CHALMERS — GÖTEBORGS UNIVERSITET

EXAM IN CRYPTOGRAPHY

TDA352 (Chalmers) - DIT250 (GU)

17 January 2020, 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.
The grades are:

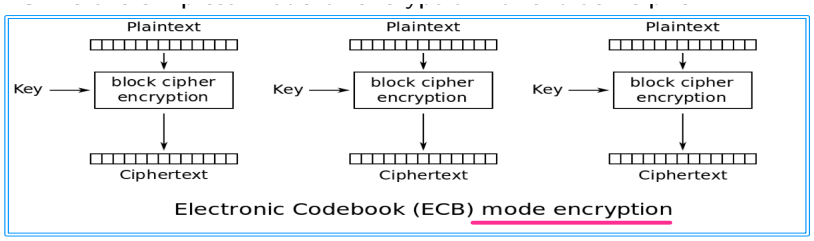
CTH Grades:	22-30 $\rightarrow$ 3	31-39 $\rightarrow$ 4	40-50$\rightarrow$5
GU Grades:	22-39 $\rightarrow$ G	40-50 $\rightarrow$ VG	

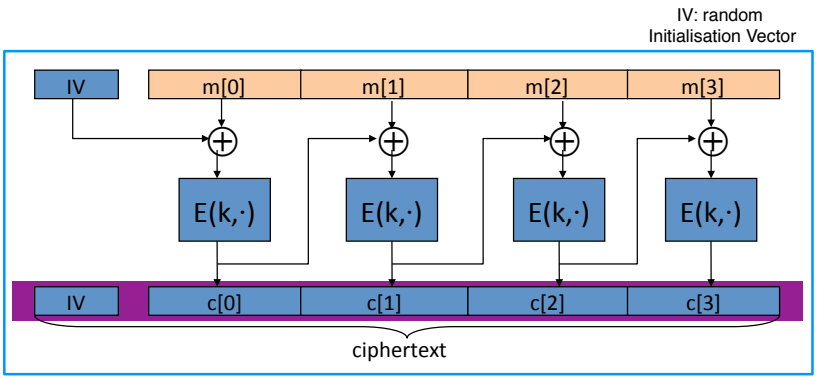
Good luck!

1 Symmetric Ciphers (12 p)

(a) Imagine you are designing a block cipher based cryptographic file system. Describe the modes ECB, CBC, and CTR, and discuss their relative merits in this setting. Which mode would you choose? (5 p)

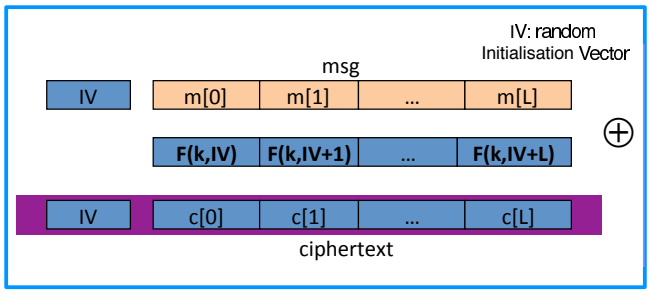
Solution: Below we describe the encryption of the ECB, CBC and CTR modes of operation. The decryption works in a reverse way.





Cipher Block Chaining (CBC) encryption mode

The CTR block cipher is defined as follows: $\mathbf{E}(k, m)$: pick a random $IV \in \{0,1\}^{nt}$
Let $F : K \times \{0,1\}^n \to \{0,1\}^n$ be a secure PRF and do:



