CS 5430

Key Establishment

Prof. Clarkson

Spring 2017

Review

- Secure channel:
  - Bidirectional, multi-message conversations
  - Confidentiality goal: The channel does not reveal anything about messages except for their timing and size
  - Integrity goal: If Alice sends a sequence of messages m1, m2, ... then Bob receives a subsequence of that, and furthermore Bob knows which subsequence; and the same for Bob sending to Alice
- Cryptography employed:
  - Authenticated encryption to protect confidentiality and integrity
  - Message numbers to further protect integrity
  - Key derivation function to create many shared keys out of one session key
- Still need to share the session key!

Session key generation

Back to this assumption:

For now, let's assume Alice and Bob already have a single shared session key k

We need a means for Alice and Bob to generate that key…

Key establishment

Theorem [Boyd 1993]: impossible to establish secure channel between principals who do not already...

- share a key with each other, or
- separately share a key with a trusted third party, or
- have the means to ascertain a public key for each other

...i.e., you can't get something for nothing

Key establishment

Theorem [B 

ish secure channel bet ady...

- share a ke
- separately d party, or
- have the for each other

for nothing

Key establishment

- Terminology:
  - **user** is a principal who will use the generated session key for further communication
  - other **principals** might be involved but won't learn or use the key
- Key transport protocol: session key is generated by one principal then transferred to all users
- Key agreement protocol: session key is generated as a function of inputs from all users and transferred to all users

KEY ESTABLISHMENT WITH TTP

Key establishment

Let's build something "really simple"...

