Data Security: Replay Attacks

Dr. Dan Massey

Replay Attacks:

Not In the Book
Review These Slides for Exams

Key Exchange Algorithms

- Goal: Alice, Bob agree on a ***shared secret*** key
  - Key cannot be sent in clear
    - Attacker can listen in
    - Could send enciphered key… but enciphered with what key?
    - Could be derived from exchanged data plus data not known to an eavesdropper
  - Assume all cryptosystems, protocols publicly known
    - Adversary knows the protocols and cyrptosystems
    - Anything transmitted is assumed known to attacker

Trusted Third Party Exchange

- Bootstrap problem: how do Alice, Bob begin?
  - Alice can’t send key to Bob in the clear!
- Assume some trusted third party, Cathy
  - Alice and Cathy share secret key k_A
  - Bob and Cathy share secret key k_B
  - Some external technique was used to establish shared keys with Cathy
- Can Alice and Bob use Cathy to exchange new Alice/Bob shared key k_s

Naïve Strategy to Lean A Secret Key

Alice $\xrightarrow{\{\text{ request for session key to Bob }\}\ k_A}$ Cathy

Notation: denote a message encrypted by k_A as {msg} k_A

Alice $\xleftarrow{\{\ k_s\ \}\ k_A\ ||\ \{\ k_{s'}\}\ k_B}$ Cathy

Alice $\xrightarrow{\{\ k_s\ \}\ k_B}$ Bob

Alice and Bob now share secret key Ks.
Ks is known only to Alice, Bob, and Cathy. Cathy is a trusted third party.

Alice $\xrightarrow{\{\ \textit{Buy 100 shares of XYZ Stock}\ \}\ k_S}$ Bob

Bob believes this an authentic message from Alice and places the order.

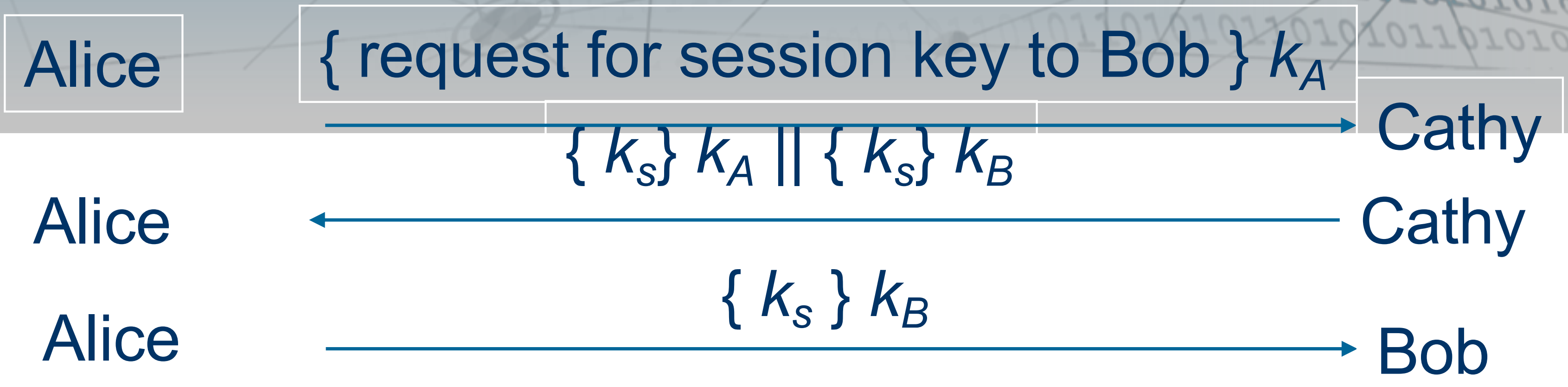
Modified From Introduction to Computer Security ©2004 Matt Bishop

The Replay Attack Problem

- How does Bob know he is talking to Alice?
  - Replay attack: Eve records message from Alice to Bob, later replays it; Bob may think he’s talking to Alice, but he isn’t
  - Session key reuse: Eve replays message from Alice to Bob, so Bob re-uses session key
- Protocols must provide authentication and defense against replay

