CHALMERS — GÖTEBORGS UNIVERSITET

# EXAM IN CRYPTOGRAPHY

**TDA352 (Chalmers) - DIT250 (GU)**

**18 January 2019, 08:30 – 12.30**

Tillåtna hjälpmedel: Typgodkänd räknare. Annan minnestömd räknare får användas efter godkännande av kursansvarig vid dennes besök i skrivsalen.
No extra material is allowed during the exam except of pens and a simple calculator (with cleared memory). No smartphones or other electronic devices are allowed.
Answers must be given in *English* and should be clearly justified.

**Teacher/Examiner:** Katerina Mitrokotsa
**Questions during exam:** Katerina Mitrokotsa, phone 031 772 1040

The exam is divided in four main topics and the total number of points is 50 (plus 6 *bonus points*).
The grades are:

| | | | |
|---|---|---|---|
| CTH Grades: | 22-30 $\rightarrow$ 3 | 31-39 $\rightarrow$ 4 | 40-50$\rightarrow$5 |
| GU Grades: | 22-39 $\rightarrow$ G | 40-50 $\rightarrow$ VG | |

**Good luck!**

# 1 Symmetric Ciphers (16 p)

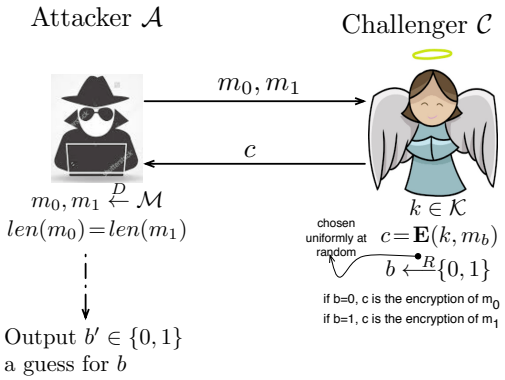
(a) Let us consider the following symmetric cipher:

$$E(k,m_0) = E(k,m_{00}||m_{01}) = m_{00}||E(k \oplus m_{01})$$

where $m_{00}$ and $m_{01}$ denote the first and second bit of a message $m_0$. Prove that this symmetric cipher is not semantically secure. (4 p)

*Hint:* Use a security game and describe a successful strategy of the attacker.

**Solution:**




According to the definition a cipher (**E**,**D**) is **semantically secure** (with one time key) if for any 'efficient' adversary, it holds:

$$P(b' = b) < \frac{1}{2} + negligible$$

We need to show that $\mathcal{A}$ can win the security game *(described above)*:

Let $\mathcal{A}$ choose:

$m_0 = m_{00}||m_{01} = 0||0$ and $m_1 = m_{10}||m_{11} = 1||1$

If $\mathcal{C}$ chooses $\mathbf{b = 0}$, then $m_b = m_0$ and $c = m_{00}||(k \oplus m_{01}) = 0||c_0 \Rightarrow c = 0||c_0$ then $b' = 0$

If $\mathcal{C}$ chooses $\mathbf{b = 1}$, then $m_b = m_1$ and $c = m_{10}||(k \oplus m_{11}) = 1||c_1 \Rightarrow c = 1||c_1$ then $b' = 1$

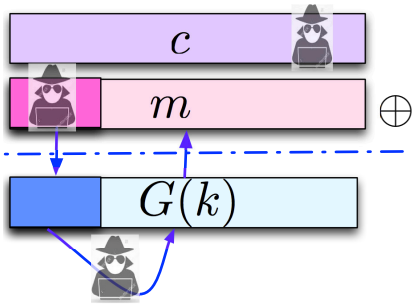
$\mathcal{A}$ can output $\mathbf{b'} = \{$first bit of $\mathbf{c}\}$ as a guess for $b$.

Then, we have:

$|\mathbf{P(W_0)} - \mathbf{P(W_1)}| = |\mathbf{1 - 0}| = \mathbf{1}$

(b) Let us consider that $\mathcal{G} : \mathcal{K} \rightarrow \{0,1\}^n$ is a predictable PRG. Describe an attack that can be performed against a stream cipher that uses the predictable PRG $\mathcal{G}$. (2 p)

**Solution:** Let us consider that $\mathcal{G}$ is predictable. Let $\mathcal{A}$ (adversary) be the algorithm that can efficiently compute $\mathcal{G}(s)|_{i+1}$ given $\mathcal{G}(s)|_{1,\ldots,i}$