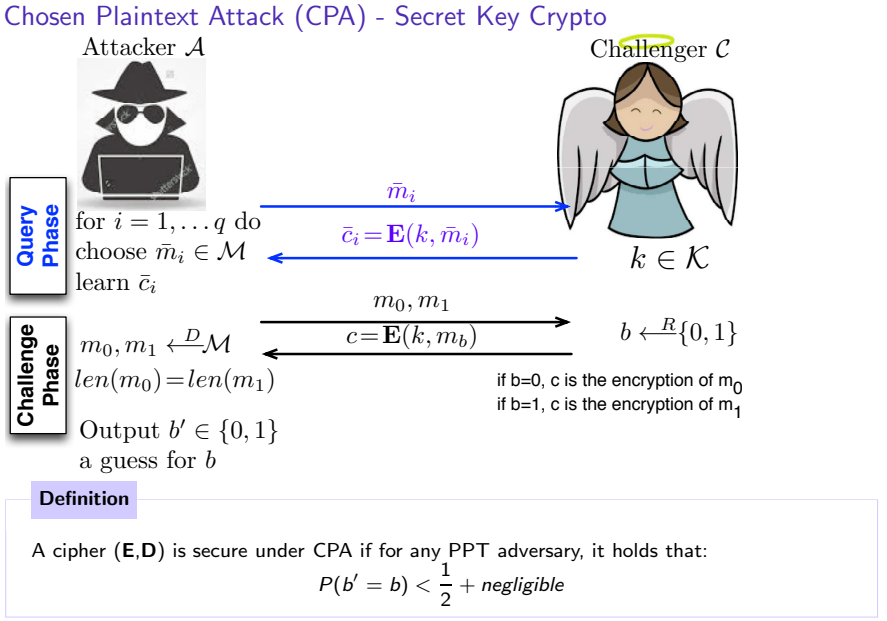
CTR (deterministic counter mode) encryption mode

ECB is rather fast and parallel since for each block you can directly compute its encryption. The same applies for the CTR mode of operation. However, the CBC mode of operation is sequential since each encrypted block is used in the encryption of the subsequent block.

However, ECB is not semantically secure when a message is longer than one block since the encryption is deterministic. CBC provides stronger security guarantees when the IV is unpredictable but the ciphertext is longer since the IV is also needed. The latter also applies in CTR since an IV is also used in this mode.

(b) How do we define IND-CPA security in secret key encryption? (3 p)

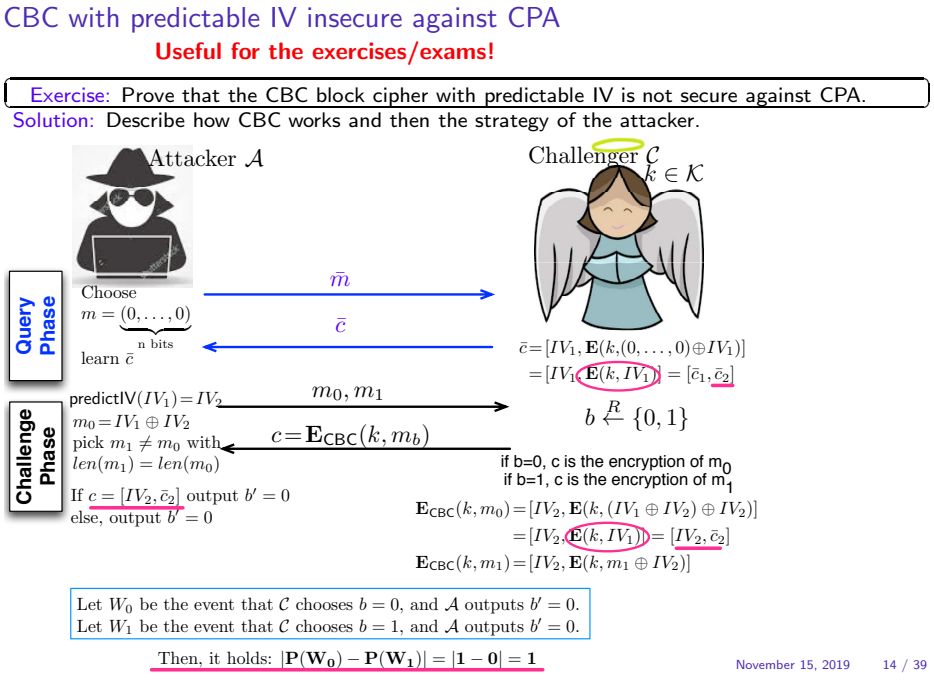
Solution:



(c) Prove that the CBC block cipher with predictable IV is not secure against CPA. (4 p)

Hint: Use in your description a security game and a successful strategy of the attacker in the case the attacker has access to a predictable IV.