Simple Replay Attack

Eve Observed

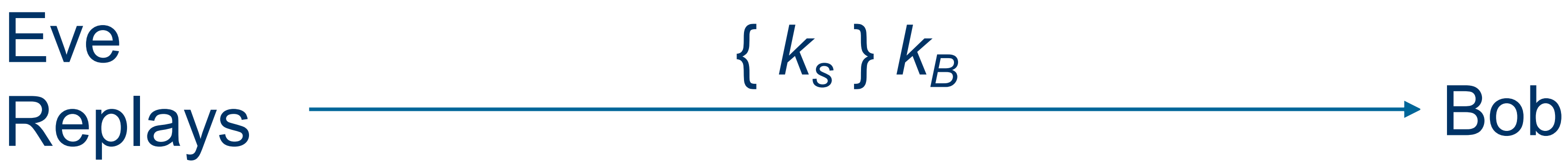


Alice { request for session key to Bob } k_A → Cathy

Alice ← $\{ k_s \} k_A \,||\, \{ k_s \} k_B$ Cathy

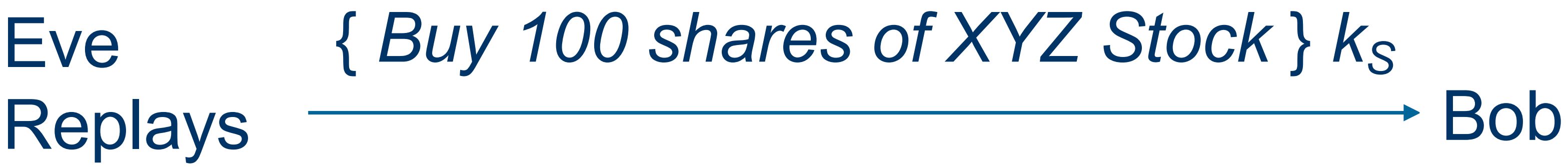
Alice $\{ k_s \} k_B$ → Bob

Two week later Eve launches a replay attack



Eve Replays $\{ k_s \} k_B$ → Bob

Bob believes this a new key exchange with Alice, it is signed by Cathy.
Eve can now replay the messages Alice previous sent to Bob using Ks.