- a key transport protocol
- with a trusted server
- who picks the session key

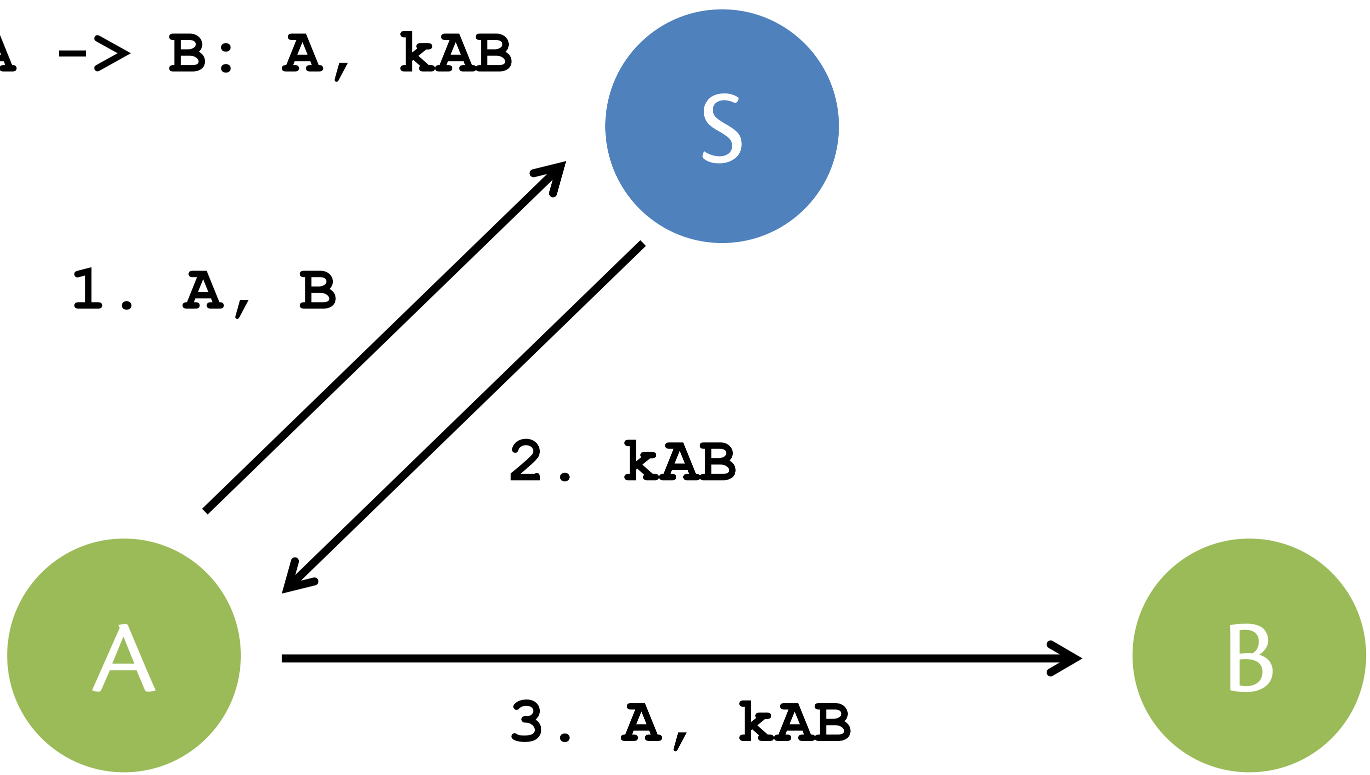
Transport protocol

- **Assume:** trusted server S with whom A and B already share a long-term key
  - A shares kAS with S
  - B shares kBS with S
- **Output:** new session key kAB shared by A and B
  - S trusted to generate key (correctly, randomly)
  - S ought to immediately forget kAB
- **Security goals:**
  1. only A and B (and S) know that key (confidentiality)
  2. (more to come...)

ATTEMPT #1

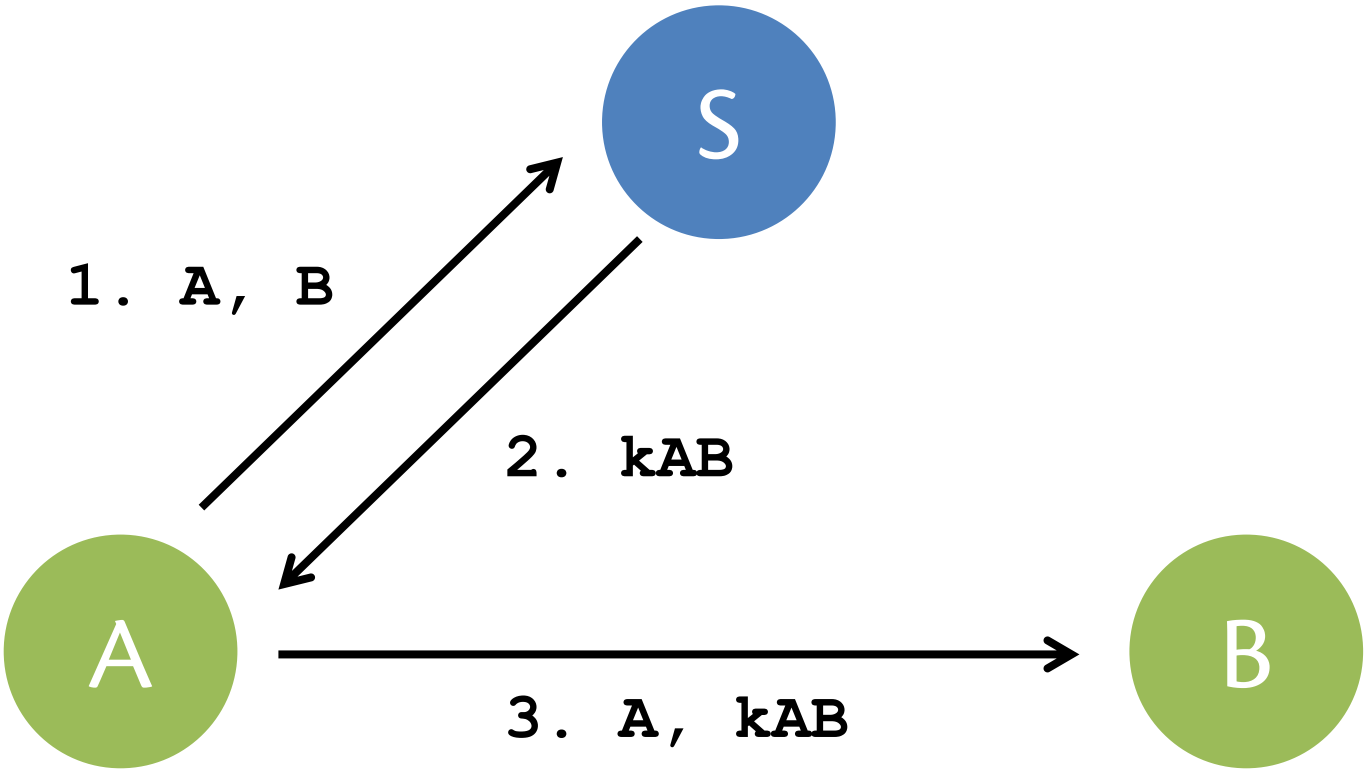
Naïve protocol

```
1. A -> S: A, B
2. S -> A: kAB
3. A -> B: A, kAB
```