Solution:



2 Public Key Encryption (9 p)

(a) Bob receives two messages and their corresponding ciphertexts, m_1, c_1 and m_2, c_2 with El Gamal encryption under the same public key:

$$Enc(m_1) = (g^{r_1}, h^{r_1} \cdot m_1)$$
$$Enc(m_2) = (g^{r_2}, h^{r_2} \cdot m_2)$$

for a cyclic group $G =< g >$ of order q and $h = g^x$, $pk = (G, g, q, h)$ and $sk = (x)$, and $r_1, r_2, x \xleftarrow{R} \mathbb{Z}_q$.

Bob does not have the secret key x used for the encryption of the messages. Can he generate a new message and the corresponding ciphertext for this message only by having access to m_1, c_1 and m_2, c_2? (3 p)

Solution:

(Correction Bob does not really need access to the secret key). Since Bob has access to $c_1 = Enc(m_1)$ and $c_2 = Enc(m_2)$ he can easily compute:

$$\begin{aligned} c_1 \cdot c_2 &= Enc(m_1) \cdot Enc(m_2) \\ &= (g^{r_1} \cdot g^{r_2}, h^{r_1} \cdot m_1 \cdot h^{r_2} \cdot m_2) \\ &= (g^{r_1+r_2}, h^{r_1+r_2} \cdot m_1 \cdot m_2) \\ &= Enc(m_1 \cdot m_2) \end{aligned}$$

Thus, for the message $m = m_1 \cdot m_2$ he is able to compute the corresponding ciphertext $c = Enc(m_1 \cdot m_2) = c_1 \cdot c_2$ by having access to m_1, c_1 and m_2, c_2

(b) How is this property called? Does textbook RSA have the same property? Explain and justify your answer. (2 p)

Solution:

The property is called homomorphic.

Indeed for textbook RSA we have:

Let m_1 and its corresponding ciphertext c_1 when encrypting with textbook RSA. I.e. it holds:

$$c_1 = m_1^e \pmod N$$

where $e \xleftarrow{R} \mathbb{Z}_{\Phi(N)}$ such that $GCD(e, \Phi(N)) = 1$.

Similarly for a message m_2 and corresponding cipher text c_2 we have:

$$c_2 = m_2^e \pmod N$$

.

Thus, it is easy to see that by having access to m_1, c_1 and m_2, c_2 it is possible to have the encryption that corresponds to the message $m = m_1 \cdot m_2$. More precisely, it holds:

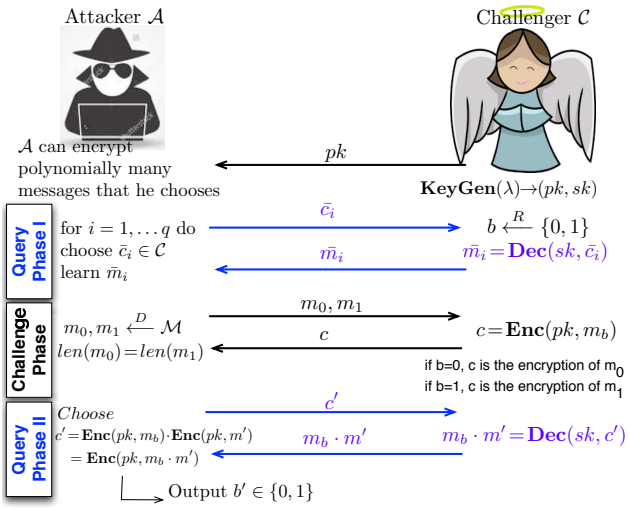
$$\begin{aligned} c_1 \cdot c_2 &= Enc(m_1) \cdot Enc(m_2) \\ &= m_1^e \pmod N \cdot m_2^e \pmod N \\ &= (m_1 \cdot m_2)^e \pmod N \\ &= Enc(m_1 \cdot m_2) \end{aligned}$$

Thus, for the message $m = m_1 \cdot m_2$ Bob is able to compute the corresponding ciphertext $c = Enc(m_1 \cdot m_2) = c_1 \cdot c_2$ by having access to m_1, c_1 and m_2, c_2

(c) Does this property have an effect on the security of textbook RSA and and El Gamal? Show that one of the two is susceptible to IND-CCA attacks using a security game. (4 p).

