Cryptography and Architectures for Computer Security

Exam Code: 095947 (old 090959), A.Y. 2014–2015, Semester: 2

Prof. G. Pelosi

July 1st, 2015 – Exam Session

Name: ..... Surname: .....

Student ID: ..... Signature: .....

Time: 2h:30'. Use of textbooks, notes, phones or Internet connected devices is not allowed. Prior to turn in your paper, write your name on any additional sheet and sign it.

Question 1 [3 pts]
Consider a TLS ciphersuite as a tuple $(\mathcal{A}_{kex}, \mathcal{A}_{auth}, \mathcal{A}_{sym}, \mathcal{A}_{hash})$

(a) Which choice between *(Diffie-Hellman-2048, RSA-1024, AES-128-CBC, SHA-2-256)* and *(Diffie-Hellman-Ephemeral-512, RSA-2048, AES-128-CBC, SHA-2-256)* is the one providing the highest security margin?

(b) During the TLS key exchange, is it possible for an active attacker to alter the value of the session nonce, setting it to an arbitrary value decided by him?
Does this action get detected before the end of the TLS handshake?

(c) Is there any difference between picking *(Diffie-Hellman-Ephemeral-2048, RSA-2048, None, SHA-2-256)* and *(Diffie-Hellman-2048, RSA-2048, None, SHA-2-256)* as a TLS ciphersuite?

Solution:

(a) Despite providing perfect forward secrecy, the second ciphersuite is weaker than the first one, as it is possible to break DHE-512 with a relatively small computational effort, thus deriving the session key. The first choice, despite employing a suboptimal choice for key lenghts (the RSA keylength voids the effort of the 2048 bit DH key)

(b) It is possible for an active attacker to tamper precisely with the value of the session nonce, however, as the final message in the TLS handshake includes the hash of all the handshake messages, such tampering will be detected on the server side as the hash computed locally will not match with the one computed by the client.

(c) No. The temporary session key which is needed to provide confidentiality on the transported data is not used, as there is no symmetric bulk encryption. (Both choices are valid ones in TLS, although, obviously, not among the ideal ones).

Remember: In the TLS jargon, Diffie-Hellman-Ephemeral (DHE) differs from the static Diffie-Hellman (DH) in the way that static Diffie-Hellman key exchanges always use the same Diffie-Hellman (half)keys.

When a key exchange uses Ephemeral Diffie-Hellman a temporary DH key is generated for every connection and thus the same shared key is never used twice. This enable *Perfect Forwards Secrecy*, which means that if the private key of the server gets leaked, past communication is still secure.

Question 2 [3 pts]
While reviewing an implementation of AES-128-CBC, you discover that it simply uses the last ciphertext block from the previously encrypted message as the `IV` value C_0 for encrypting the next message. The implementation's authors argue that as long as the `IV` of the very first message was chosen uniformly at random, all resulting subsequent ciphertext blocks will also be distributed uniformly at random, thus providing a secure solution[1].
Discuss the robustness of this CBC implementation with respect to a *Chosen Plaintext Attack*.

Solution:
The described implementation cannot be recommended as a secure one because it enables an adversary (who record every data transmission) with the ability to verify guesses on the plaintext blocks of the encrypted communication.

Given a plaintext split up in series of a blocks $P_1, P_2, \dots P_L$, the CBC mode of operation applies the symmetric key cipher of choice (say $\mathbb{E}_k(...)$) to compute the ciphertext blocks in the following way: $C_0 = \texttt{IV}$, $C_i = \mathbb{E}_k(P_i \oplus C_{i-1})$, with $1 \leq i \leq L$.

The usual practice of applying a CBC mode of operation is to choose a new, **unpredictable** random `IV` for every message that is encrypted.

The implementation mentioned above chooses all but the initial `IV` by setting it equal to the final ciphertext block of the preceding encrypted message. This enable a passive adversary to mount a chosen-plaintext attack (CPA) to verify his guess as to whether a particular plaintext block has a particular value.