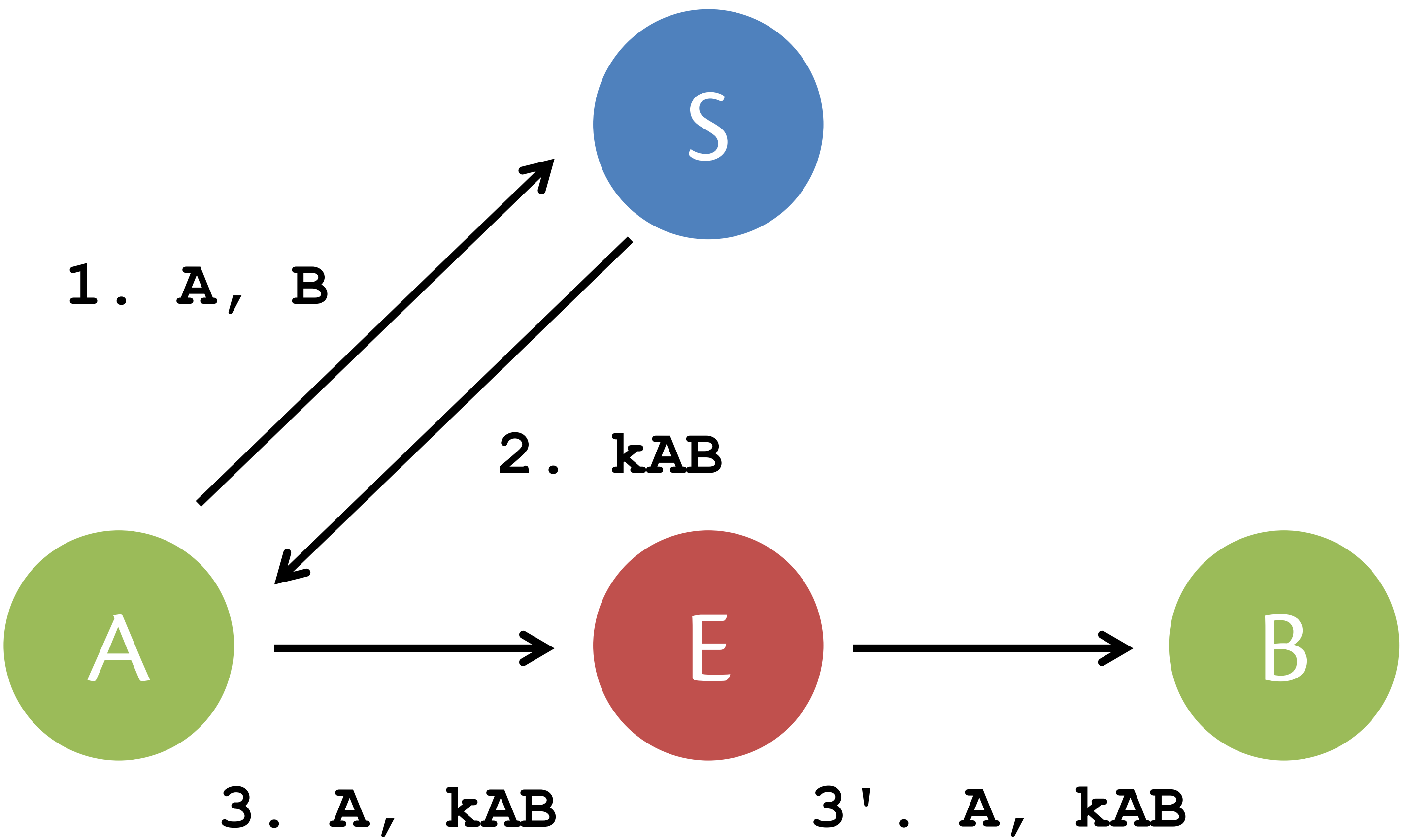
Naïve protocol

Can attacker violate conf. goal and learn kAB?