Solution:

Yes indeed due to the homomorphic property both El Gamal and textbook RSA are susceptible to IND-CPA attacks. Below is described why El Gamal is susceptible to IND-CPA. Similarly, it can be shown for textbook RSA.



Quiz Question: What should the attacker do next? He knows m' so he can get $\frac{m_b \cdot m'}{m'} = m_b$!

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

3 Data Integrity (17 p)

(a) Explain how textbook RSA can be used for digital signatures, *i.e.,* explain how to sign and how to verify the signature. (3 p)

Solution:

Textbook RSA Signature Scheme

KeyGen$(\lambda) \to (\mathbf{pk}, \mathbf{sk})$

1. generate two distinct λ-bit primes p and q, compute $N = pq$ and $\Phi(N)$.
2. choose an integer $e \xleftarrow{R} \mathbb{Z}_{\Phi(N)}$ such that $GCD(e, \Phi(N)) = 1$ and compute its (modular) inverse $d = e^{-1} \mod \Phi(N)$.
3. set: $\mathbf{pk} = (N, e)$ and $\mathbf{sk} = (N, d)$

$$\mathbf{Sign}(\mathbf{sk}, \mathbf{m}) \to \sigma : compute \quad \mathbf{m^d} \pmod{\mathbf{N}} = \sigma$$

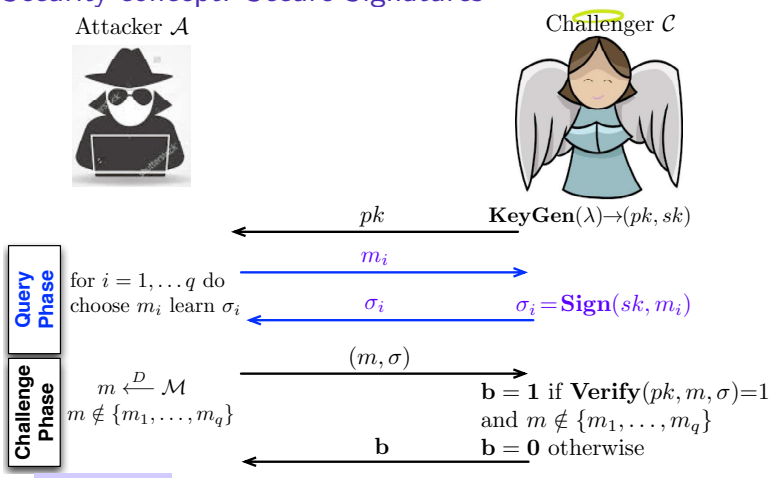
$$\mathbf{Verify}(\mathbf{pk}, \mathbf{m}, \sigma) \to \mathbf{1/0} : \text{outputs 1 if and only if} \quad \sigma^{\mathbf{e}} \pmod{\mathbf{N}} \stackrel{?}{=} \mathbf{m}$$

Φ denotes Euler's phi (or Totient) function i.e., the number of positive integers less than n and relatively prime to n (more info about this in **lec06**).
For now remember $\Phi(N) = \Phi(pq) = (p-1)(q-1)$.

(b) How do we define an existential forgery in digital signatures? (2 p)

Solution:

When an existential forgery is possible then an attacker can win the following game with non-negligible probability.