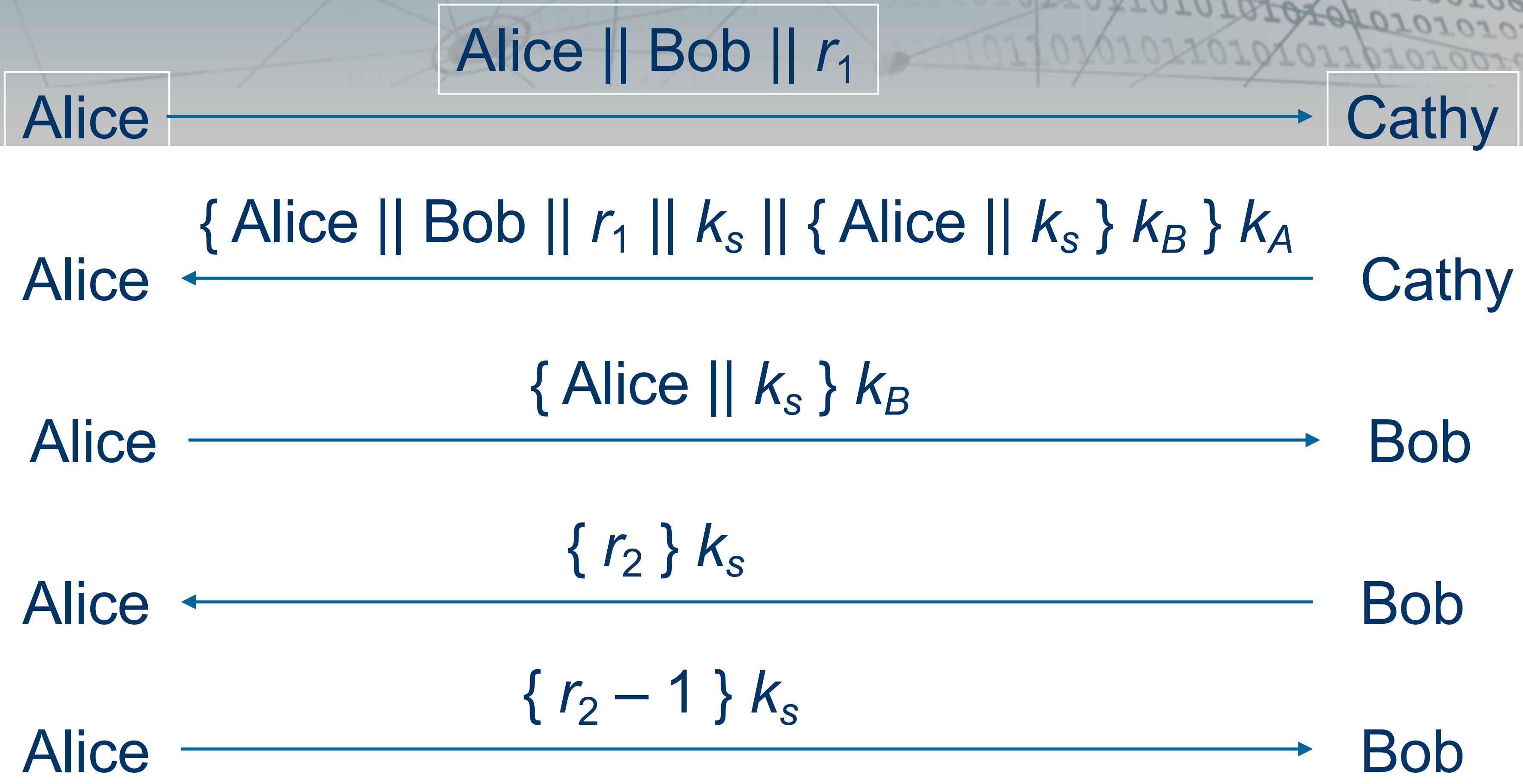


Eve Replays { *Buy 100 shares of XYZ Stock* } k_S → Bob

Bob believes this an authentic message from Alice and places the order!

Modified From Introduction to Computer Security ©2004 Matt Bishop

Needham-Schroeder



Alice || Bob || r_1

Alice → Cathy

{ Alice || Bob || r_1 || k_s || { Alice || k_s } k_B } k_A

Alice ← Cathy

{ Alice || k_s } k_B

Alice → Bob

{ r_2 } k_s

Alice ← Bob

{ $r_2 - 1$ } k_s

Alice → Bob

From Introduction to Computer Security ©2004 Matt Bishop

Protecting Against A Replay (1/2)

- Second msg: $\{ \text{Alice} \mid\mid \text{Bob} \mid\mid r_1 \mid\mid k_s \mid\mid \{ \text{Alice} \mid\mid k_s \} k_B \} k_A$
  - Enciphered using key only Alice and Cathy know
    - So Cathy enciphered it
  - Response to first message
    - As r_1 in it matches r_1 in first message
- Third message: $\{ \text{Alice} \mid\mid k_s \} k_B$
  - Alice knows only Bob can read it
    - So only Bob can derive session key from message
  - Any messages enciphered with that key are from Bob

