Key Exchange Over the Wiretap Channel

Thor Tunge

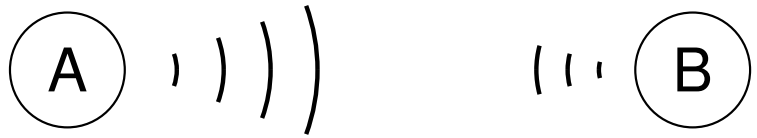
Norwegian University of Science and Technology

Plan

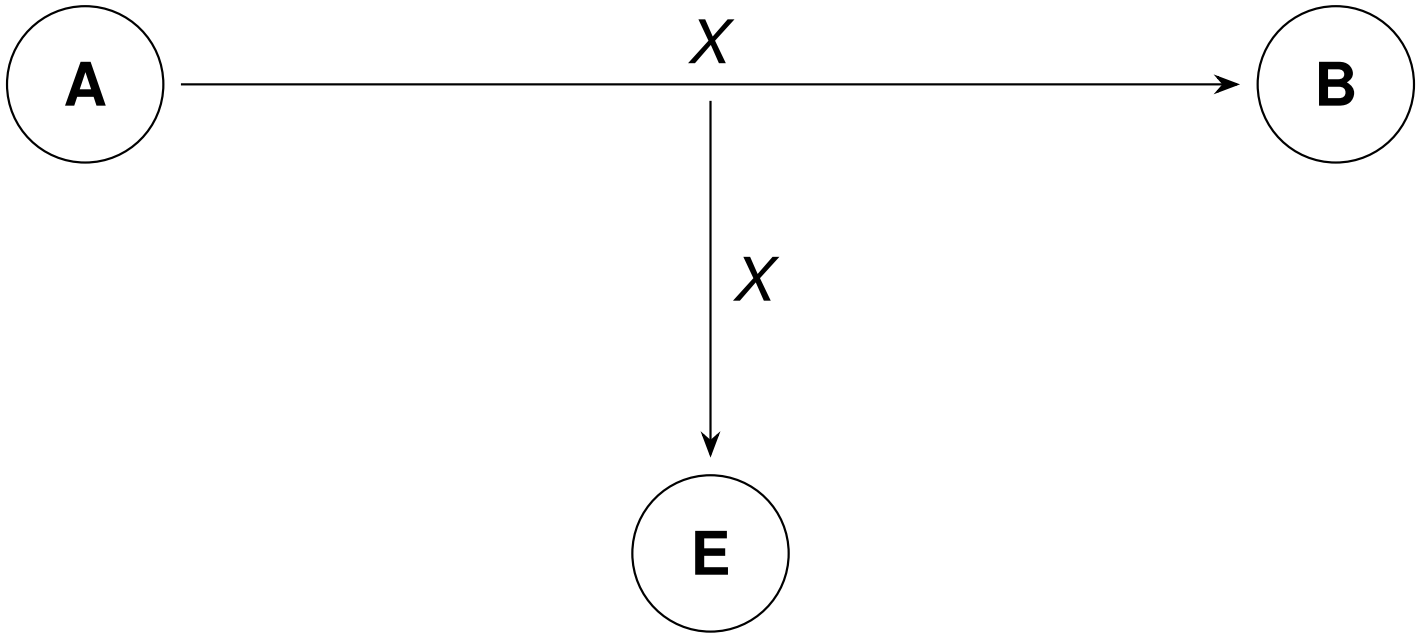
1. Motivation
2. The wiretap channel
3. Quantifying Secrecy
4. Key exchange model

Expensive Asymmetric Crypto

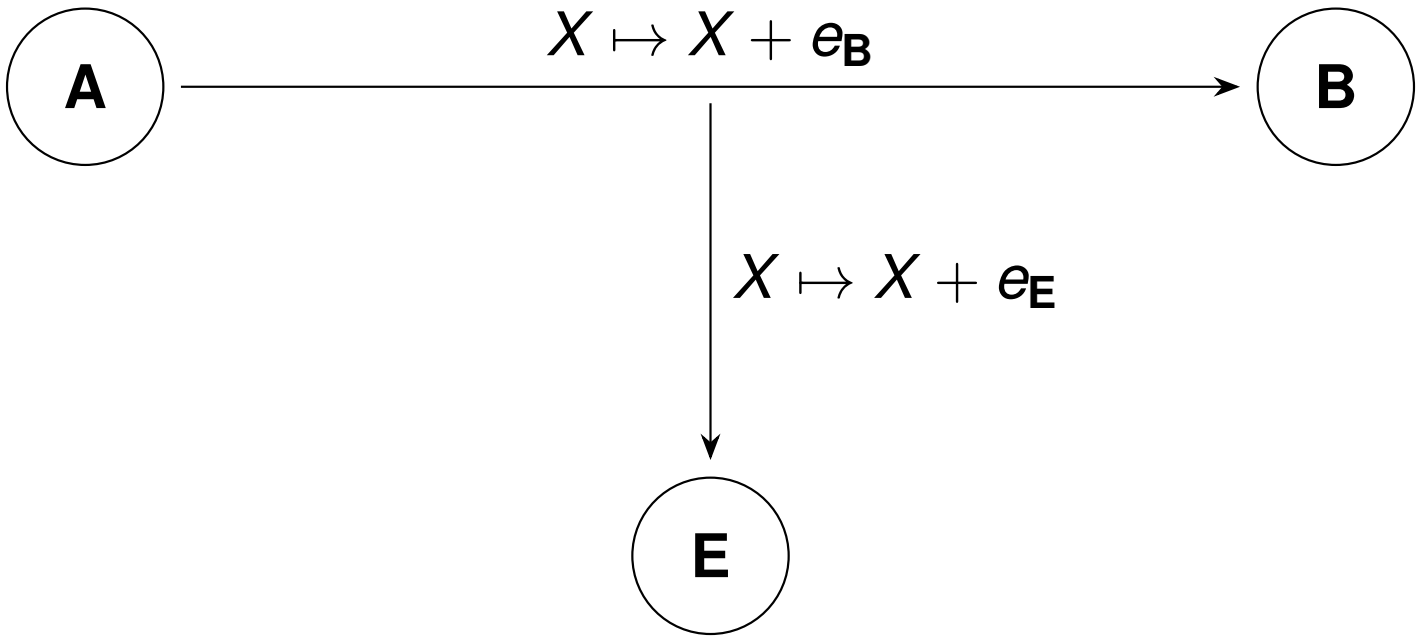
- DH is expensive for very small devices
- All asymmetric crypto is expensive
- Without asymmetric information, how can we get keys?