Definition

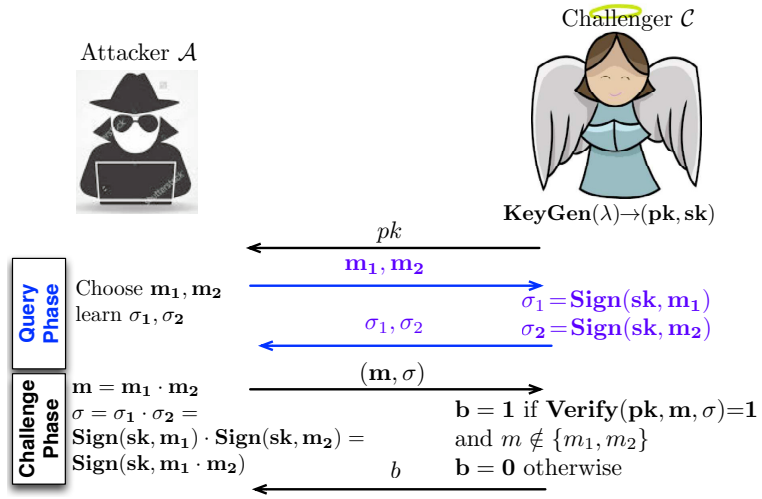
A public signature scheme (**KeyGen, Sign, Verify**) is secure, if for any efficient adversary $\mathcal{A}$ it holds:

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

(c) Bob has received from Alice two signed documents (m_1, σ_1) and (m_2, σ_2). Show that Bob can perform an existential forgery attack on textbook RSA signatures, *i.e.,* given messages, m_1, m_2, and the corresponding signatures, σ_1 and σ_2, show how to construct a new message m and a corresponding valid signature σ without having access to the key. (5 p)

Hint: Show this by describing an attacker and his strategy to win the existential forgery game against textbook RSA signatures.

Solution:



The attacker $\mathcal{A}$ is able to create a valid pair of a message m and a corresponding signature σ, using the homomorphic property of the RSA signature scheme and thus successfully create a forgery, i.e., $\mathbf{P}(\text{Challenger outputs 1}) = \mathbf{1}$

(d) How can we change the way we produce RSA signatures to protect against the above existential forgery attack? Give the construction and show that the attack no longer works. (3 p)

Solution:

How to avoid this type of attack? The hash-and-sign Paradigm

Instead of signing a message **m** we **sign the hash of the message H(m)**!

Why does it work for textbook RSA signatures?

$$\textbf{Sign}(\textbf{sk}, \textbf{H}(\textbf{m}_1)) \cdot \textbf{Sign}(\textbf{sk}, \textbf{H}(\textbf{m}_2)) = H(m_1)^d \cdot H(m_2)^d = \Big(H(m_1) \cdot H(m_2)\Big)^d$$

$$\neq \Big(H(m_1 \cdot m_2)\Big)^d = \textbf{Sign}(\textbf{sk}, \textbf{H}(\textbf{m}_1 \cdot \textbf{m}_2))$$

Hash-and-Sign Construction

Let S= **(KeyGen, Sign, Verify)** be a **public-key signature scheme** scheme for messages of length **n** and **H** be hash function with output **n** bits. Then, we may construct a new signature scheme S'=**(KeyGen', Sign', Verify')** defined as follows:

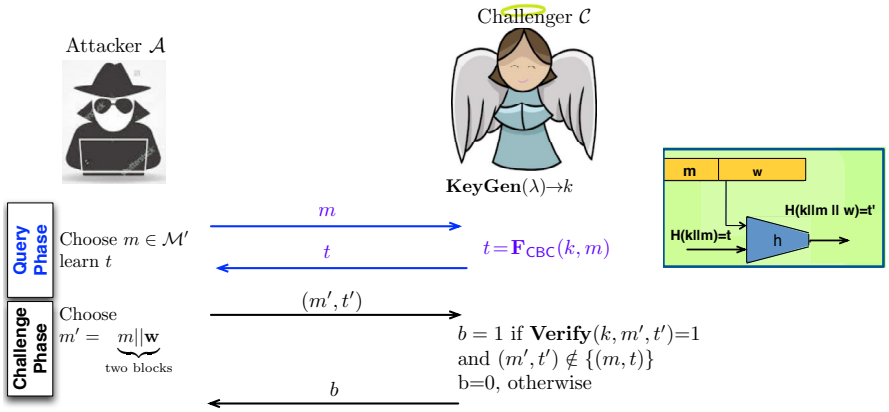
- $\textbf{KeyGen}'(\lambda) \rightarrow (\textbf{pk}, \textbf{sk})$: is a key generation algorithm that runs KeyGen and outputs a public and a private key $(\textbf{pk}, \textbf{sk})$.
- $\textbf{Sign}'(\textbf{sk}, \textbf{m}) \rightarrow \sigma$: is a signing algorithm that, given a secret key **sk** and a message **m** it computes $\textbf{Sign}(\textbf{sk}, \textbf{H}(\textbf{m})) = \sigma$
- $\textbf{Verify}'(\textbf{pk}, \textbf{m}, \sigma) \rightarrow \{\textbf{1}, \textbf{0}\}$: is a **deterministic** verification algorithm that, given the public key **pk** and a signature σ, it computes $Verify(pk, H(m), \sigma)$ and outputs **1** if the signature verifies or **0** if it does not verify.