If f $\mathcal{A}$ knows the ciphertext and the beginning of the plaintext ((*e.g.*, a header) then he can recover the first bits of the PRG as indicated in the figure. Often the headers of some

encrypted messages in communication protocols are standard and well known. For instance in the SMTP protocol every message starts with the message "To:". Thus, if the adversary has the ciphertext and knows what are the first bits of the plaintext (i.e., headers). He can recover the first bits of the PRG used. Since we have the assumption that the PRG is predicable this implies that given the first bits of a PRG we can predict the last. Thus, the adversary can recover all the bits of the PRG used *e.g.,* $G(k)$ and thus is able to recover the plaintext *i.e.,* $G(k) \oplus c = m$

(c) Consider that the OTP is used to encrypt two different messages $m_1$ and $m_2$ and the corresponding cipher texts are $c_1$ and $c_2$ correspondingly. The same key $k$ is used to encrypt both messages. Describe an attack (two time pad) that could be performed in order to recover the plaintexts $m_1$ and $m_2$.

*Hint:* Consider that the messages $m_1$ and $m_2$ represent messages (in binary form) in a language (*e.g.,* using ASCII code of each character). Give a concrete example explaining how the attack could be successful. (2 p).

**Solution:** Lets assume that the same key is used to encrypt two messages:

$c_1 = m_1 \oplus k$ and $c_2 = m_2 \oplus k$.

Then we have:

$$\begin{aligned} c_1 \oplus c_2 &= (m_1 \oplus k) \oplus (m_2 \oplus k) \\ &= (m_1 \oplus m_2) \oplus (k \oplus k) \\ &= m_1 \oplus m_2 \end{aligned}$$

If we have $m_1 \oplus m_2$ we can easily recover $m_1$ and $m_2$ More precisely, we can recover $m_1$ and $m_2$ due to the redundancy in English language, since not all combinations of letters are likely or possible.

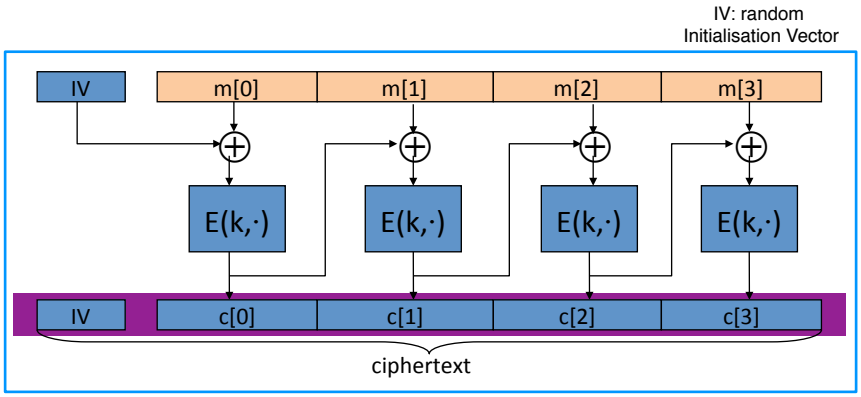
Let us consider as example that we have the following language: $L = \{00100, 10011, 11100, 10100\}$

If we have $m_1 \oplus m_2 = 10111$ then we can deduce $m_1 = 00100$ and $m_2 = 10011$, since only by XORing $m_1$ and $m_2$ we get $m_1 \oplus m_2$

(d) Describe how encryption and decryption works in CBC (Cipher Block Chaining) mode with IV for block ciphers. (1 p)

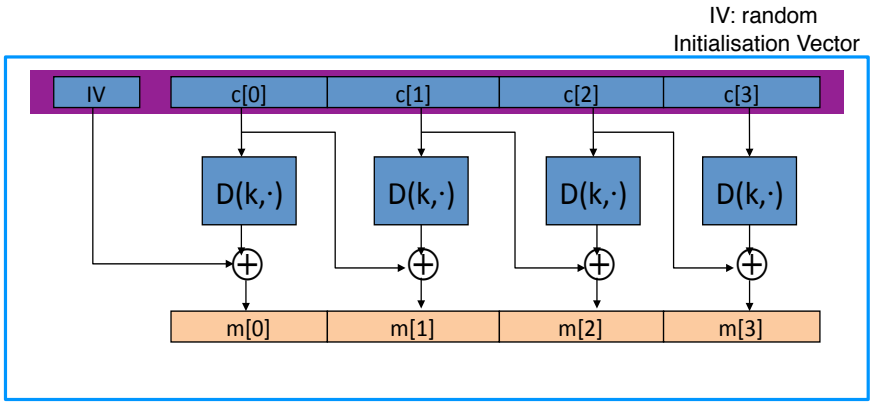
Let $(\mathbf{E}, \mathbf{D})$ be a block cipher.