Channel in Standard Crypto



The Wiretap Channel



— $e_{\mathbf{E}} > e_{\mathbf{B}}$ (hopefully).

— Errors from bit-flips with probability $p_{\mathbf{B}}, p_{\mathbf{E}}$ (independently).

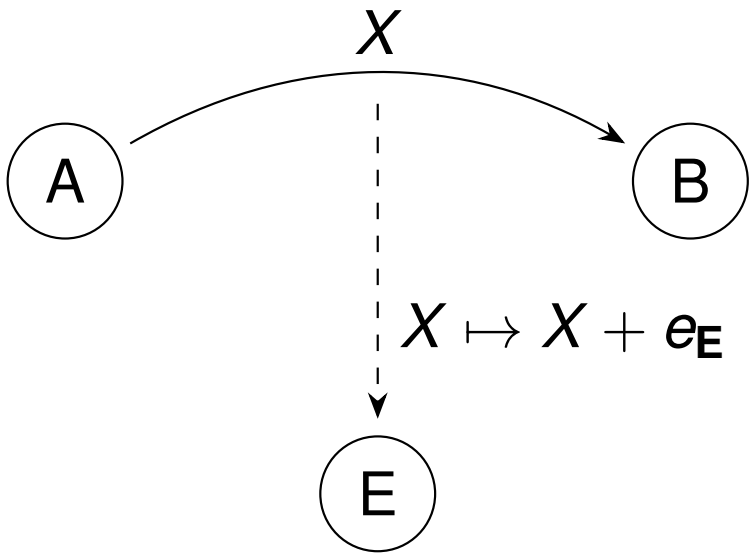
Paradox

— Keys *must* be equal.
— Both channels are noisy.
— Error correcting codes ruin the model.

Toy example

$p_{\mathbf{B}} = 0$ and $p_{\mathbf{E}} = 0.5$

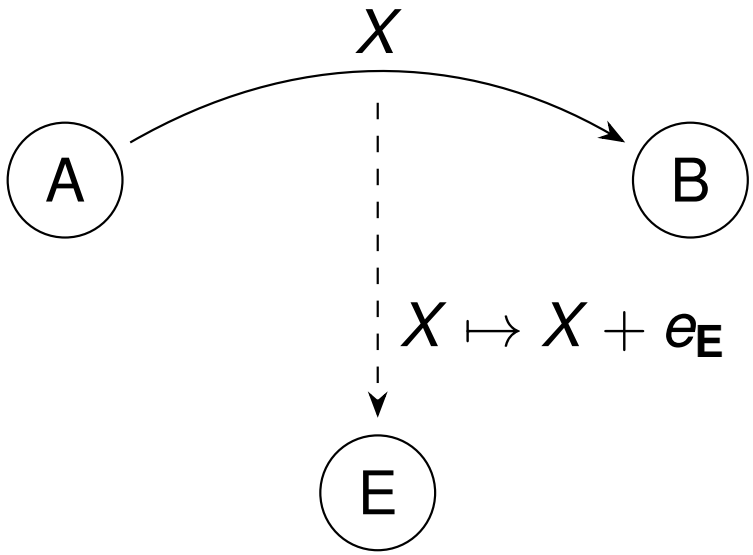
1. Alice: $X \leftarrow \{0,1\}^{100}$
2. Alice: Sends X to Bob.
3. Bob: Receives X.



Toy example

$p_{\mathbf{B}} = 0$ and $p_{\mathbf{E}} = 0.5$

1. Alice: $X \leftarrow \{0,1\}^{100}$
2. Alice: Sends X to Bob.
3. Bob: Receives X.



— Eve observes $X + e_{\mathbf{E}}$ independent of X.

— Eve cannot say anything about X given $X + e_{\mathbf{E}}$.

— Send 1 message $\Rightarrow$ 100 secret bits.

— Only 50% bits are different, but *all* are secret.

Quantifying Secrecy

— Let X, Y, Z be what Alice, Bob and Eve observes.

Quantifying Secrecy

— Let X, Y, Z be what Alice, Bob and Eve observes.
— The entropy $H(X)$ is a measure of the uncertainty of X.

Quantifying Secrecy

— Let X, Y, Z be what Alice, Bob and Eve observes.

— The entropy $H(X)$ is a measure of the uncertainty of X.