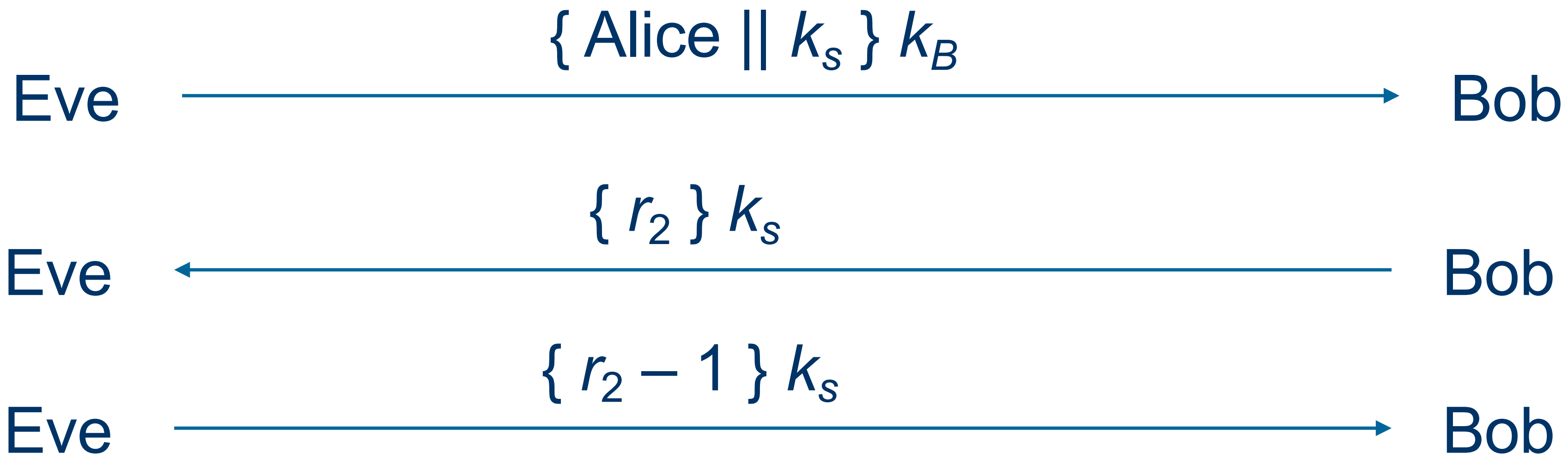
Protecting Against a Replay (2/2)

- Third message (Bob's View) : { Alice || k_s } k_B
  - Enciphered using key only Bob and Cathy know
    - So Cathy enciphered it
  - Names Alice and the session key
    - Cathy provided session key, says Alice is other party
- Fourth message. { r_2 } k_s
  - Uses session key to determine if it is replay from Eve
  - If not a replay attack, Alice will respond correctly in fifth message
  - If a replay attack attempt, Eve can't decipher r_2 and so can't respond and any guess at a response is likely to be incorrect

Modified From Introduction to Computer Security ©2004 Matt Bishop

Session Key Compromise Problem

- All keys are related to Cathy remain secret
- But Eve is able to obtain the session key
  - Maybe it was a small key, human error, etc.