The CBC block cipher is defined as follows. $\mathbf{E}_{\mathsf{CBC}}(k, m)$: choose a **random** $IV \in \{0,1\}^{nt}$ and do:

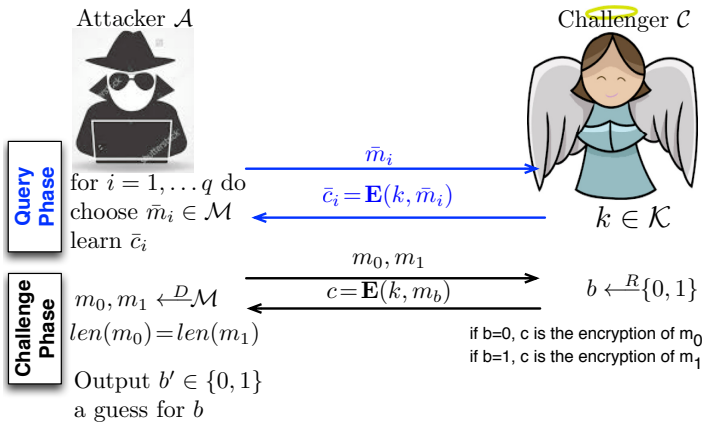



- Each ciphertext-block is **chained** and XOR-ed to the next plaintext block.
- The ciphertext is longer than the plaintext due to the IV.

For the CBC block cipher the decryption is defined as follows:




(e) *Bonus points: Give the definition of chosen-plaintext attacks (IND-CPA) for secret key encryption using a security game. (3 p)*



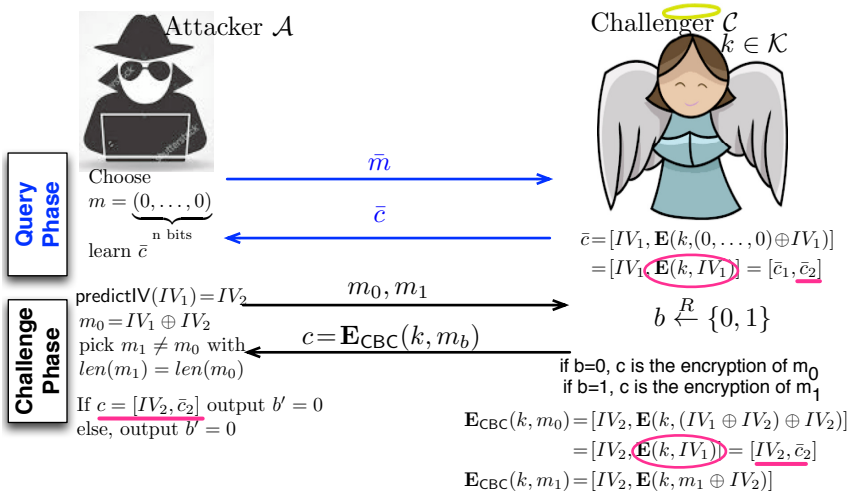

**Solution:**

Definition: A cipher (**E**,**D**) is secure under CPA if for any PPT adversary, it holds that:

$$P(b' = b) < \frac{1}{2} + negligible$$

(f) Show that CBC (Cipher Block Chaining) mode for block ciphers is not secure against Chosen Plaintext attacks (CPA) when the IV is predictable. (4 p)

*Hint:* Use in your description a security game and a successful strategy of the attacker in the case he is able to predict the IV. **Solution:**




Let $W_0$ be the event that $\mathcal{C}$ chooses $b = 0$, and $\mathcal{A}$ outputs $b' = 0$.
Let $W_1$ be the event that $\mathcal{C}$ chooses $b = 1$, and $\mathcal{A}$ outputs $b' = 0$.

Then, it holds: $|\mathbf{P}(\mathbf{W_0}) - \mathbf{P}(\mathbf{W_1})| = |\mathbf{1} - \mathbf{0}| = \mathbf{1}$

## 2 Public Key Encryption (12 p)

(a) Describe how the textbook RSA encryption works (1 p)

*Hint:* Describe the algorithms with their corresponding input and output.

**Solution:**

- $\mathbf{KeyGen}(\lambda) \rightarrow (\mathbf{pk}, \mathbf{sk})$
  - (a) generate two distinct $\lambda$-bit primes $p$ and $q$, compute $N = pq$ and $\Phi(N)$. $\Phi$ denotes Euler's phi (or Totient) function. It holds: $\Phi(N) = \Phi(pq) = (p-1)(q-1)$.
  - (b) choose an integer $e \xleftarrow{R} \mathbb{Z}_{\Phi(N)}$ such that $GCD(e, \Phi(N)) = 1$ and compute its (modular) inverse $d = e^{-1} \mod \Phi(N)$.
  - (c) set: $\mathbf{pk} = (N, d)$ and $\mathbf{sk} = (N, e)$
- $\mathbf{Enc}(\mathbf{pk}, \mathbf{m}) \rightarrow \mathbf{c}$ : *compute* $c = m^e \mod N$
- $\mathbf{Dec}(\mathbf{sk}, \mathbf{m}) \rightarrow \mathbf{m}$ : *compute* $m = c^d \mod N$