- X has one outcome $\Rightarrow H(X) = 0$
- X uniform $\Rightarrow H(X)$ maximal

Quantifying Secrecy

- Let X, Y, Z be what Alice, Bob and Eve observes.
- The entropy $H(X)$ is a measure of the uncertainty of X.
  - X has one outcome $\Rightarrow H(X) = 0$
  - X uniform $\Rightarrow H(X)$ maximal
- $H(X) = -\sum_x \log(\Pr[X = x])\Pr[X = x]$

Quantifying Secrecy

— Let X, Y, Z be what Alice, Bob and Eve observes.

— The entropy $H(X)$ is a measure of the uncertainty of X.

- X has one outcome $\Rightarrow H(X) = 0$
- X uniform $\Rightarrow H(X)$ maximal

— $H(X) = -\sum_x \log(\Pr[X = x])\Pr[X = x]$

— What is the uncertainty of X given Z?

Quantifying Secrecy

— Let X, Y, Z be what Alice, Bob and Eve observes.

— The entropy $H(X)$ is a measure of the uncertainty of X.

- X has one outcome $\Rightarrow H(X) = 0$
- X uniform $\Rightarrow H(X)$ maximal

— $H(X) = -\sum_x \log(\Pr[X = x])\Pr[X = x]$

— What is the uncertainty of X given Z?

— Conditional entropy $H(X|Z)$.

Quantifying Secrecy

— Let X, Y, Z be what Alice, Bob and Eve observes.

— The entropy $H(X)$ is a measure of the uncertainty of X.

- X has one outcome $\Rightarrow H(X) = 0$
- X uniform $\Rightarrow H(X)$ maximal

— $H(X) = -\sum_x \log(\Pr[X = x])\Pr[X = x]$

— What is the uncertainty of X given Z?

— Conditional entropy $H(X|Z)$.

— $H(X|Z) = \sum_z H(X|Z = z)\Pr[Z = z]$

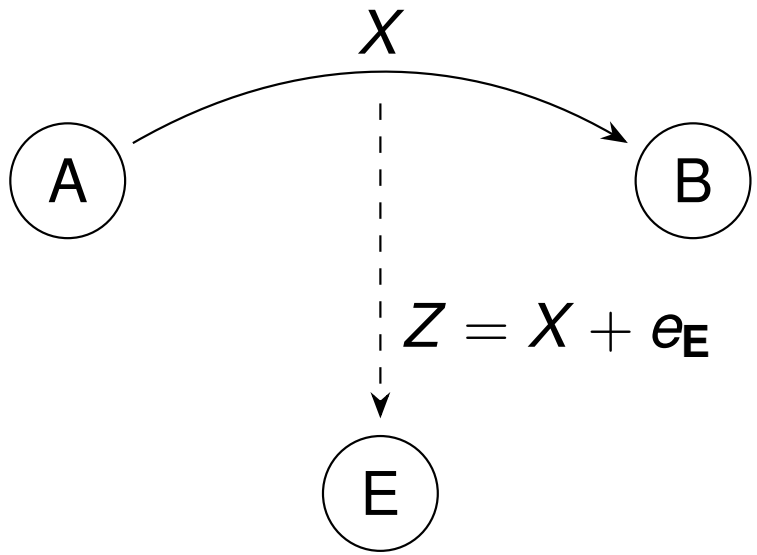
Mutual information

$$I(X;Z) = H(X) - H(X|Z).$$

Example

$p_{\mathbf{B}} = 0$ and $p_{\mathbf{E}} = 0.01$

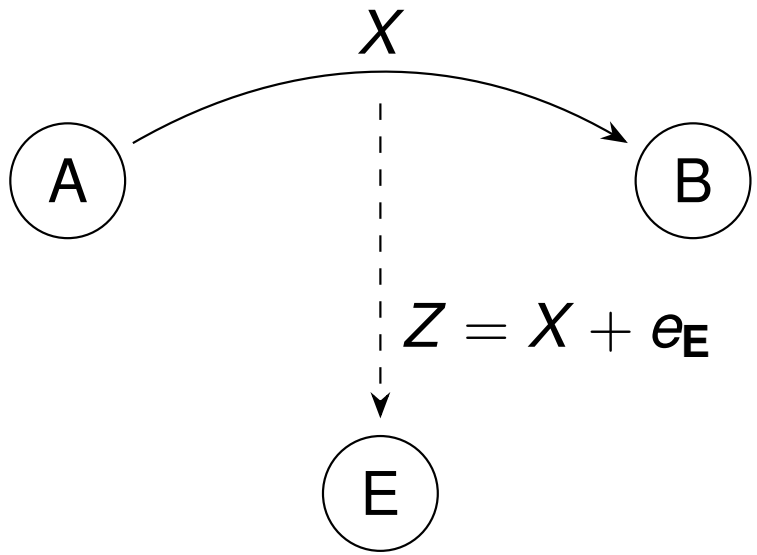
1. Alice: $X \leftarrow \{0,1\}^{100}$
2. Alice: Sends X to Bob
3. Bob: Receives X



Example

$p_{\mathbf{B}} = 0$ and $p_{\mathbf{E}} = 0.01$

1. Alice: $X \leftarrow \{0,1\}^{100}$
2. Alice: Sends X to Bob
3. Bob: Receives X



— Now Eve observes $Z = X + e_{\mathbf{E}}$ dependent on X.