Say an adversary who has observed the ciphertext $C_0, \dots, C_{j-1}, C_j, \dots C_L$ wants to determine whether plaintext block P_j, with $1 \leq j < L$, is equal to some string P^*.

Note that the adversary knows the Initialization Vector that will be used when encrypting the next message, that is C_L.

The adversary causes the sender to encrypt a message M whose initial block P'_1 is equal to $P'_1 = C_{j-1} \oplus C_L \oplus P^*$.
The first ciphertext block C'_1 will be computed as: $C'_1 = \mathbb{E}_k(P'_1 \oplus C_L) = \mathbb{E}_k(C_{j-1} \oplus P^*)$.

However, the adversary also know that $C_j = \mathbb{E}_k(P_j \oplus C_{j-1})$.
This implies that $C'_1 = C_j$ iff $P_j = P^*$.

In this way, an adversary can verify a guess P^* for the value of any plaintext block P_j.

Note that, if the adversary knows that P_j *is one of two possible values* then the adversary can determine the actual value of P_j by executing the above attack a single time.

Similarly, if the adversary knows that P_j is one out of N possible values, then by repeating the above attack $N/2$ times (on average) he can determine the actual value of P_j.

[1] The initial `IV` used by SSL 3.0 (TLS 1.0) was a (pseudo)random string generated and shared during the initial handshake phase, subsequent `IV`s were chosen following the deterministic pattern previously described.

Question 3 [5 pts]
Consider a substitution-permutation block cipher with a 128-bit block and a 256-bit key, employing 32 identical 4×4 bit S-boxes for the substitution layer. The best (i.e., highest) linear bias for the S-Box is $\varepsilon=\frac{1}{8}$, while the highest differential probability is $\frac{1}{4}$. The permutation layer is built so that the 4 output bits of each S-Box are employed as inputs to four different boxes in the subsequent layer. The diffusion of the cipher is such that given a change in an S-box input at round i, 4 s-boxes will be involved at round $i+1$, 16 at round $i+2$ and all of them from round $i+3$ onwards. The cipher round acts on the state performing with the substitution layer, the permutation layer and the round key addition, in this order. A single key extra key addition is present before the first round takes place.

(a) Compute the value of a conservative estimate of the linear bias useful for retrieving the last round key for the described block cipher assuming it is 4 rounds long, and the amount of plaintext-ciphertext pairs available. Is it possibly broken by linear cryptanalysis?

(b) Compute the value of the differential probability useful for retrieving the last round key for the described block cipher assuming it is 4 rounds long, and the amount of plaintext-ciphertext pairs available. Is it possibly broken by differential cryptanalysis?

(c) Keeping the same round structure, and the same key length, is it possible to render the block linear and differential cryptanalysis immune? If yes, describe what should be tuned and to which extent.

Solution:

To provide a conservative estimate of the linear bias, assume that the best linear bias can always be employed to approximate an S-Box. Consequentially, to build a linear approximation able to retrieve the last round the best achievable linear bias piles up 1 S-Box from round 1, 4 from round 2, and 16 from round 3, for a total of 21 active S-Boxes, considering for all of them the best linear bias $\varepsilon=\frac{1}{8}$. The bias for the full approximation is thus $2^{21-1}(\frac{1}{8})^{21} = 2^{20}\frac{1}{2^{62}} = \frac{1}{2^{42}}$, requiring $\approx 2^{84}$ plaintext-ciphertext pairs to be exploited. Since a bruteforce over the full keyspace requires 2^{256}, the cipher is possibly broken by linear cryptanalysis.

Similarly to the linear cryptanalysis case, we will be exploiting the best differential probability for all the S-Boxes. The number of active S-Boxes does not change with respect to the previous analysis, thus we obtain a differential probability for the full approximation which is $(\frac{1}{4})^{21} = \frac{1}{2^{42}}$, requiring 2^{42} plaintext-ciphertext pairs to be exploited. The cipher is quite likely to be broken by differential cryptanalysis.