(b) Consider an RSA system with modulus $N = pq$, public key $e$ and private key $d$. Show that if an adversary finds out $\Phi(N) = (p-1)(q-1)$, she can easily factorise N and thus break the encryption. (3 p)

**Solution:** Recall that:

$$\Phi(N) = (p-1)(q-1) = pq - p - q + 1 = N - p - q + 1$$

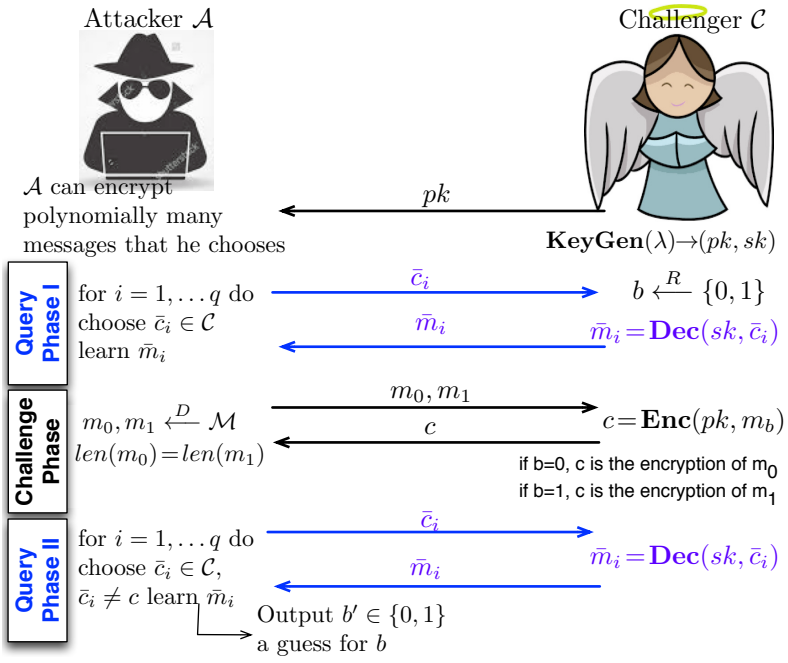
So if the Adversary knows $\Phi(N)$, he also knows

$$p + q = N + 1 - \Phi(N).$$

But if you know both $p \cdot q = N$ and $p + q = \alpha$, it is easy to compute $p$ and $q$: using $q = N/p$. You know $p + N/p = \alpha$, which gives the ordinary second-degree equation $p^2 - \alpha p + N = 0$ to solve for $p$.

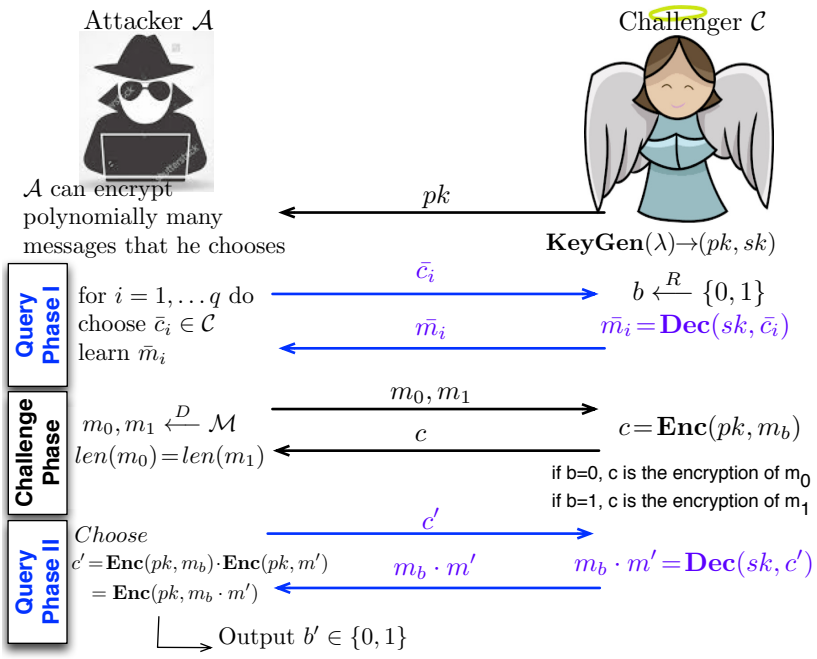
(c) Define the IND-CCA security game (indistinguishability chosen ciphertext attacks) and show that the textbook RSA encryption scheme is not secure under IND-CCA. (5 p)