(e) Consider that we employ a hash function (based on the Merkle-Damgård construction) to construct a MAC. More precisely it holds: $MAC(k,m) = H(k||m)$. Is this MAC construction secure against existential forgery? Prove your reasoning.

Hint: Show this by describing an attacker and his strategy to win the existential forgery game against the MAC construction. (4 p)

Solution:

What if we use: $\textbf{MAC}(\textbf{k}, \textbf{m}) = \textbf{H}(\textbf{k}||\textbf{m})$



The attacker is able to produce a valid pair $(\textbf{m}', \textbf{t}')$ thus $b = 1$. So it holds:

$\textbf{P}(\text{Challenger outputs 1}) = \textbf{1}$

4 Cryptographic Protocols (12 p)

1. We recall the Fiat-Shamir authentication protocol. Let $N = p \cdot q$, where p and q are primes. The prover P wants to convince the verifier V that he knows a square-root of $y \in \mathbb{Z}_N^*$, *i.e.,* a number x such that $y = x^2 \in \mathbb{Z}_N^*$, without revealing x to V. They use the following protocol. All computations are in $\mathbb{Z}_N^*$.

   - P generates a random r, computes $R = r^2$ and sends R to V (the commitment).
   - V generates a uniformly random bit b and sends it to P (the challenge).
   - If $b = 0$, P responds with $z = r$, if $b = 1$ with $z = r \cdot x$ (the response).

(a) What computation will V perform to check P's values? (2 p)

Solution:

V will check if it holds: $z^2 = R \cdot y^b \pmod N$. Indeed we have:

$$z^2 = \begin{cases} r^2 \cdot y^0, & \text{if } b = 0 \\ r^2 \cdot y, & \text{if } b = 1 \end{cases} = \begin{cases} r^2, & \text{if } b = 0 \\ r^2 \cdot x^2, & \text{if } b = 1 \end{cases}$$

(b) Discuss how a cheating P, who does not know x, can achieve a probability of 0.5 of passing the test. (2 p)

Solution:

A cheating P (who does not know x) can pass the protocol when $b = 0$ since then it holds: $z^2 = R$. Since b is chosen at random, there is $1/2$ probability that $b = 0$ and thus that the cheating P passes the protocol.

(c) This protocol is used in decoders for Pay-TV access control. The decoder plays the role of the verifier, while the prover is a smart-card bought by the viewer. Here y is the card number, which is publicly known and transmitted to the decoder. The secret x is stored in the smart-card software. The broadcast periodically contains an instruction to check authenticity of the smart-card, together with the random b to be used in the next run of the protocol.

Early uses of this protocol in decoders did not generate the commitment r at random each time but used same r repeatedly (since V did not anyhow have memory enough to check that r was different each time). Explain how this gave opportunities for production of pirate cards. (3 p)

Solution: If for two runs of the protocol the same r is used then the same $R = r^2 \pmod N$ is transmitted. Thus, the transcripts for the two runs of the protocol would be: (R, b, z) and (R, b', z').

