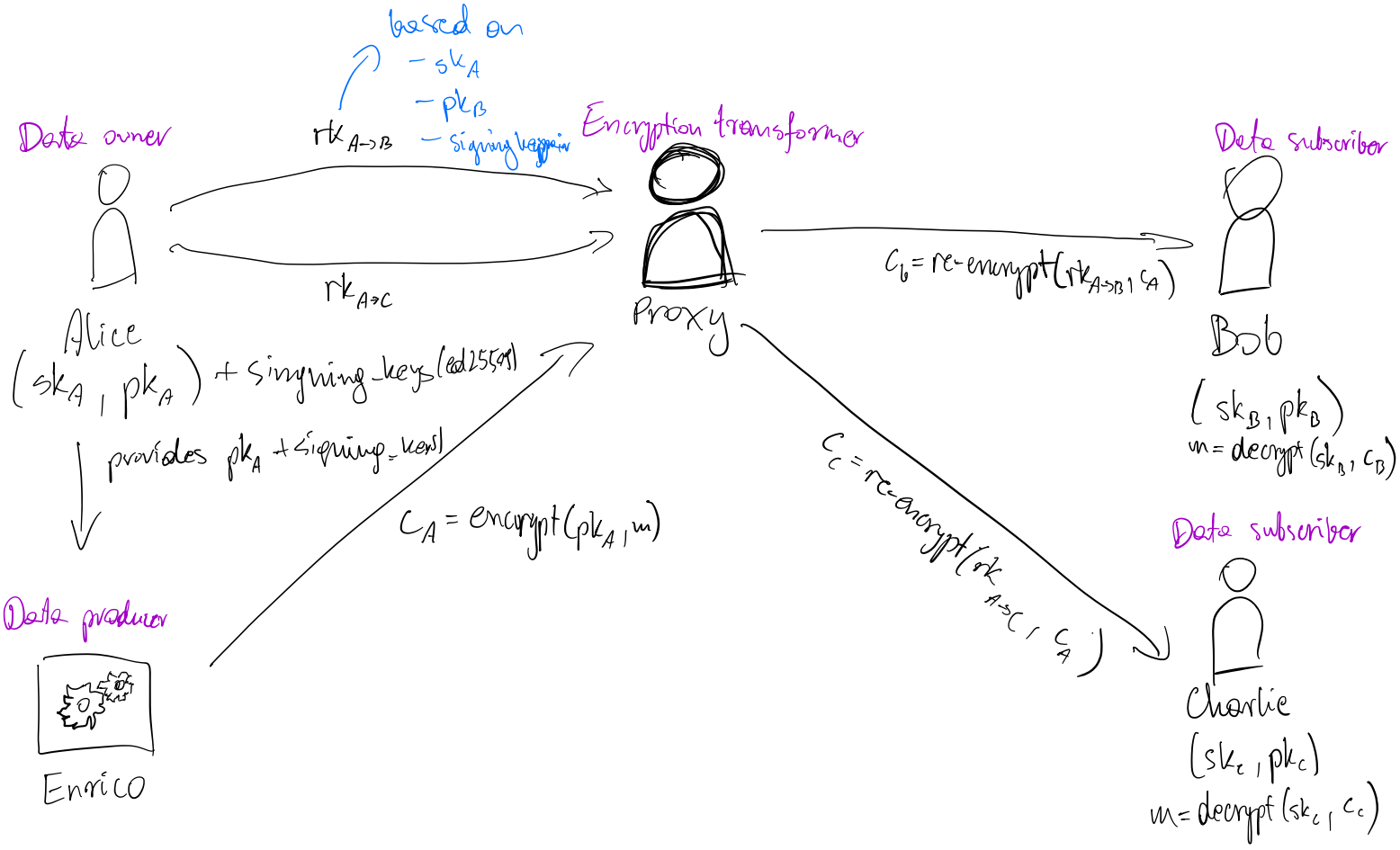

Summary

Alice has her pair of keys (sk_A, pk_A).

Message d is encrypted with a symmetric key DEK.

$D = encrypt\_data(d, DEK)$

Message D is an encrypted version of message d with the DEK key.
Now, it's time for Alice to encrypt the DEK key, and put it into a capsule only she can open while decrypting it with her secret key sk_A.

$$(c_A, k_A) = \text{encrypt}(DEK, pk_A)$$

There is a ciphertext c_A, and a capsule k_A. The capsule can only be opened with a secret key sk_A which is a counterpart to pk_A.

$$DEK = \text{decrypt}(c_A, k_A, sk_A)$$

In order to enable Bob to open the capsule, Alice has to provide a re-encryption key $r_{A \to B}$, such that allow to transform (c_A, k_A) into (c_B, k_B).

$$r_{A \to B} = \text{generate\_reencryption\_key}(sk_A, pk_B)$$

$$(c_B, k_B) = \text{reencrypt}(c_A, k_A, r_{A \to B})$$

Bob would now be able to retrieve DEK.

$$DEK = \text{decrypt}(c_B, k_B, sk_B)$$

To improve the level of security, Alice needs to split the reencryption key $r_{A \to B}$, and send parts to multiple locations. This process makes it possible to reduce trust required from a re-encryption proxy. If it doesn't know the whole $r_{A \to B}$, then it has no incentive to misbehave.

Instead of generating a single $r_{A\to B}$, we will have it split into multiple fragments, called kfrags.

$$kfrags_{A\to B} = split\_reencryption\_key(sk_A, pk_B, M, N)$$

where $kfrags_{A\to B} = \sum_{i=1}^{N} kfrag_{A\to B,i}$

Once Alice have $kfrags_{A\to B}$ generated, she needs to publish it to the network of proxies. These proxies will later use partial reencryption keys $kfrags_{A\to B}$ to transform c_A into $cfrags_{A\to B}$.

$$cfrags_{A\to B} = \sum_{i=1}^{N} reencrypt\_frag(c_A, k_A, kfrag_{A\to B,i})$$

Bob will need exactly M of all distinct $cfrags_{A\to B}$ available in order to reveal DEK. The order of cfrags doesn't matter.

$$(c_B, k_B) = merge(c_A, k_A, cfrags^{*}_{A\to B})$$

And now, finally, Bob can access DEK

$$DEK = decrypt(c_B, k_B, sk_B)$$

Following DEK access, it's time to see what is hidden under the ciphertext D.

$$d = decrypt\_data(D, DEK)$$

□