**Solution:** The IND-CCA security game is defined as follows: **Definition:** A public key



cipher (**KeyGen**, **Enc**, **Dec**) is secure under CCA if for any 'efficient' adversary it holds: $P(b' = b) < \frac{1}{2} + \text{negligible}$

Textbook RSA has the homomorphic property and thus is susceptible to CCA attacks. More precisely, the following game can be performed:




(d) We consider double RSA encryption using a common modulus $N$ and two public keys $e_1$ and $e_2$ with corresponding private keys. Thus, a message $m$ is encrypted first using RSA encryption with the key $e_1$; the result is encrypted again using key $e_2$. Explain why this approach does not increase security. (3 p)

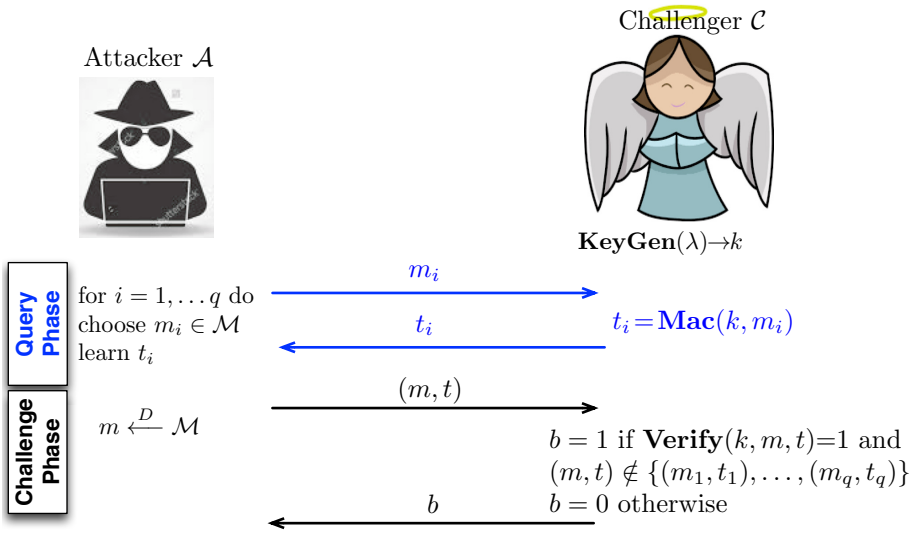
**Solution:** The general argument against double encryption is that it is subject to the meet-in-the-middle attack, which has time complexity similar to that of a single brute force attack. In the particular case of RSA encryption, double encryption is also meaningless, since the double encryption is equivalent to the single RSA encryption with public key $e_1 e_2$ and private key $d_1 d_2$. It is easy to verify this since it holds $(m^{e_1} \pmod N)^{e_2} \pmod N = m^{e_1 e_2} \pmod N$

## 3 Data Integrity (15 p)

(a) *Bonus points: How do we define a secure MAC (message authentication code)? (3 p).*

*Hint: Give the security game and formal definition.*

**Solution:**




**Definition:** A message authentication code (**KeyGen, MAC, Verify**) is secure, if for any 'efficient' adversary $\mathcal{A}$ it holds:

$$P(\text{Challenger outputs } 1) \text{ is negligible}$$

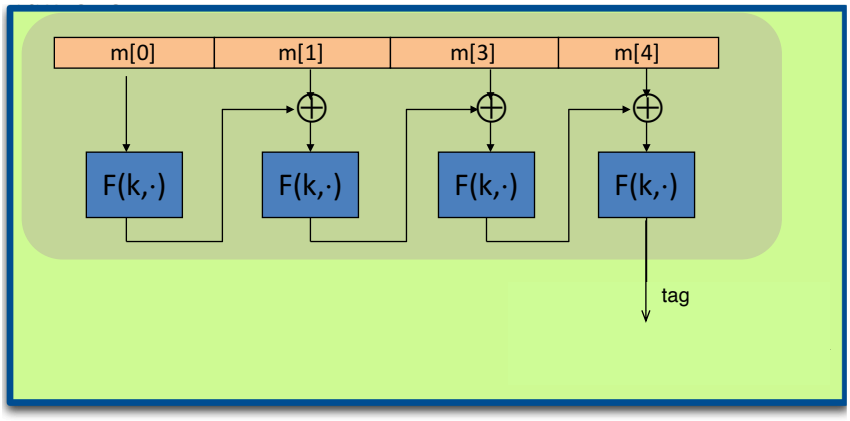
(b) Describe how raw CBC-MAC works. (2 p).

