligible probability of success.

$\mathcal{A}(s)$:

$x \leftarrow \{0,1\}^{|s|}$;
$y \leftarrow F_s(x)$;
$x' \leftarrow \{0,1\}^{|s|}$;
$y' \leftarrow F_s(x')$;
return $\langle (x \cdot y), (x' \cdot y') \rangle$

Given the definition of $\mathcal{A}$ and considering $\Pi = (Gen, H_3)$ it can be observed that

$$Pr(\mathsf{HashColl}_{\Pi,\mathcal{A}} = 1) = 1 > \varepsilon \quad \forall \varepsilon \in \mathcal{NGL}$$

because the two messages namely $m_1 = x \cdot y$ and $m_2 = x' \cdot y'$ have the same resulting hash $H_3^s(m_1) = H_3^s(m_2)$ but $m_1 \neq m_2$. It is clear that when H_3^s is fed with $(x \cdot F_s(x))$ and then with $(x' \cdot F_s(x'))$ produce a collision in particular $0^{|s|}$ [2].

Exercise 3.

Given a hash function $\Pi = (\mathsf{Gen}, H)$ for messages of length $\ell(n)$ it is possible to formalize the notion of second pre-image resistance through the experiment $\mathsf{HashSec}$ defined as follows:

[2] Or more generally $0^{\ell(n)}$ where ℓ is a polynomial such that H_3^s returns a string of length $\ell(n)$ where n is the implicit parameter in s

$\mathsf{HashSec}_{\Pi,\mathcal{A}}(n)$:

$$
\begin{array}{ll}
1: & s \leftarrow Gen(1^n); \\
2: & x \leftarrow \{0,1\}^{\ell(n)}; \\
3: & y \leftarrow \mathcal{A}(s,x); \\
4: & \textbf{return } (x \neq y) \wedge (H^s(x) = H^s(y))
\end{array}
$$

Π is said to be second pre-image resistant if and only if for every PPT adversary $\mathcal{A}$ there is a negligible function $\varepsilon \in \mathcal{NGL}$ such that

$$Pr(\mathsf{HashSec}_{\Pi,\mathcal{A}}(1^n) = 1) = \varepsilon(n) \tag{3}$$

It is required to prove that collision resistance implies second pre-image resistance. About collision resistance the hypothesis is that for every PPT adversary $\mathcal{B}$ there is a negligible function $\varepsilon \in \mathcal{NGL}$ such that

$$Pr(\mathsf{HashColl}_{\Pi,\mathcal{B}}(1^n) = 1) = \varepsilon(n) \tag{4}$$

It is possible to proceed with a proof by reduction: it is assumed the existence of a PPT adversary $\mathcal{A}$ for Π that can find a collision in the sense of the experiment HashSec. Out of any successful adversary $\mathcal{A}$ we build and adversary $\mathcal{B}$ that uses $\mathcal{A}$ as a subroutine.

$$
\begin{gathered}
\forall B \in PPT.\neg Coll^{\mathsf{HashColl}}(B,\Pi) \Rightarrow \forall A \in PPT.\neg Coll^{\mathsf{HashSec}}(A,\Pi) \\
\Downarrow \\
\exists A \in \mathsf{PPT}.Coll^{\mathsf{HashSec}}(A,\Pi) \Rightarrow \exists B \in \mathsf{PPT}.Coll^{\mathsf{HashColl}}(B,\Pi)
\end{gathered}
$$

where $\mathcal{A}$ is an adversary for Π in the sense of the experiment HashSec and $\mathcal{B}$ is an adversary for Π in the sense of the experiment HashColl (alg. 1). It is possible to formalize the adversary $\mathcal{B}$ as follows:

$\mathcal{B}(s)$:

$$
\begin{array}{l}
x \leftarrow \{0,1\}^{\ell(n)}; \\
y \leftarrow \mathcal{A}(s,x); \\
\textbf{return } (x,y)
\end{array}
$$

If $\mathcal{A}$ succeeds (i.e. it finds $y \neq x$ such that $H^s(y) = H^s(x)$), then $\mathcal{B}$ succeeds too, thereby succeeding in the experiment HashColl. Since $\mathcal{B}$ succeeds using $\mathcal{A}$ as a subroutine it must have probability of succeeding equals to ε (eq. 3).

$$Pr(\mathsf{HashSec}_{\mathcal{A},\Pi}(1^n) = 1) = Pr(\mathsf{HashColl}_{\mathcal{B},\Pi}(1^n) = 1) = \varepsilon(n)$$

If ε is not negligible we would have a contradiction with eq. 4 because $\mathcal{B}$ would be constructed as a PPT adversary in the sense of experiment HashColl that has non negligible probability of finding a collision, so ε must be negligible. Hence if Π is collision resistance is also second pre-image resistance.