The Case Against Case #2

Why some people don’t like having the controller distribute traffic keys

Yoav Nir / I2NSF co-chair

This Presentation

- It’s not about me telling you whether case #2 is a good idea or not.
- It’s about getting the discussion going.
- I’ll run through it as quickly as possible, leaving as much time as possible for discussion.

What is Case #2

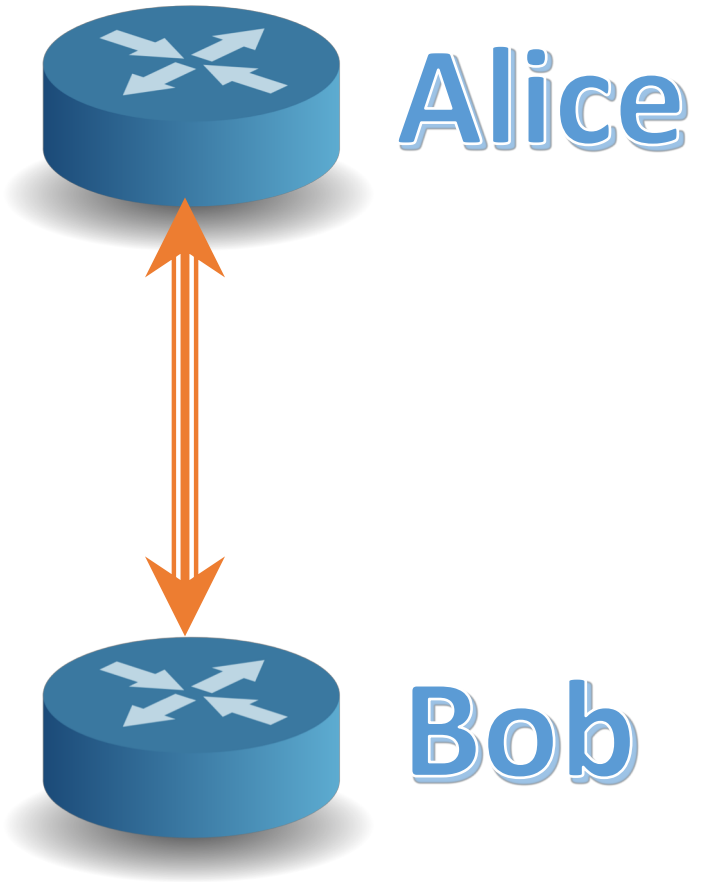
- draft-ietf-i2nsf-sdn-ipsec-flow-protection describes two "cases" or modes of operation:
  - Case #1 where the SDN controller provides the NSFs with SPD and PAD (protected domains and credentials) and they set up traffic keys using IKEv2.
  - Case #2 where the SDN controller provides the NSFs with SPD and SAD (protected domains and traffic keys).
- Some people don't like Case #2 so much.

What's Wrong With Case #2

- It's icky.
  - (that's a technical term)
- We don't like having a traffic key shared between three nodes.
  - "Three can keep a secret, if two of them are dead" – Benjamin Franklin
  - Case #2 adds a bunch of ways for the traffic key to leak.
- We don't like having traffic keys transported.
  - Any key transport is an opportunity for key leakage.
  - Keys should be generated and stored within the cryptographic boundary.

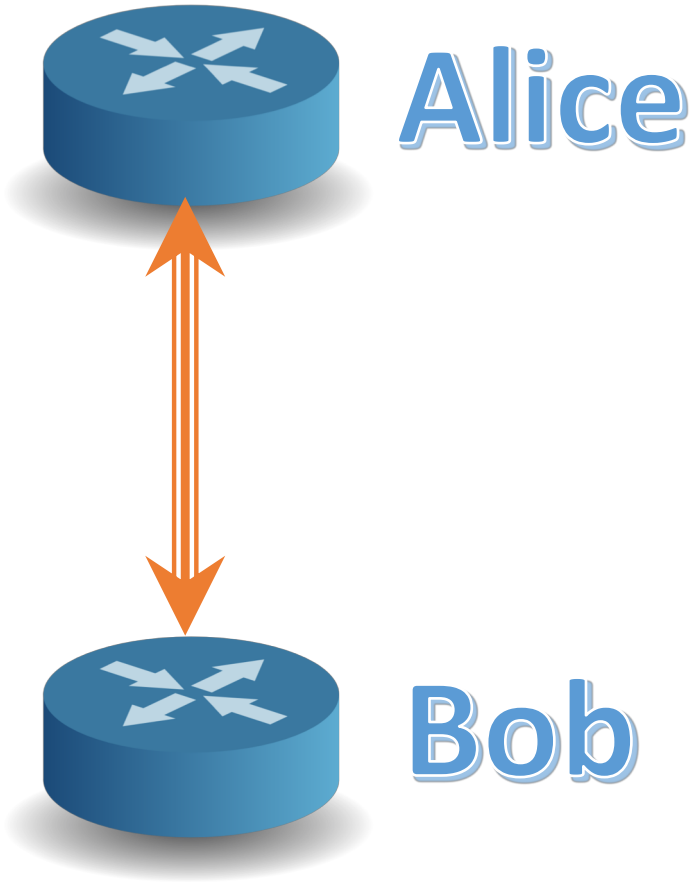
However…

- All these concerns are about not sharing / transporting keys with the controller.
- In an SDN environment, you cannot protect against a rogue controller.
- Suppose you have two NSFs, Alice and Bob, using IKEv2 to negotiate traffic keys.