**Solution:**

Let $\mathbf{F} : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ be a block cipher (PRP) where $\mathcal{M} = \{0,1\}^n$

then the CBC-MAC is defined as follows:

$\mathbf{F}_{\mathsf{CBC}} : \mathcal{K} \times \mathcal{M}' \to \mathcal{M}$ where $\mathcal{M}' = \{0,1\}^{\ell n}$



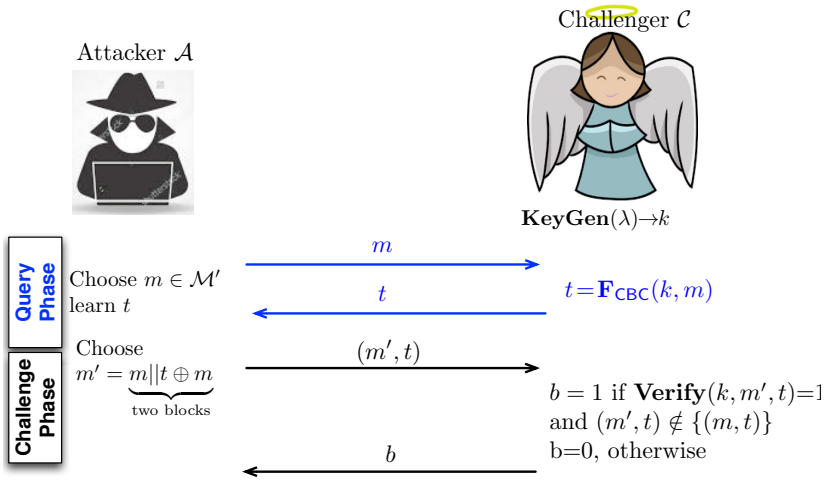

(c) Show that raw CBC-MAC is insecure. (5 p).

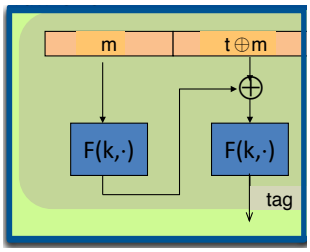
*Hint:* Describe an existential forgery. Use a security game and a successful strategy of the attacker. Consider that in the challenge phase, a message with length two blocks is used.

**Solution:**

The attacker may choose the message $m' = m||(t \oplus m)$

Then, it holds:




$$\mathbf{F}_{\mathsf{CBC}}(k,m') = \mathbf{F}_{\mathsf{CBC}}(k,\underbrace{\mathbf{F}_{\mathsf{CBC}}(\mathbf{k},\mathbf{m})}_{t} \oplus (t \oplus m)) = \mathbf{F}_{\mathsf{CBC}}(k,t \oplus (t \oplus m)) = \mathbf{F}_{\mathsf{CBC}}(k,m) = t$$

Thus, $(m',t)$ is a valid pair (message, tag). So we have: $\mathbf{Verify}(\mathbf{k},\mathbf{m}',\mathbf{t}) = \mathbf{1}$ and $(m',t) \notin \{(m,t)\}$)

So it holds: $\mathbf{P}(\text{Challenger outputs } 1) = \mathbf{1}$

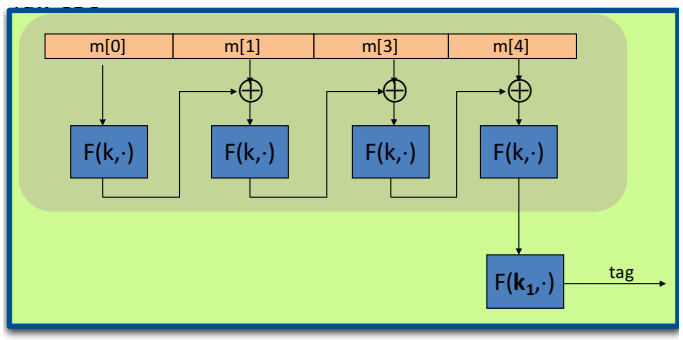
(d) How may we avoid the security problem in raw CBC-MAC? Describe a solution. (2 p).

**Solution:**

A solution to the above attack in the Encrypted CBC-MAC that works as follows. Let $\mathbf{F}: \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{M}$ be a block cipher (PRP) where $\mathcal{M} = \{0,1\}^n$

then the CBC-MAC is defined as follows:

$\mathbf{F}_{\mathsf{ECBC}}: \mathcal{K}^2 \times \mathcal{M}' \rightarrow \mathcal{M}$ where $\mathcal{M}' = \{0,1\}^{\ell n}$




If the Block cipher ($\mathbf{F}$) is **secure** then the encrypted CBC-MAC ($\mathbf{F}_{\mathsf{ECBC}}$) is also **secure**!