Eavesdropping attack

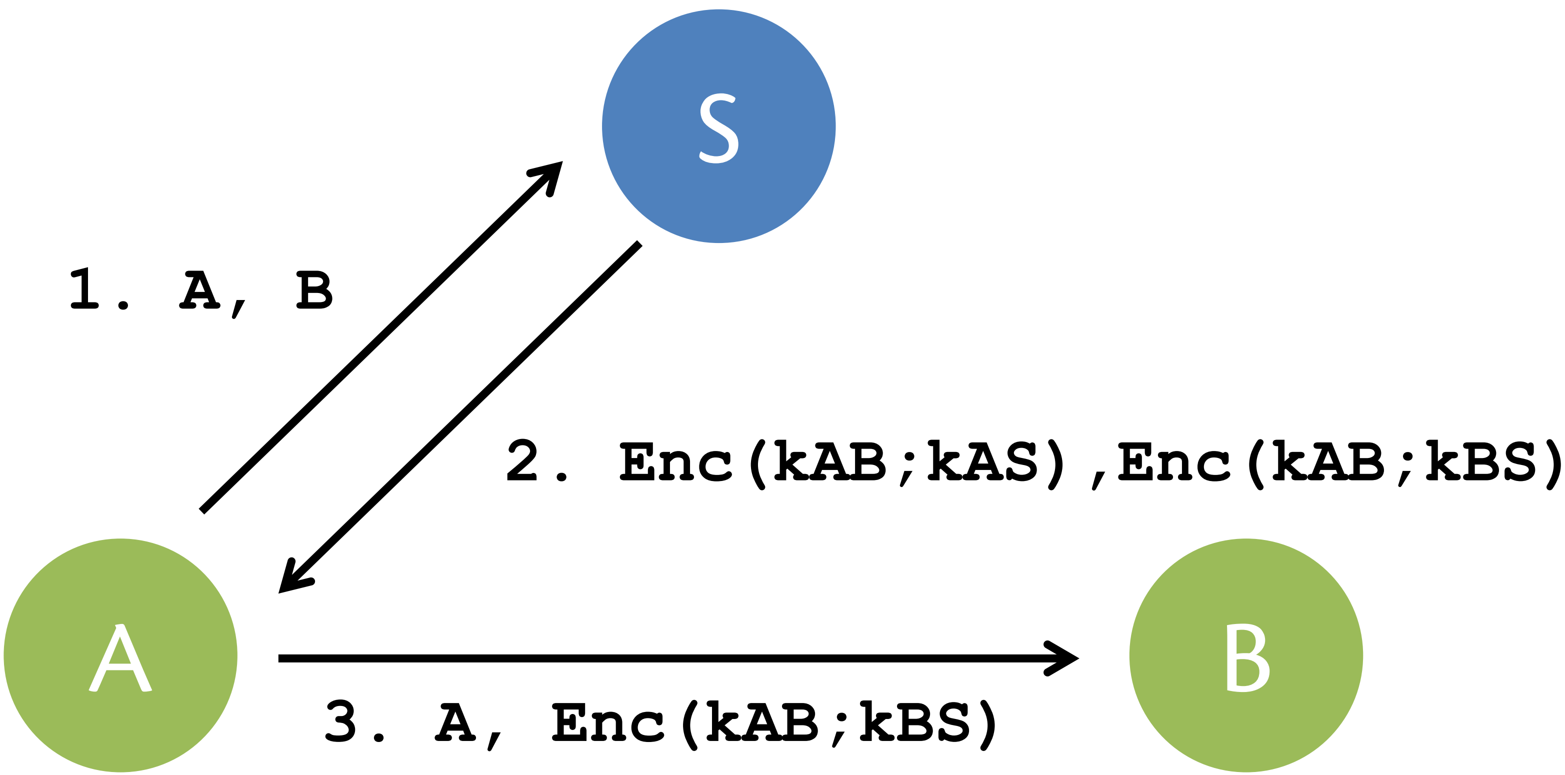
Can attacker violate conf. goal and learn kAB? Yes!



