Task 3.5

Part a)

Is this method secure?

Yes it is secure. At any time of delivery the box is locked by one or two padlocks. Since Alice and Bob are the only owner of the keys to unlock the box, it can't be open by the transmitter.

Does it also work with cryptographic means? Which problems could arise?

In my opinion it could be problematic, that the message is encrypted two times. The problem could arise when Bob tries to remove his encryption, after he got the message back from Alice. Since Alice didn't decrypt the message and only encrypts the already encrypted text, it could be hard for bob to remove his encryption.

Part b)

Does it work?

Yes, relating to the principles of the XOR operator, the mentioned encryption works. Since XOR is associative, the order of performing the XOR doesn't matter.

$$(key1 \oplus Text) \oplus key2 = key1 \oplus (Text \oplus key2)$$

Additionally XOR of the same element is always zero ($A \oplus A = 0$) and XOR with zero has no impact ($A \oplus 0 = A$). So Bob can easily remove his key, after obtaining the message back, from the encrypted text:

$$key1 \oplus \big(key1 \oplus (Text \oplus key2)\big) = 0 \oplus (Text \oplus key2) = Text \oplus key2$$

Can confidentiality and integrity be assured?

Confidentiality can be assured, because on every transmission the text is encrypted with either one or more key. No unauthorized access to the plain text is possible. (Of course only if the keys are not known for the attacker)

Integrity cannot be assured. Since the attacker can always intercept the message and encrypt it additionally by using XOR with his own key. It would be possible to obtain the third encryption key by using the already addressed XOR operations, but for this the two original keys plus the plain text have to be available to one member of the communication. But this is not the case at any time of the transmission.

Would choosing different random keys for each message have an impact?

No, it is regardless whether the keys are random on every new message, as long as the same key is used for en-and decryption.