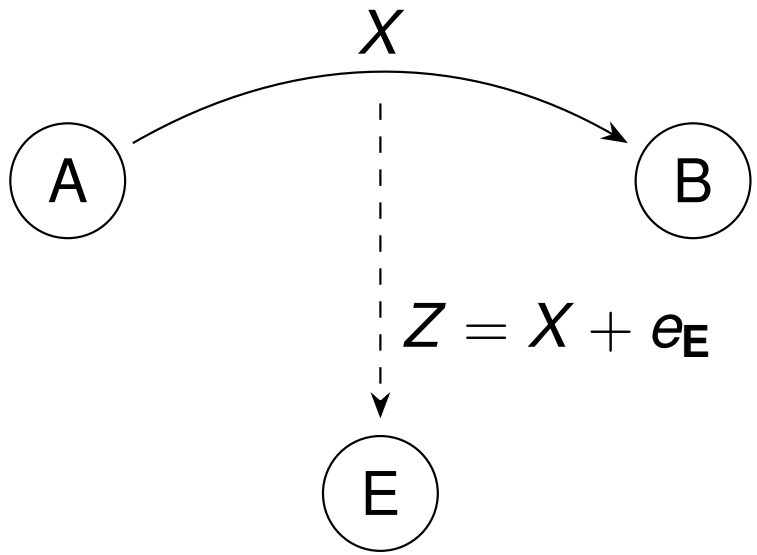
— Eve's mutual information: $I(X|Z) \simeq 0.9$ per bit.

— Send 10 messages $\Rightarrow$ 100 secret bits.

Example

$p_{\mathbf{B}} = 0$ and $p_{\mathbf{E}} = 0.01$

1. Alice: $X \leftarrow \{0,1\}^{100}$
2. Alice: Sends X to Bob
3. Bob: Receives X



— Now Eve observes $Z = X + e_{\mathbf{E}}$ dependent on X.

— Eve's mutual information: $I(X|Z) \simeq 0.9$ per bit.

— Send 10 messages $\Rightarrow$ 100 secret bits.

— Eve knows nothing $\Rightarrow$ information theoretic security.

If Alice uses these bits, the problem is computational.

Correctness Problem

What can we do when Bob's channel is noisy?

Correctness Problem

What can we do when Bob's channel is noisy?

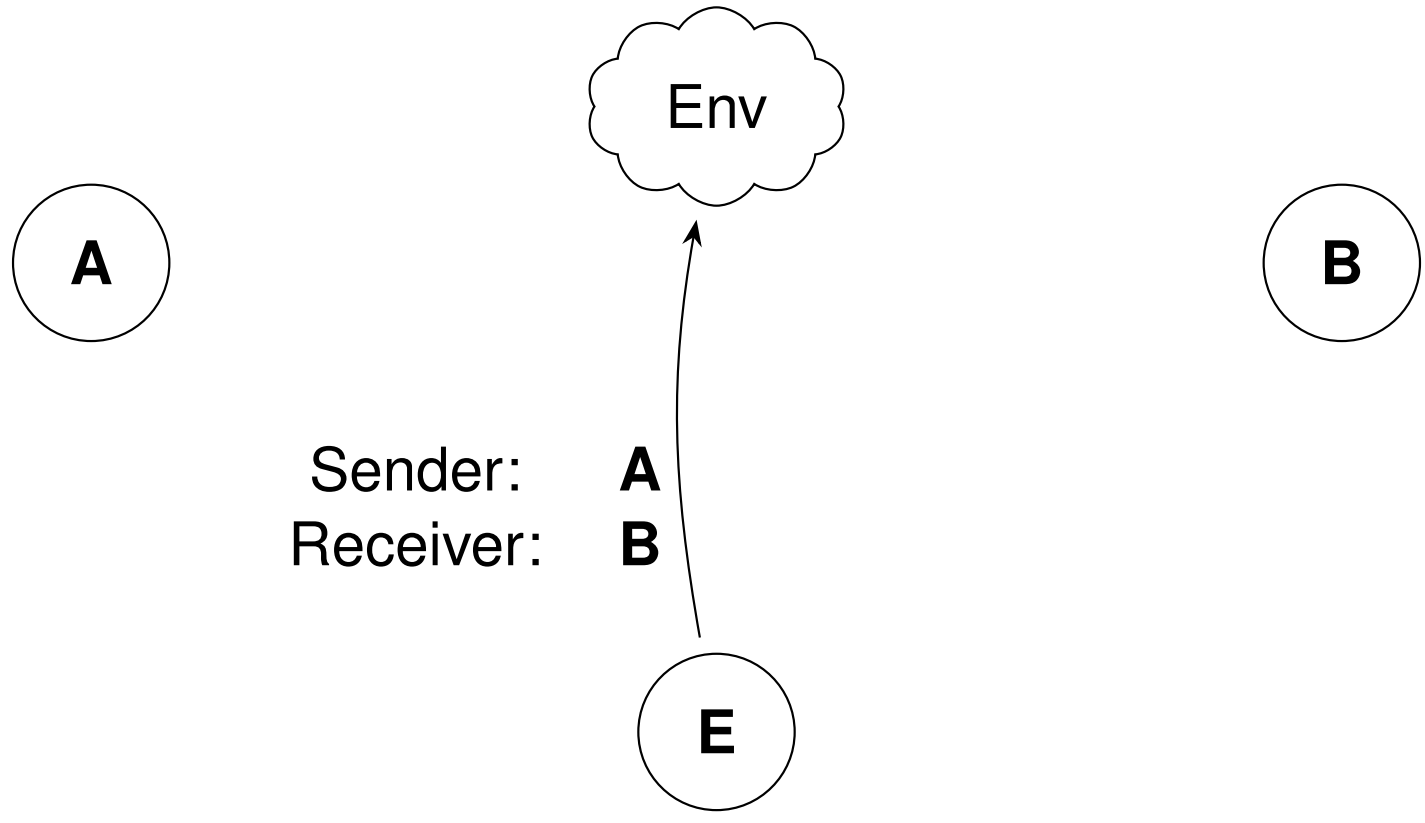
1. Use codes anyways
    - Alice and Bob estimate $p_{\mathbf{B}}$.
    - Use a code that Bob can *just* decode.
    - Not many codewords close to any given message.
    - Requires huge messages.