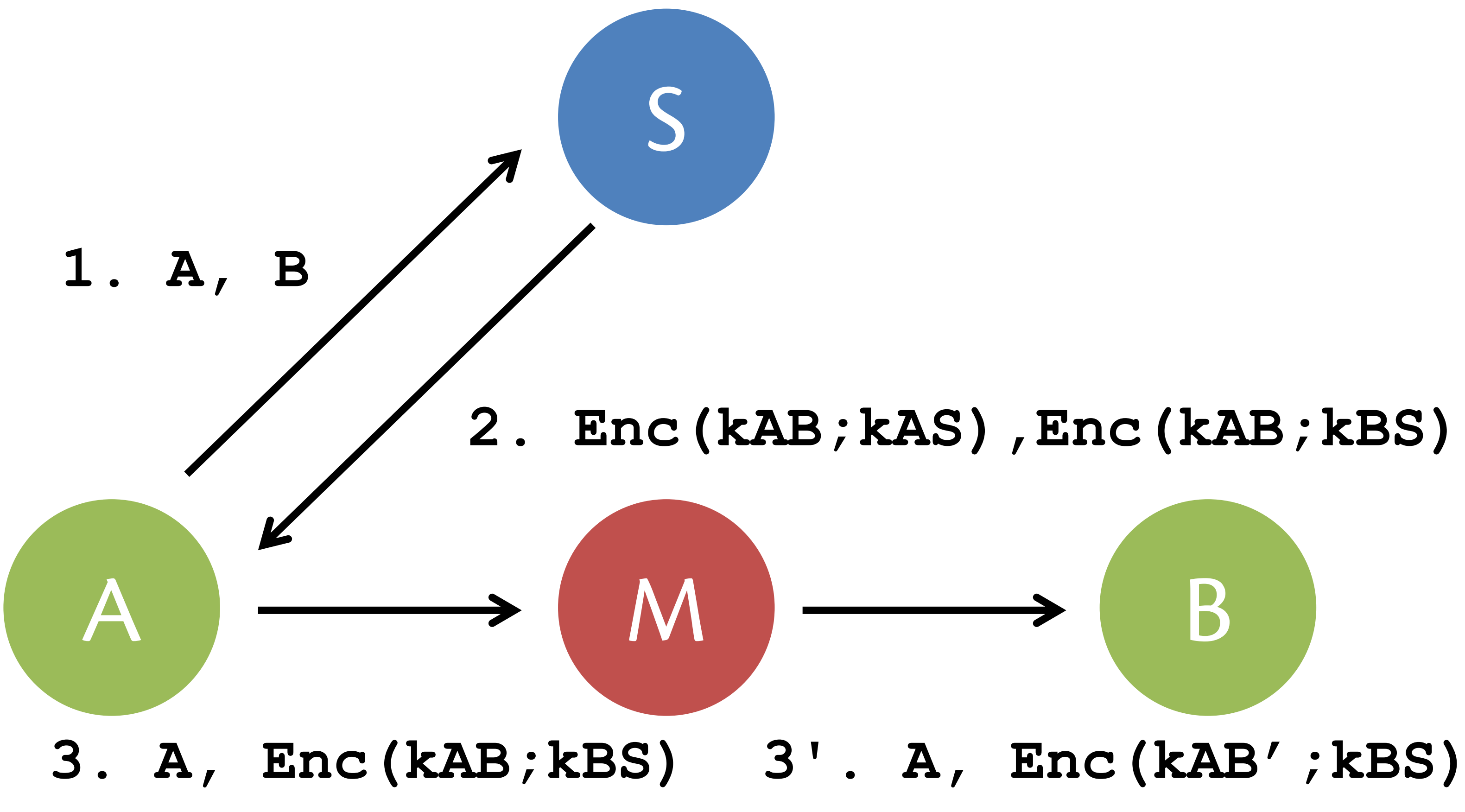
ATTEMPT #2

Countermeasure: Encryption

Key seems confidential… but do A and B understand its purpose?