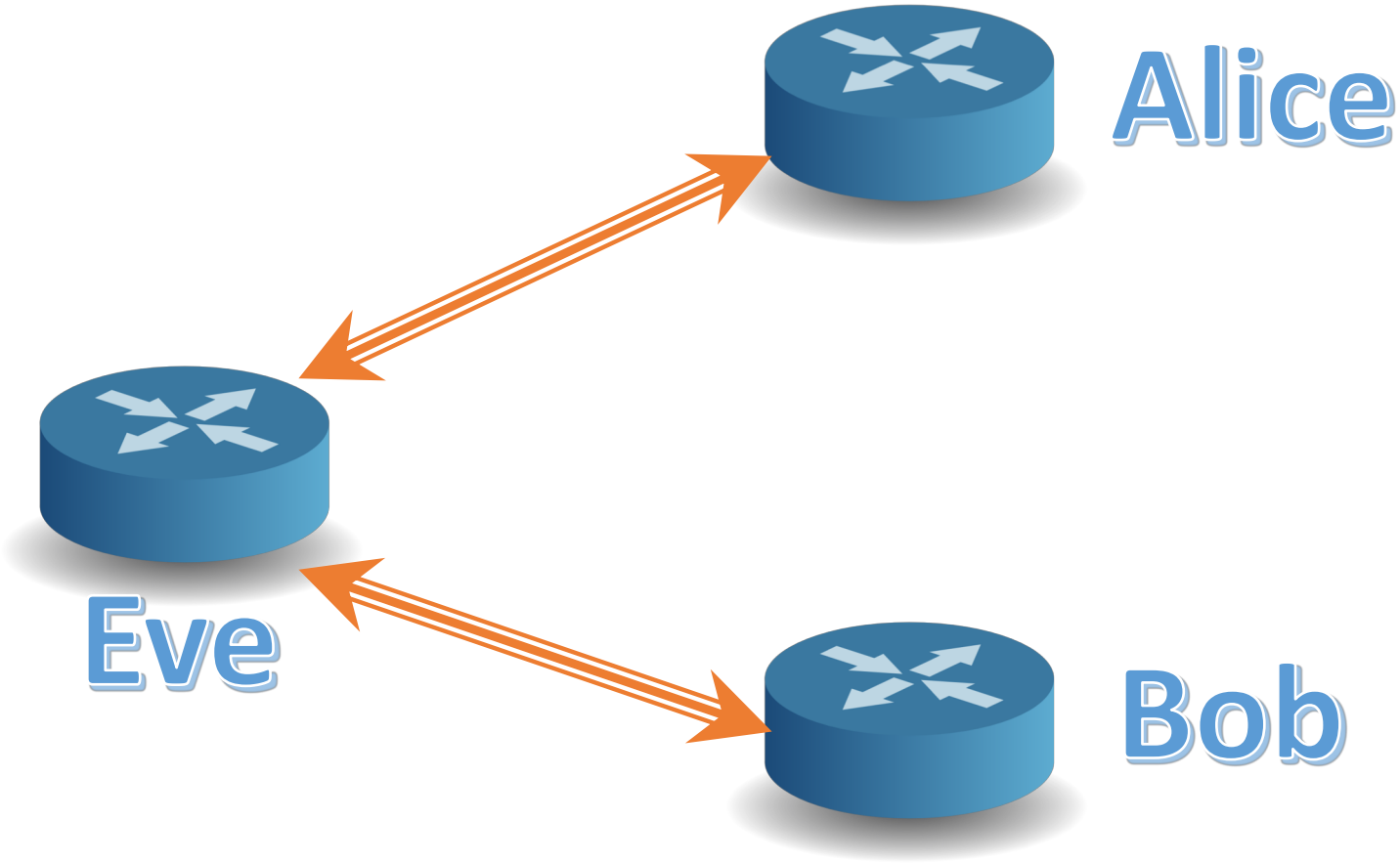
Introducing Eve

- The rogue SDN controller adds a new NSF called Eve.
- It changes Alice's SPD to show that Bob's protected domain is behind Eve.
- It changes Bob's SPD to show that Alice's protected domain is behind Eve.



Eve Gets to See and Record the Plaintext

- The rogue SDN controller adds a new NSF called Eve.
- It changes Alice's SPD to show that Bob's protected domain is behind Eve.
- It changes Bob's SPD to show that Alice's protected domain is behind Eve.



On The Other Hand

- That diagram in the previous slide is a weird way to run a network.
- It’s conspicuous. Eavesdropping is obviously happening here.
- An SDN controller running the network like that will be discovered.
- Sharing keys will not be discovered.
- Some IPsec encryption is done for compliance: HIPPA, PCI, others.
- Are these fine with key sharing? With extracting keys from within the cryptographic boundary?

Why Not Just Case #1?

- We've heard some reasons why not:
  - Simpler implementation
    - Every Linux distro/Windows/Mac comes with IKE/IPsec.
  - Small, cheap NSFs don't have a good random source / time source
    - Then how are they doing the TLS of SSH handshake with the SDN controller?
- These don't sound convincing.

Mitigations For Case #2

- Obviously… Case #1 – let them do IKEv2.
- draft-carrel-ipsecme-controller-ike
  - Sort of a mid-point between Case #1 and Case #2,
  - Simplified key exchange using DH through the Controller.
  - No authentication – public DH keys are distributed by the controller.
  - The controller does not have the private keys, so it can’t calculate the traffic keys.
  - Traffic keys never leave the NSF cryptographic boundary.
  - Still can’t prevent the Controller from introducing a MitM.

Discussion Goes Here