Exercises (from Katz and Lindell’s book)

11.6 Consider the following public-key encryption scheme. The public key is \((G, q, g, h)\) and the private key is \(x\), generated exactly as in the ElGamal encryption scheme. In order to encrypt a bit \(b\), the sender does the following:

(a) If \(b = 0\) then choose a uniform \(y \in \mathbb{Z}_q\) and compute \(c_1 := g^y\) and \(c_2 := h^y\). The ciphertext is \((c_1, c_2)\).

(b) If \(b = 1\) then choose independent uniform \(y, z \in \mathbb{Z}_q\), compute \(c_1 := g^y\) and \(c_2 := g^z\), and set the ciphertext equal to \((c_1, c_2)\).

Show that it is possible to decrypt efficiently given knowledge of \(x\). Prove that this encryption scheme is CPA-secure if the decisional Diffie-Hellman problem is hard relative to \(G\).

11.7 Consider the following variant of El Gamal encryption. Let \(p = 2q + 1\), let \(G\) be the group of squares modulo \(p\) (so \(G\) is a subgroup of \(\mathbb{Z}_p^*\) of order \(q\)), and let \(g\) be a generator of \(G\). The private key is \((G, g, q, x)\) and the public key is \((G, g, q, h)\), where \(h = g^x\) and \(x \in \mathbb{Z}_q\) is chosen uniformly. To encrypt a message \(m \in \mathbb{Z}_q\), choose a uniform \(r \in \mathbb{Z}_q\), compute \(c_1 := g^r \mod p\) and \(c_2 := h^r + m \mod p\), and let the ciphertext be \((c_1, c_2)\). Is this scheme CPA-secure? Prove your answer.

Hint for 11.6: Prove that if "not CPA-secure" then "DDH problem is efficiently solved".

12.1 Show that Construction 4.7 for constructing a variable-length MAC from any fixed-length MAC can also be used (with appropriate modifications) to construct a signature scheme for arbitrary-length messages from any signature scheme for messages of fixed length \(\ell(n) \geq n\).

More on back…
12.5 Another approach (besides hashing) that has been tried to construct secure RSA-based signatures is to \textit{encode} the message before applying the RSA permutation. Here the signer fixes a public encoding function $\text{enc} : \{0,1\}^\ell \rightarrow \mathbb{Z}_N^*$ as part of its public key, and the signature on a message $m$ is $\sigma := [\text{enc}(m)^d \mod N]$.  

(a) How is verification performed in encoded RSA?

(b) Discuss why appropriate choice of encoding function for $\ell \ll \|N\|$ prevents the “no-message attack” described in Section 12.4.1.  

(c) Show that encoded RSA is insecure if $\text{enc}(m) = 0x00m/0^\kappa/10$ (where $\kappa \overset{\text{def}}{=} \|N\|$, $\ell = |m| \overset{\text{def}}{=} 4\kappa/5$, and $m$ is not the all-0 message). Assume $e = 3$.

(d) Show that encoded RSA is insecure for $\text{enc}(m) = 0|m|0|m$ (where $\ell = |m| \overset{\text{def}}{=} (\|N\| - 1)/2$ and $m$ is not the all-0 message). Assume $e = 3$.

(e) Solve parts (c) and (d) for arbitrary $e$.

12.11 The Lamport scheme uses $2\ell$ values in the public key to sign messages of length $\ell$. Consider the variant in which the private key contains $2\ell$ values $x_1, \ldots, x_{2\ell}$ and the public key contains the values $y_1, \ldots, y_{2\ell}$ with $y_i := f(x_i)$. A message $m \in \{0,1\}^{\ell'}$ is mapped in a one-to-one fashion to a subset $S_m \subset \{1, \ldots, 2\ell\}$ of size $\ell$. To sign $m$, the signer reveals $\{x_i\}_{i \in S_m}$. Prove that this gives a one-time-secure signature scheme. What is the maximum message length $\ell'$ that this scheme supports?

4.7 Let $F$ be a pseudorandom function. Show that each of the following MACs is insecure, even if used to authenticate fixed-length messages. (In each case Gen outputs a uniform $k \in \{0,1\}^n$. Let $\langle i \rangle$ denote an $n/2$-bit encoding of the integer $i$.)

(a) To authenticate a message $m = m_1, \ldots, m_\ell$, where $m_i \in \{0,1\}^n$, compute $t := F_k(m_1) \oplus \cdots \oplus F_k(m_\ell)$.

(b) To authenticate a message $m = m_1, \ldots, m_\ell$, where $m_i \in \{0,1\}^{n/2}$, compute $t := F_k(\langle 1 \rangle m_1) \oplus \cdots \oplus F_k(\langle \ell \rangle m_\ell)$.

(c) To authenticate a message $m = m_1, \ldots, m_\ell$, where $m_i \in \{0,1\}^{n/2}$, choose uniform $r \leftarrow \{0,1\}^n$, compute $$t := F_k(r) \oplus F_k(\langle 1 \rangle m_1) \oplus \cdots \oplus F_k(\langle \ell \rangle m_\ell),$$ and let the tag be $\langle r, t \rangle$.
4.13 We explore what happens when the basic CBC-MAC construction is used with messages of different lengths.

(a) Say the sender and receiver do not agree on the message length in advance (and so Vrfy\textsubscript{k}(m, t) = 1 iff t \equiv \text{Mac}_{k}(m), regardless of the length of m), but the sender is careful to only authenticate messages of length 2n. Show that an adversary can forge a valid tag on a message of length 4n.

(b) Say the receiver only accepts 3-block messages (so Vrfy\textsubscript{k}(m, t) = 1 only if m has length 3n and t \equiv \text{Mac}_{k}(m)), but the sender authenticates messages of any length a multiple of n. Show that an adversary can forge a valid tag on a new message.

4.14 Prove that the following modifications of basic CBC-MAC do not yield a secure MAC (even for fixed-length messages):

(a) Mac outputs all blocks t\textsubscript{1}, \ldots, t\textsubscript{\ell}, rather than just t\textsubscript{\ell}. (Verification only checks whether t\textsubscript{\ell} is correct.)

(b) A random initial block is used each time a message is authenticated. That is, choose uniform t\textsubscript{0} \in \{0, 1\}\textsuperscript{n}, run basic CBC-MAC over the “message” t\textsubscript{0}, m\textsubscript{1}, \ldots, m\textsubscript{\ell}, and output the tag \langle t\textsubscript{0}, t\textsubscript{\ell} \rangle. Verification is done in the natural way.

MATH E M A T I C A  Q U E S T I O N S

Let \( N = 12801889219865986943874426789172837719929575398179139903346010225932249438875660672837312104315480979024966347267720662254924720490903440140409487830138442554051215639407252719582615491056895127372123401970340184655821416714383833567438594837829393436445708175846840391647287652219983832401360628720836954408208209 \) be an RSA public modulus ( \( e = N \) as in Cocks' variation.).

1) Without factoring \( N \), provide a message \( m \) that ends with “2017” in base 10 together with its RSA signature \( \sigma \). Show that \( \sigma \) is a valid signature.

2) Without factoring \( N \), check that the exponent \( e' = 999858280201913599008802868696830357098395840037288384624455770410649259059950052168890075728986418115945133344092917628768649110448940746235437111354648093 \) is also valid to verify signed messages. Show at least 5 examples.

3) Given \( e \) and \( e' \), factor \( N \). What is special about the factors of \( N \) ?

More on back…