Eve $\xrightarrow{\{\text{Alice} \,||\, k_s\}\, k_B}$ Bob

Eve $\xleftarrow{\{r_2\}\, k_s}$ Bob

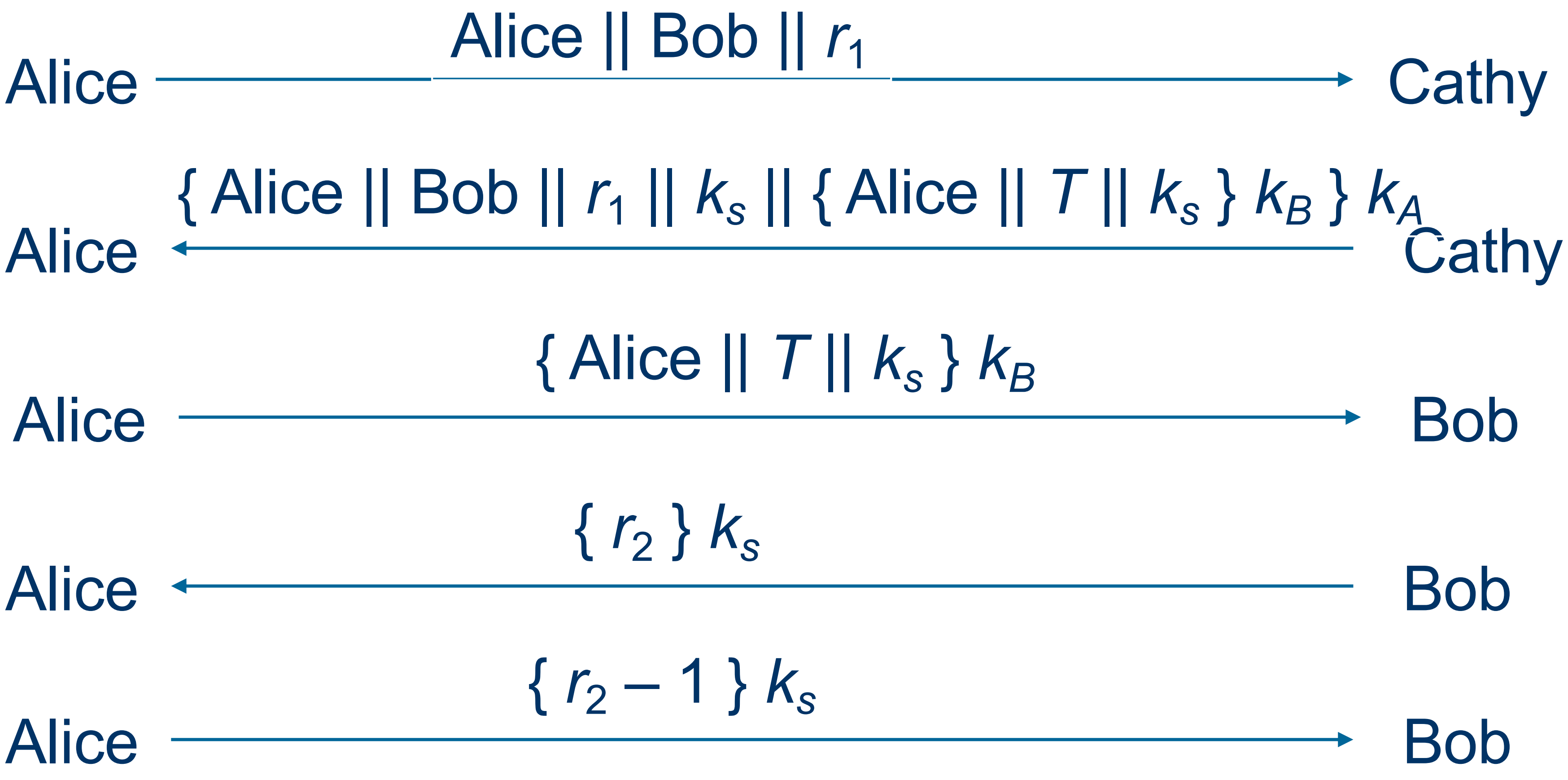
Eve $\xrightarrow{\{r_2 - 1\}\, k_s}$ Bob

Modified From Introduction to Computer Security ©2004 Matt Bishop

Solution:
Denning-Sacco Modification

- In protocol above, Eve impersonates Alice
- Problem: Eve can respond to Bob's message
  - Eve knows Ks and thus can learn r2 and encode r2-1
- Solution: use time stamp *T* to detect replay
- Weakness: if clocks not synchronized, may either reject valid messages or accept replays
  - Parties with either slow or fast clocks vulnerable to replay
  - Resetting clock does *not* eliminate vulnerability

Needham-Schroeder with Denning-Sacco Modification



Alice || Bob || r_1

Alice → Cathy

{ Alice || Bob || r_1 || k_s || { Alice || T || k_s } k_B } k_A

Alice ← Cathy

{ Alice || T || k_s } k_B

Alice → Bob

{ r_2 } k_s

Alice ← Bob

{ $r_2 - 1$ } k_s

Alice → Bob

From Introduction to Computer Security ©2004 Matt Bishop

Heilmeier Questions

- What are you trying to do? Articulate your objectives using absolutely no jargon.
- How is it done today, and what are the limits of current practice?
- What is new in your approach and why do you think it will be successful?
- Who cares? If you succeed, what difference will it make?
- What are the risks?
- How much will it cost?
- How long will it take?
- What are the mid-term and final “exams” to check for success?