Correctness Problem

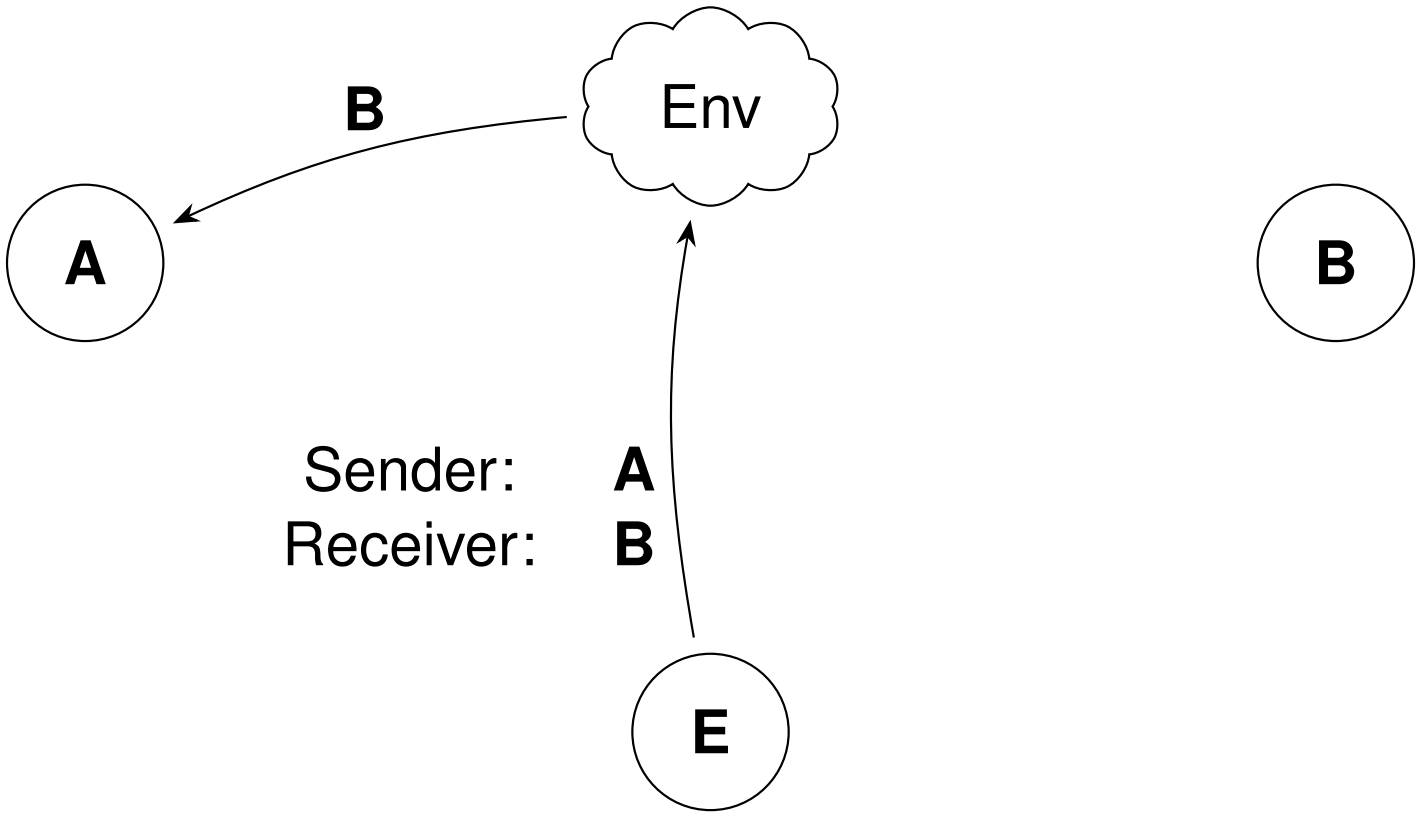
What can we do when Bob's channel is noisy?

1. Use codes anyways
   - Alice and Bob estimate $p_{\mathbf{B}}$.
   - Use a code that Bob can *just* decode.
   - Not many codewords close to any given message.
   - Requires huge messages.
2. Use *error reconciliation*
   - Alice and Bob exchange messages over a clear channel.
   - They find where their string differ and correct errors.
   - Leaks a portion of the shared message.
   - Requires $p_{\mathbf{E}} - p_{\mathbf{B}}$ above some threshold.