Given the constraints, the cipher can be tweaked to be immune to both cryptanalysis raising the number of rounds. In detail, adding a round after the fourth activates 32 S-boxes per added round. To provide linear cryptanalysis immunity, consider that adding round will provide a multiplicative factor of $2(\frac{1}{8})^{32}) = \frac{1}{2^{64}}$ on the linear bias over the current one of the cipher, raising the number of required ciphertexts by a factor $(2^{64}) = 2^{128}$. Two extra rounds are thus sufficient to provide linear cryptanalysis immunity (2^{42+256} ptx-ctx pairs required). Concerning differential cryptanalysis, a round provides a multiplicative factor on the differential probability equal to $(\frac{1}{8})^{32} = 2^{96}$. As a consequence, the differential probability for r rounds is $2^{42} \cdot r2^{96}$. To achieve differential cryptanalysis immunity, it should hold that $2^{42} \cdot (r-4)2^{96} \geq 2^{256}$, thus $r \geq 7$, i.e. three more rounds should be added.

Question 4 [6 pts]

(a) Name two advantages of using cyclic groups of *prime order* in cryptographic schemes that rely on the difficulty of the Discrete Logarithm Problem or the Diffie-Hellman Problem.

(b) Consider the cyclic group $(\mathbb{Z}_{169}^*, \cdot)$.

- How many generators are there?
- Determine one generator g of $\mathbb{Z}_{169}^*$ and exhibit at least one generator for largest (proper) subgroup of $\mathbb{Z}_{169}^*$.
- Denote as $G{=}\langle 40\rangle$ the largest prime subgroup of $(\mathbb{Z}_{169}^*, \cdot)$ and compute the discrete logarithm $x \equiv_{|G|} \log_{40}^{\mathsf{D}} 14$, applying the Baby-Step Giant-Step algorithm.

Solution:

(a) see lectures...

(b)

- $|\mathbb{Z}_{169}^*| = \varphi(169) = \varphi(13^2) = 156 = 2^2 \cdot 3 \cdot 13$
  Number of generators: $\varphi(|\mathbb{Z}_{169}^*|){=}\varphi(156){=}48$
- There are as many subgroups as the number of factors of $|\mathbb{Z}_{169}^*| = 156$, that is 10. Among these 8 are proper subgroups with cardinality 2,3,4,12,13,26,52,78, respectively. The remaining two subgroups are the one including only the neutral element and the group itself.
  We test $g = 2$ to be a generator as follows:

$$
\begin{aligned}
&g^2 \equiv_{169} 4 \not\equiv_{169} 1,\\
&g^3 \equiv_{169} 8 \not\equiv_{169} 1,\\
&g^4 \equiv_{169} 16 \not\equiv_{169} 1,\\
&g^{12} \equiv_{169} 4096 \equiv_{169} 40 \not\equiv_{169} 1,\\
&g^{13} \equiv_{169} 40 \cdot 2 \equiv_{169} 80 \not\equiv_{169} 1,\\
&g^{13\cdot 2} \equiv_{169} 80^2 \equiv_{169} 147 \not\equiv_{169} 1,\\
&g^{13\cdot 4} \equiv_{169} 80^4 \equiv_{169} 146 \not\equiv_{169} 1,\\
&g^{13\cdot 6} \equiv_{169} 80^6 \equiv_{169} 168 \not\equiv_{169} 1,
\end{aligned}
$$

  $g = 2$ is a generator.
- The generator of the largest proper subgroup is $g_1 = g^{\frac{156}{13\cdot 6}} \equiv_{169} 2^2 \equiv_{169} 4$.
- The largest prime subgroup G has cardinality $|G| = 13$.
  $g{=}40$, $m = \left\lceil \sqrt{|G|} \right\rceil {=}4$, $g^{-m} \equiv_{169} 40^{-4} \equiv_{169} 40^{13-4} \equiv_{169} 40^9 \equiv_{169} 14$

  assuming $x = i \cdot m + j$, with $0 \le i, j \le m$,
  we have: $g^x \equiv_{169} 14 \Leftrightarrow g^j \equiv_{169} 14(g^{-m})^i$

  Baby-Steps

j:	0	1	2	3	4
g^j:	1	40	79	118	157

  Giant-Steps

i:	0	1	2	3	4
$14(g^{-m})^i$:	14	27	40	...	...