(e) Give three advantages of digital signatures in comparison to MACs. (3 p)

**Solution:** Below we describe the main advantages of digital signatures in comparison to MACs. Any two are sufficient to get three points:

- Simpler key-distribution & key management.

- Alice needs to sign a message only once to guarantee integrity if she sends a message to multiple recipients!
- Signatures are publicly verifiable!
- Transferable signatures: Bob can convince a third party that Alice has signed a message!
- Non-repudiation: If Alice signs a message she cannot deny she has signed it.

# 4 Cryptographic Protocols (13 p)

1. Let $p, q$ two large prime numbers such that $N = p \cdot q$. Let $s \in \mathbb{Z}_N$ such that $gcd(s, N) = 1$ and it holds $v = s^2 \pmod N$. Peggy (the prover) and Victor (the verifier) run the following zero-knowledge protocol:

| **Verifier** (Victor) $\mathcal{V}$ | | **Prover** (Peggy) $\mathcal{P}$ |
|---|---|---|
| $(N, v)$ | | $(N, s, v)$ |
| | | $s$ secret key |
| | | pick a random |
| | | $r \in \{1, 2, \ldots, N-1\}$ |
| pick a random | $\xleftarrow{w}$ | $w = r^2 \pmod N$ |
| $c \in \{0, 1\}$ | $\xrightarrow{c}$ | compute |
| check | $\xleftarrow{z}$ | $z = rs^c \pmod N$ |
| $z^2 = wv^c \pmod N$ | | |

(a) What is the probability that a fake Peggy (not having the secret $s$) to be identified correctly. Justify your answer and explain how we may decrease the success probability of a fake Peggy. (1 p).

**Solution:** We notice that for $c = 0$, it holds $z = r \cdot s^0 \pmod N = r \pmod N$.

Thus, a fake Peggy not having the secret s can be correctly identified when $c = 0$ which happens with probability $P(c = 0) = \frac{1}{2}$.

Thus, the probability of a fake Peggy being successful is dropped to $\frac{1}{2}$.

By running the protocol $n$ times the probability of success for a fake Peggy is $(\frac{1}{2})^n$.

(b) Can Victor transfer his knowledge, that indeed Peggy has the secret $x$, to someone else? Explain why. (2 p).

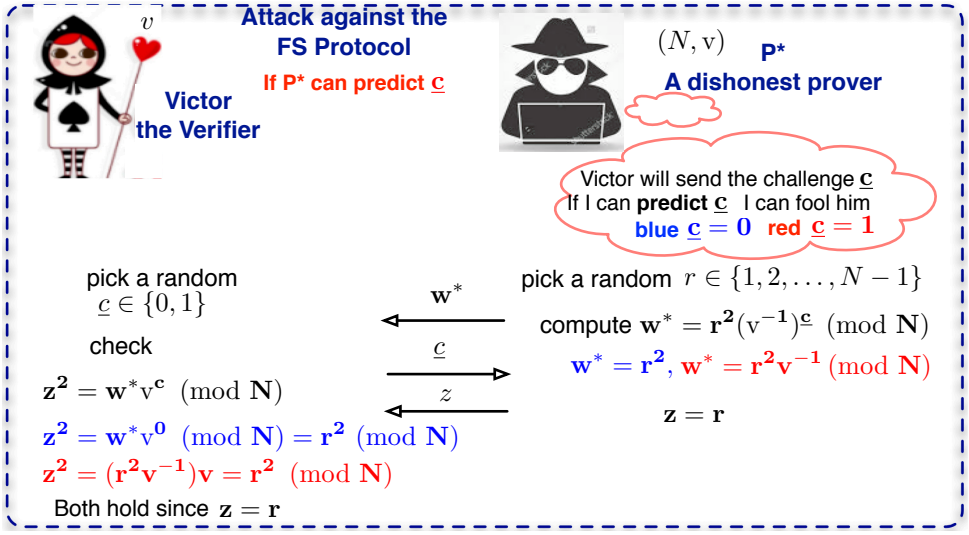
**Solution:** No! $\mathcal{V}$ could have produced $(w, c, z)$ by generating $c$ and $z$ at random and computing $w = z^2 v^{-c} \pmod N$.

(c) Consider that an attacker (who does not have access to the secret key) can always predict Victor's challenge. Describe how the attacker may always successfully pass the protocol. (3 p).

**Solution:**

In this attack, we consider that the dishonest prover $\mathbf{P}^*$ can predict the challenge that **V** (Victor) will send.

$\mathbf{P}^*$ manages to fool **V** i.e., persuades him that he has the secret **s**, while in reality he does not have it.