We also know that it holds: $z^2 = R \cdot y^b \pmod N$ and $z'^2 = R \cdot y^{b'} \pmod N$. Thus, we have:

$$\frac{z^2}{y} = R = z'^2 \pmod N \Rightarrow y = (\frac{z}{z'})^2 \pmod N$$

Thus, $x = \frac{z}{z'} \pmod N$

More precisely it holds:

$$(\frac{z}{z'})^2 = y^{b-b'} \pmod N$$

Since 2 and $(b - b')$ are relatively prime using Bezout's identity we have:

$$u \cdot 2 + v \cdot (b - b') = 1 \Rightarrow y^{v(b'-b)} = (\frac{s'}{s})^{ve} = y^{1-ue} \pmod N$$

as a consequence: $y = ((\frac{z}{z'})^v \cdot y^u)^2$

2. Consider that we have three parties P_1, P_2 , P_3 and each of them has a secret value $a = 4$, $b = 3$ and $c = 1$ correspondingly. Show how P_1, P_2 and P_3 using the secure multi party computation protocol for addition (that we have seen in the lectures) based on Shamir's Secret Sharing Scheme with $t = 1$ can compute the sum σ; i.e., fill in the following table. Consider that we are in $\mathbb{Z}_{11}$. (5 p)

	P_1	P_2	P_3
$a = 4$	a_1	a_2	a_3
$b = 3$	b_1	b_2	b_3
$c = 1$	c_1	c_2	c_3
$\sigma = a + b + c$	σ_1	σ_2	σ_3

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 $p_1(0) = a = 4$. Similarly for the other two polynomials it should hold: $p_2(0) = b = 3$ and $p_3(0) = c = 1$. More precisely, lets assume that P_1 selects the polynomial: $p_1(x) = 4 + 2x$. Then, we we have:

$$\begin{aligned} p_1(1) &= 4 + 2 \cdot 1 = 6 = a_1 \\ p_1(2) &= 4 + 2 \cdot 2 = 8 = a_2 \\ p_1(3) &= 4 + 2 \cdot 3 = 10 = a_3 \end{aligned}$$

Lets assume that P_2 selects the polynomial $p_2(x) = 3 + x$. Then, we have:

$$\begin{aligned} p_2(1) &= 3 + 1 = 4 = b_1 \\ p_2(2) &= 3 + 2 = 5 = b_2 \\ p_2(3) &= 3 + 3 = 6 = b_3 \end{aligned}$$

Lets assume that P_3 selects the polynomial $p_3(x) = 1 + x$. Then, we have:

$$\begin{aligned} p_3(1) &= 1 + 1 = 2 = c_1 \\ p_3(2) &= 1 + 2 = 3 = c_2 \\ p_3(3) &= 1 + 3 = 4 = c_3 \end{aligned}$$

Then, the shares of the sum σ_1, σ_2 and σ_3 can be calculated as follows:

$$\begin{aligned} \sigma_1 &= a_1 + b_1 + c_1 = 6 + 4 + 2 = 12 \\ \sigma_2 &= a_2 + b_2 + c_2 = 8 + 5 + 3 = 16 \\ \sigma_3 &= a_3 + b_3 + c_3 = 10 + 6 + 4 = 20 \end{aligned}$$

We also have

$$\delta_1(0) = \prod_{j=\{2,3\}, i=1} \frac{j}{j-i} = \frac{2}{1} \cdot \frac{3}{2} = 3$$

$$\delta_2(0) = \prod_{j=\{1,3\}, i=2} \frac{j}{j-i} = \frac{1}{1-2} \cdot \frac{3}{1} = -3$$

$$\delta_3(0) = \prod_{j=\{1,2\}, i=3} \frac{j}{j-i} = \frac{1}{1-3} \cdot \frac{2}{2-3} = \frac{1}{-2} \cdot \frac{2}{-1} = 1$$

Thus, we have:

$$\sigma = \sigma_1 \cdot \delta_1(0) + \sigma_2 \cdot \delta_2(0) + \sigma_3 \cdot \delta_3(0) = 12 \cdot 3 + 16 \cdot (-3) + 20 \cdot 1 = 36 - 48 + 20 = 8$$

Thus, the table filled in looks as follows:

	P_1	P_2	P_3
$a = 4$	6	8	10
$b = 3$	4	5	6
$c = 1$	2	3	4
$\sigma = 8$	12	16	20