Man in the middle attack



Countermeasure: Non-malleable encryption

- Non-malleable: Adversary cannot undetectably transform a ciphertext into a related ciphertext
- Degree of integrity is somewhere in-between plain-old encryption and authenticated encryption

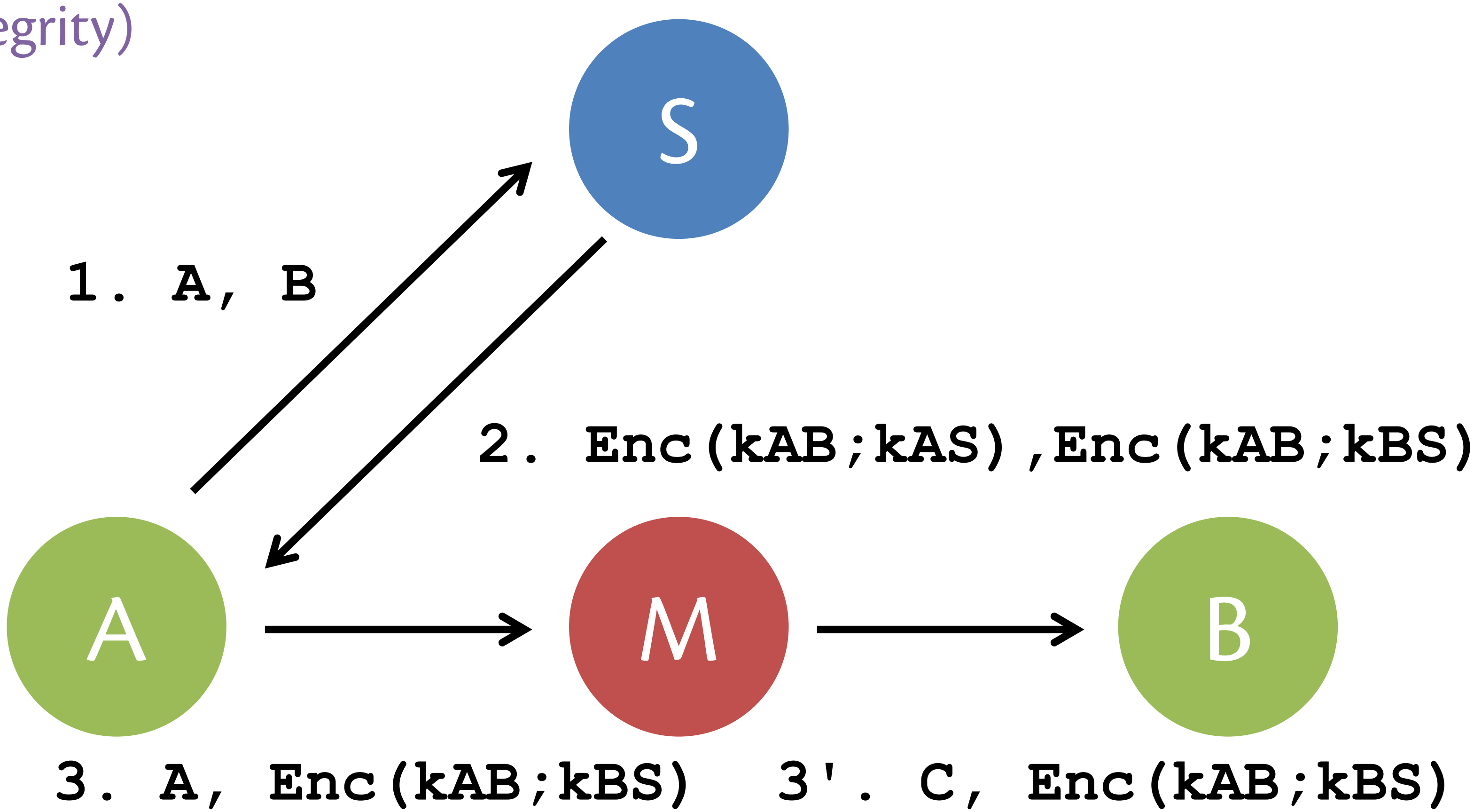
Countermeasure: Non-malleable encryption

- In the rest of this lecture, assume Enc is non-malleable
  - For symmetric schemes, the usual way to get non-malleability is with MACs, i.e., authenticated encryption
  - For asymmetric schemes, other methods possible that don't require digital signatures
    - RSA with OAEP
    - Cramer-Shoup extension of Elgamal

Another MITM attack

Key seems confidential… but do A and B understand its purpose? No!