  $x \equiv_{13} 2 \cdot 4 + 1 \equiv_{13} 9$

Question 5 [5 pts]
Consider an elliptic curve cryptosystem defined over the elliptic curve $\mathbb{E}(\mathbb{Z}_{11})$ with equation $y^2 = x^3 + 1$ over $\mathbb{Z}_{11}$.

(a) What is the order of the additive group $(\mathbb{E}(\mathbb{Z}_{11}),\ +)$?

(b) What is the sum of the points (2, 3) and (5, 4)?

(c) Describe the encryption and decryption functions of the Elliptic Curve ElGamal cryptosystem.

Solution:

(a) $|\mathbb{E}(\mathbb{Z}_{11})|$=12

x, y:	0	1	2	3	4	5	6	7	8	9	10
$x^3 + 1$ mod 11:	1	2	9	6	10	5	8	3	7	4	0
y^2 mod 11:	0	1	4	9	5	3	3	5	9	4	1

Punti sulla curva	
(0, 1)	(0, 10)
(2, 3)	(2, 8)
(5, 4)	(5, 7)
(3, 5)	(3, 6)
(9, 2)	(9, 9)
(10, 0)	$\mathcal{O}$

(b) $\mathtt{S}(x_\mathtt{S}, y_\mathtt{S}) = \mathtt{P}(x_\mathtt{P}, y_\mathtt{P}) + \mathtt{Q}(x_\mathtt{Q}, y_\mathtt{Q})$=(2, 3)+(5, 4)
$\lambda \equiv_{11} (y_\mathtt{Q} - y_\mathtt{P})(x_\mathtt{Q} - x_\mathtt{P})^{-1} \equiv_{11} (4-3)(5-2)^{-1} \equiv_{11} 4$
$x_\mathtt{S} = \lambda^2 - x_\mathtt{P} - x_\mathtt{Q} \equiv_{11} 16-2-5 \equiv_{11} 9$
$y_\mathtt{S} = \lambda(x_\mathtt{P} - x_\mathtt{S}) - y_\mathtt{P} \equiv_{11} 4(2-9)-3 \equiv_{11} 2$
$\mathtt{S}(x_\mathtt{S}, y_\mathtt{S})$=(9, 2)

(c) see lectures ...

Question 6 [8 pts]
Consider the RSA modulus $n = p \cdot q$=899

(a) Apply the Pollard's $\mathtt{P}-1$ factorization method to compute the two factors p and q (assuming $p<q$). Consider the factor p being B-power smooth, with B=6, while the factor q is not.[2]

(b) Apply the Miller-Rabin primality test to the factor p of the public RSA modulus, assuming as witness either a=3 or b=5

(c) Given the modulus factorization found as answer to (a), pick the value of an admissible secret exponent d among $d_1 = 3,\ d_2 = 35,\ d_3 = 121$, explaining the reasons of your choice.

[2] To solve the remaining parts of the exercise you can apply a trivial division algorithm as a back up factoring strategy

(d) Sign the message $m{=}100_{\text{decimal}}{\in}\mathbb{Z}_n$ (provided without any padding scheme) through applying the CRT. Describe each step of the procedure.

Solution:

(a) $a = 2^{B!} \bmod n = 2^{2\cdot3\cdot4\cdot5\cdot6} \bmod 899 \equiv_{899} (2^{24})^{5\cdot6} \equiv_{899} 78^{5\cdot6} \equiv_{899} 807^6 \equiv_{899} 342$
$\mathbf{gcd(a-1, n)}$=gcd(341, 899)= gcd(217, 341)=gcd(124, 217)= gcd(93, 124)=gcd(31, 93)=**31**
$n/31{=}29$
$p = 29,\ q = 31$

(b) ...