2. Consider that we have three parties $P_1, P_2, P_3$ and each of them has a secret value $a = 1$, $b = 2$ and $c = 3$ correspondingly. We are using the secure multi party computation (SMPC) protocol for addition (that we have seen in the lectures) based on Shamir's Secret Sharing Scheme with $t = 1$.

(a) Show how $P_1, P_2$ and $P_3$ can compute the sum $\sigma = a + b + c$, without disclosing the values $a$, $b$ and $c$. (4 p)

*Hint:* Describe how $P_1, P_2$ and $P_3$ create their shares and distribute them and how finally the sum is computed.

**Solution:** Since $t = 1$ each of the $P_1, P_2, P_3$ selects a polynomial of degree 1, the only restrictions is that each if $p_1(x)$ is the polynomial selected by the party $P_1$ then it should hold $p_1(0) = a = 1$. Similarly for the other two polynomials it should hold: $p_2(0) = b = 2$ and $p_3(0) = c = 3$.

More precisely, lets assume that $P_1$ selects the polynomial: $p_1(x) = 1 + 2x$. Then, we we have:

$$p_1(1) = 1 + 2 \cdot 1 = 3 = a_1$$
$$p_1(2) = 1 + 2 \cdot 2 = 5 = a_2$$
$$p_1(3) = 1 + 2 \cdot 3 = 7 = a_3$$

Lets assume that $P_2$ selects the polynomial $p_2(x) = 2 + x$. Then, we have:

$$p_2(1) = 2 + 1 = 3 = b_1$$
$$p_2(2) = 2 + 2 = 4 = b_2$$
$$p_2(3) = 2 + 3 = 5 = b_3$$

Lets assume that $P_3 selects the polynomial p_3(x) = 3 + x$. Then, we have:

$$p_3(1) = 3 + 1 = 4 = c_1$$
$$p_3(2) = 3 + 2 = 5 = c_2$$
$$p_3(3) = 3 + 3 = 6 = c_3$$

Then, the shares of the sum $\sigma_1$, $\sigma_2$ and $\sigma_3$ can be calculated as follows:

$$\sigma_1 = a_1 + b_1 + c_1 = 3 + 3 + 4 = 10$$

$$\sigma_2 = a_2 + b_2 + c_2 = 5+4+5 = 14$$
$$\sigma_3 = a_3 + b_3 + c_3 = 7+5+6 = 18$$

Thus, the table filled in looks as follows:

| | $P_1$ | $P_2$ | $P_3$ |
|---|---|---|---|
| $a = 1$ | 3 | 5 | 7 |
| $b = 2$ | 3 | 4 | 5 |
| $c = 3$ | 4 | 5 | 6 |
| $\sigma = a+b+c$ | 10 | 14 | 18 |

We show how to compute

$$\delta_i(0) = \prod_{j=\{1,2,3\}\setminus\{i\}} \frac{j}{j-i} \text{ for } i = \{1,2,3\}$$

It holds:

$$\delta_1(0) = \frac{2}{2-1} \cdot \frac{3}{3-1} = 2 \cdot \frac{3}{2} = 3$$

$$\delta_2(0) = \frac{1}{1-2} \cdot \frac{3}{3-2} = -3$$

$$\delta_3(0) = \frac{1}{1-3} \cdot \frac{2}{2-3} = \frac{1}{-2} \cdot \frac{2}{-1} = 1$$

Thus, we have:

$$\begin{aligned}\sigma &= \delta_1(0) \cdot \sigma_1 + \delta_2(0) \cdot \sigma_2 + \delta_3(0) \cdot \sigma_3 \\ &= 3 \cdot 10 - 3 \cdot 14 + 1 \cdot 18 = 30 - 42 + 18 = 6\end{aligned}$$

Indeed this is correct since $\sigma = a+b+c = 6$.

(b) Consider that $P_3$ decides not to collaborate with $P_1$ and $P_2$. Can $P_1$ and $P_2$ still compute the sum $\sigma$? If yes, justify why and show how. (3 p)

**Solution:** Indeed since $t = 1$ two parties are sufficient in order to compute $\sigma$.

Since only $P_1$ and $P_2$ collaborate we compute

$$\delta_i(0) = \prod_{j=\{1,2\}\setminus\{i\}} \frac{j}{j-i} \text{ for } i = \{1,2\}$$

.

It holds:

$$\delta_1(0) = \frac{2}{2-1} = 2$$

$$\delta_2(0) = \frac{1}{1-2} = -1$$

Thus, we have: $\sigma = \delta_1(0) \cdot \sigma_1 + \delta_2(0) \cdot \sigma_2 = 2 \cdot 10 - 1 \cdot 14 = 6$

Indeed this is correct since $\sigma = a+b+c = 6$.