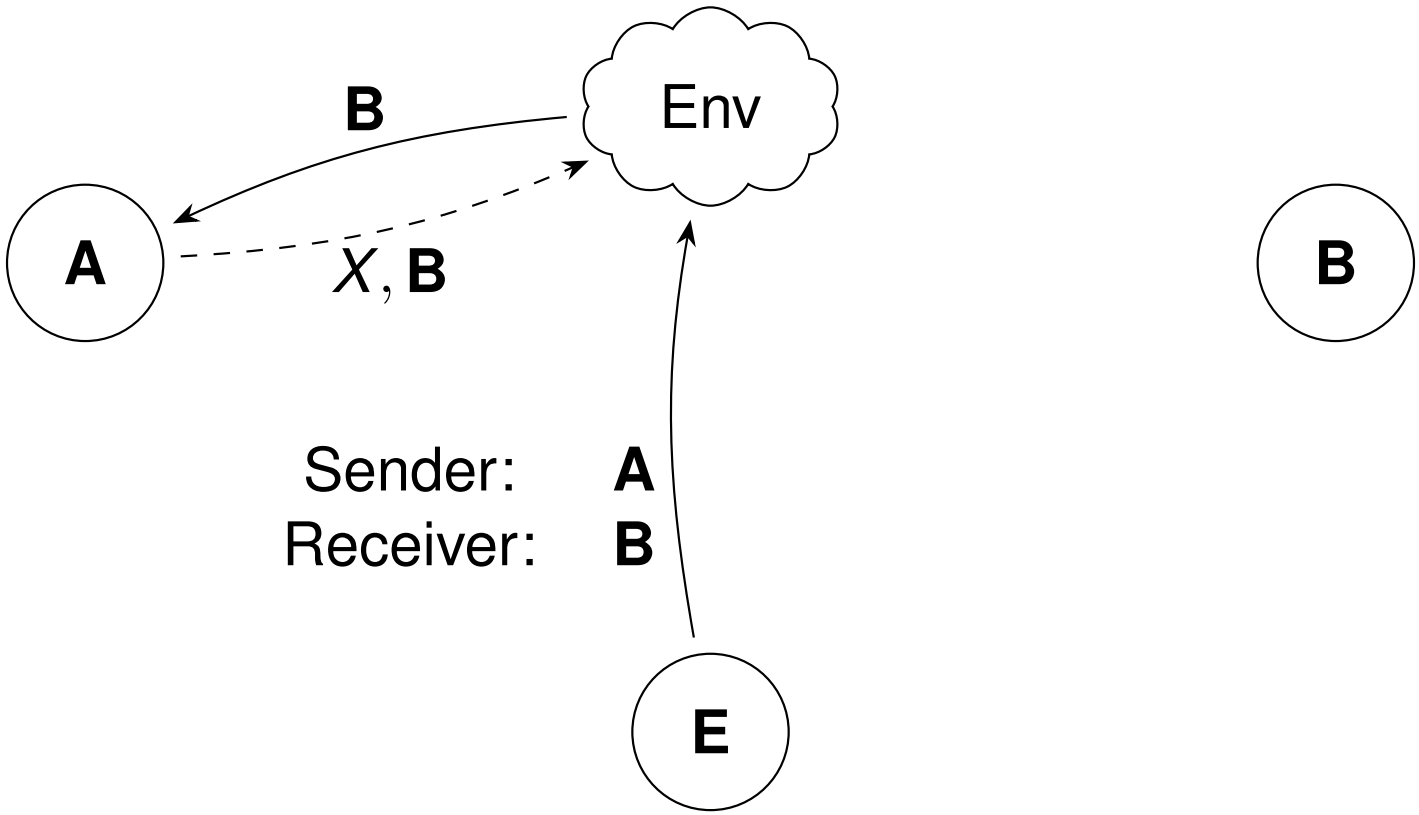
Security model



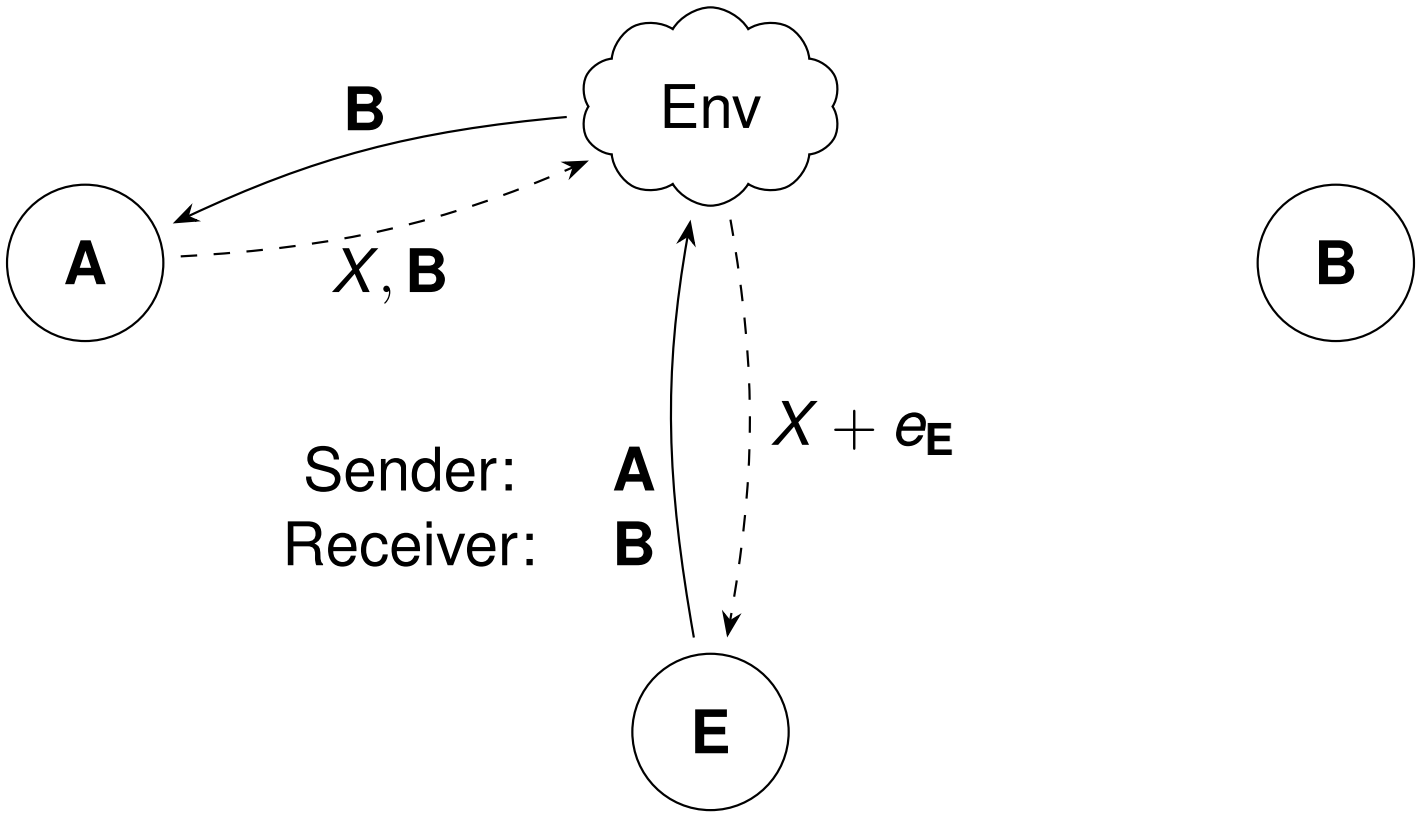
Security model



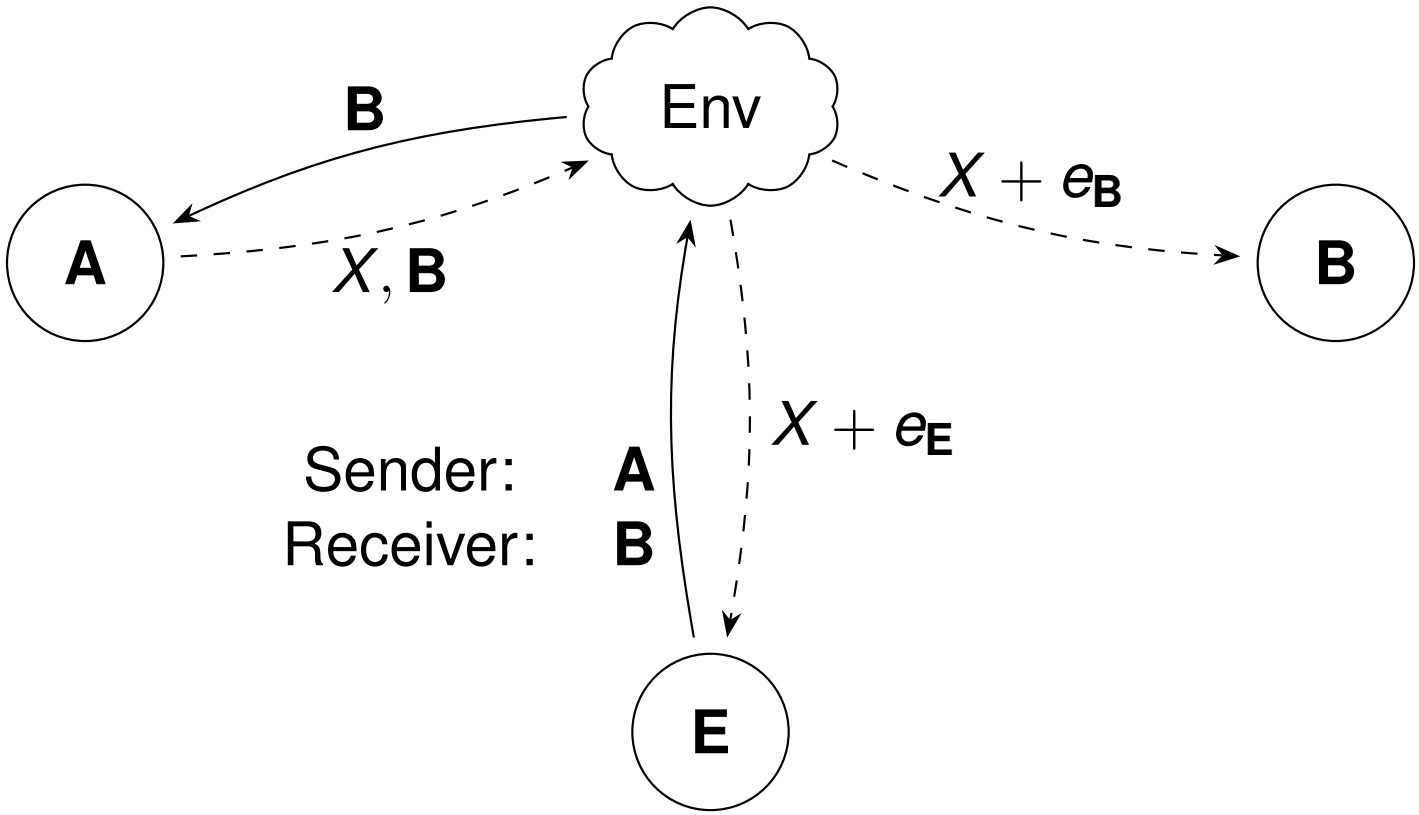
Security model



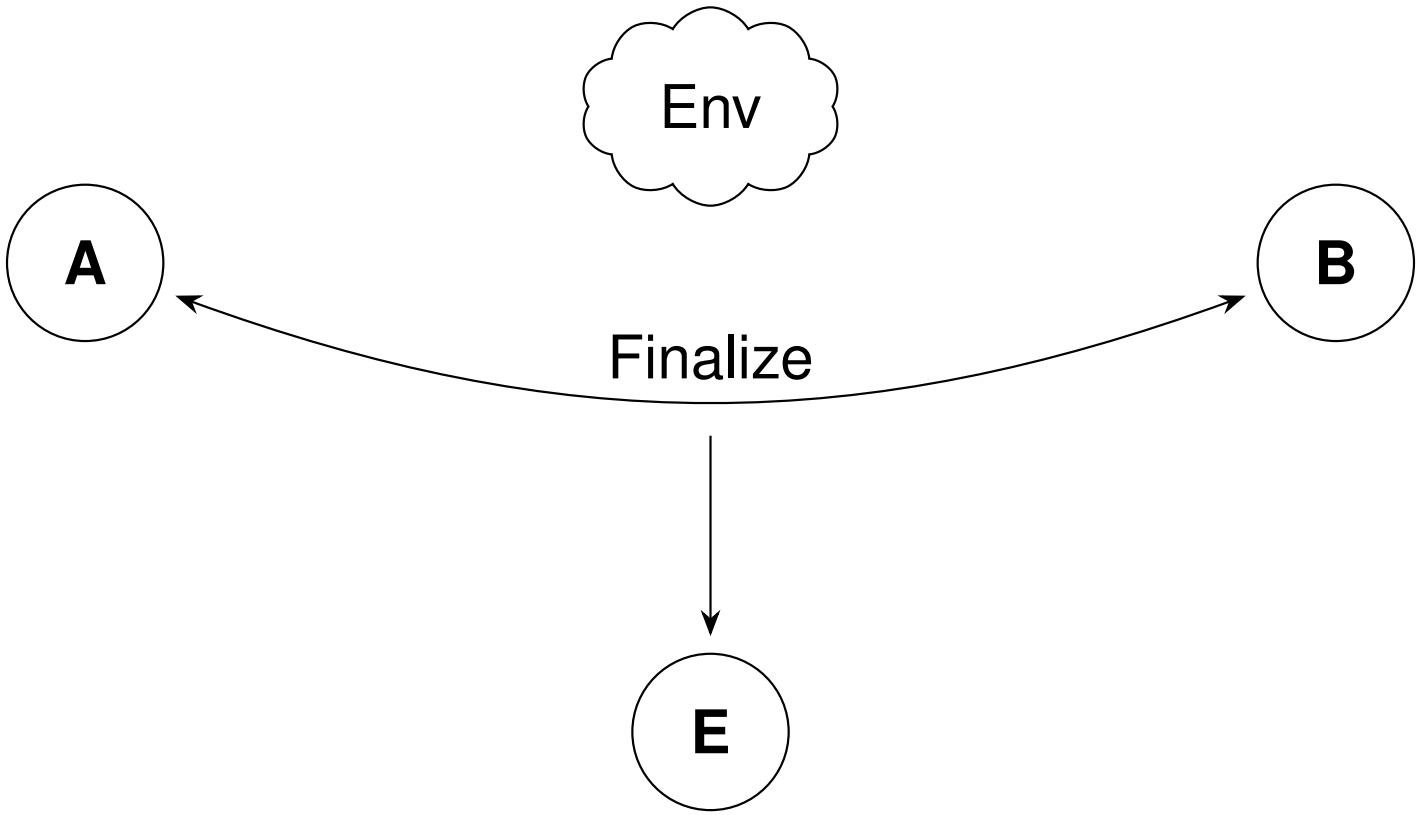
Security model



Security model



Security model



Model Ideas

— Send many messages this way.

Model Ideas

— Send many messages this way.

— Messages are sent at different times: noise is time dependent.

Model Ideas

— Send many messages this way.
— Messages are sent at different times: noise is time dependent.
— Against which types of adversarial noise is a scheme secure?

Model Ideas

— Send many messages this way.

— Messages are sent at different times: noise is time dependent.

— Against which types of adversarial noise is a scheme secure?

- Constant difference $p_{\mathbf{E}} - p_{\mathbf{B}}$?
- An adversary who drops every 100th message?
- Adversarial noise distributed around $p_{\mathbf{B}}$?

Model Ideas

- Send many messages this way.
- Messages are sent at different times: noise is time dependent.
- Against which types of adversarial noise is a scheme secure?
  - Constant difference $p_{\mathbf{E}} - p_{\mathbf{B}}$?
  - An adversary who drops every 100th message?
  - Adversarial noise distributed around $p_{\mathbf{B}}$?
- Since we want to use the keys, we have to be sure that they are secure.

Model Ideas

- Send many messages this way.
- Messages are sent at different times: noise is time dependent.
- Against which types of adversarial noise is a scheme secure?
  - Constant difference $p_{\mathbf{E}} - p_{\mathbf{B}}$?
  - An adversary who drops every 100th message?
  - Adversarial noise distributed around $p_{\mathbf{B}}$?
- Since we want to use the keys, we have to be sure that they are secure.
- Use previous keys and be happy if keys are eventually secure.

Questions?