(c) $d{=}121$ (... not required: $e \equiv_{\varphi(n)} d^{-1} \equiv_{840} 121^{191} \equiv_{840} 361$)

(d)

$$s = m^{d \bmod \varphi(n)} \bmod n \Leftrightarrow \begin{cases} s = s_p \bmod p \\ s = s_q \bmod q \end{cases} \text{ where:}$$

$s_p = m^{d \bmod \varphi(p)} \bmod p \equiv_{29} 100^{121 \bmod 28} \equiv_{29} 13^9 \equiv_{29} 5$ and
$s_q = m^{d \bmod \varphi(q)} \bmod q \equiv_{31} 100^{121 \bmod 30} \equiv_{31} 7^1 \equiv_{31} 7$

$s \equiv_n \left(M_p \cdot M_p^{'} \cdot s_p + M_q \cdot M_q^{'} \cdot s_q\right) \bmod n$
$M_p = q = 31,\ M_p^{'} = q^{-1} \bmod p = 31^{-1} \bmod 29 \equiv_{29} 2^{-1} \equiv_{29} 15$
$M_q = p = 29,\ M_q^{'} = p^{-1} \bmod q = 29^{-1} \bmod 31 \equiv_{31} (-2)^{-1} \equiv_{31} -16 \equiv_{31} 15$

$$\Rightarrow s \equiv_{899} (31 \cdot 15 \cdot 5 + 29 \cdot 15 \cdot 7) \equiv_{899} 5370 \equiv_{899} 875$$

Question 7 [4 pts]
Assume to work into the Montgomery domain: $(\mathbb{Z}_N, +, \times)$, $N = 15$.
Compute the Montgomery multiplication $C{=}A{\times}B \bmod N$, where $A{=}8_{\text{decimal}}$ and $B{=}11_{\text{decimal}}$ are binary encoded values in the Montgomery domain. Show every step of the procedure.

Solution:
$\lceil \log_2 N \rceil = 4,\ R = 2^4 = 16,$
$\gcd(R, N) = RR' - NN' = 1 \Leftrightarrow R(1) - N(1) = 1,$
$R' \stackrel{\text{def}}{=} R^{-1} \bmod N = 1 \bmod 15 = 1,$
$N' \stackrel{\text{def}}{=} N^{-1} \bmod R = 1 \bmod 16 = 1;\ N'_0 = 1$

$B = \langle B_3 B_2 B_1 B_0 \rangle = \langle 1011 \rangle_2$
$A = \langle A_3 A_2 A_1 A_0 \rangle = \langle 1000 \rangle_2$

0000	+
0000	$A_0B = 0 \cdot \langle 1011 \rangle_2$
0000	+
0000	$t\,N = (N_0' x_0)\,N = 0$
0000	perform a right-shift of 1 bit
$\vdots$	+
$\vdots$	$A_iB = 0 \cdot \langle 1011 \rangle_2$, $\mathbf{i = 1, 2}$
$\vdots$	+
$\vdots$	$t\,N = (N_0' x_0)\,N = 0$, $\mathbf{i = 1, 2}$
0000	perform a right-shift of 1 bit
0000	+
1011	$A_3B = 1 \cdot \langle 1011 \rangle_2$
1011	+
1111	$t\,N = (n_0' x_0)\,N = 1$
11010	perform a right-shift of 1 bit
1101	

$C = \langle 1101 \rangle_2 = 13 < N \Rightarrow C = 8 \times 11 = 13 \in (\mathbb{Z}_N, +, \times)$

Validation:

$C{=}8 \times 11 \stackrel{\text{def}}{=} 8 \cdot 11 \cdot R^{-1} \bmod N{=}8 \cdot 11 \cdot 1 \bmod 15{=}{-}2 \bmod 15{\equiv_{15}}13.$