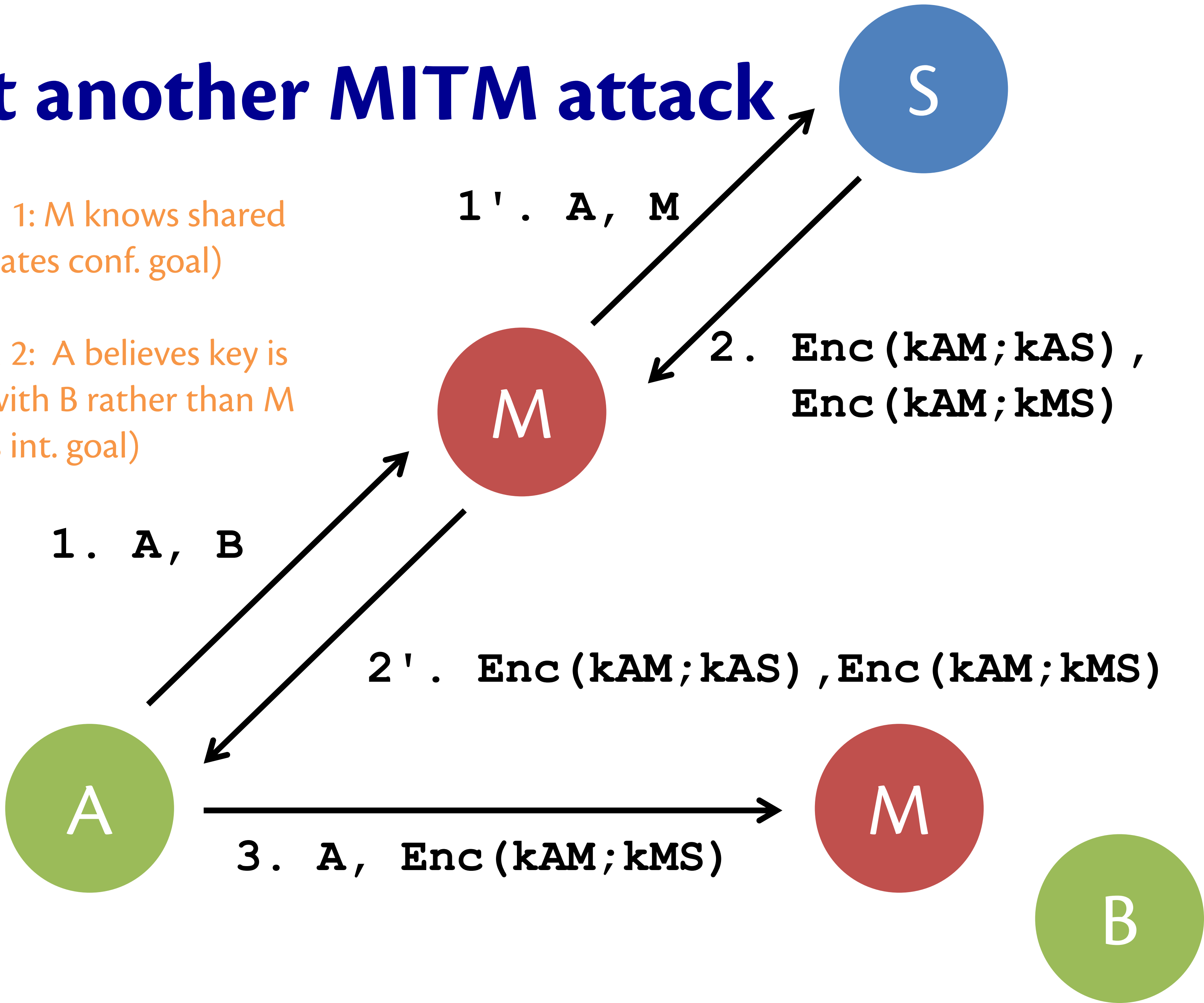
Goal: 2. Users associate correct key with correct principal identities (integrity)



Yet another MITM attack

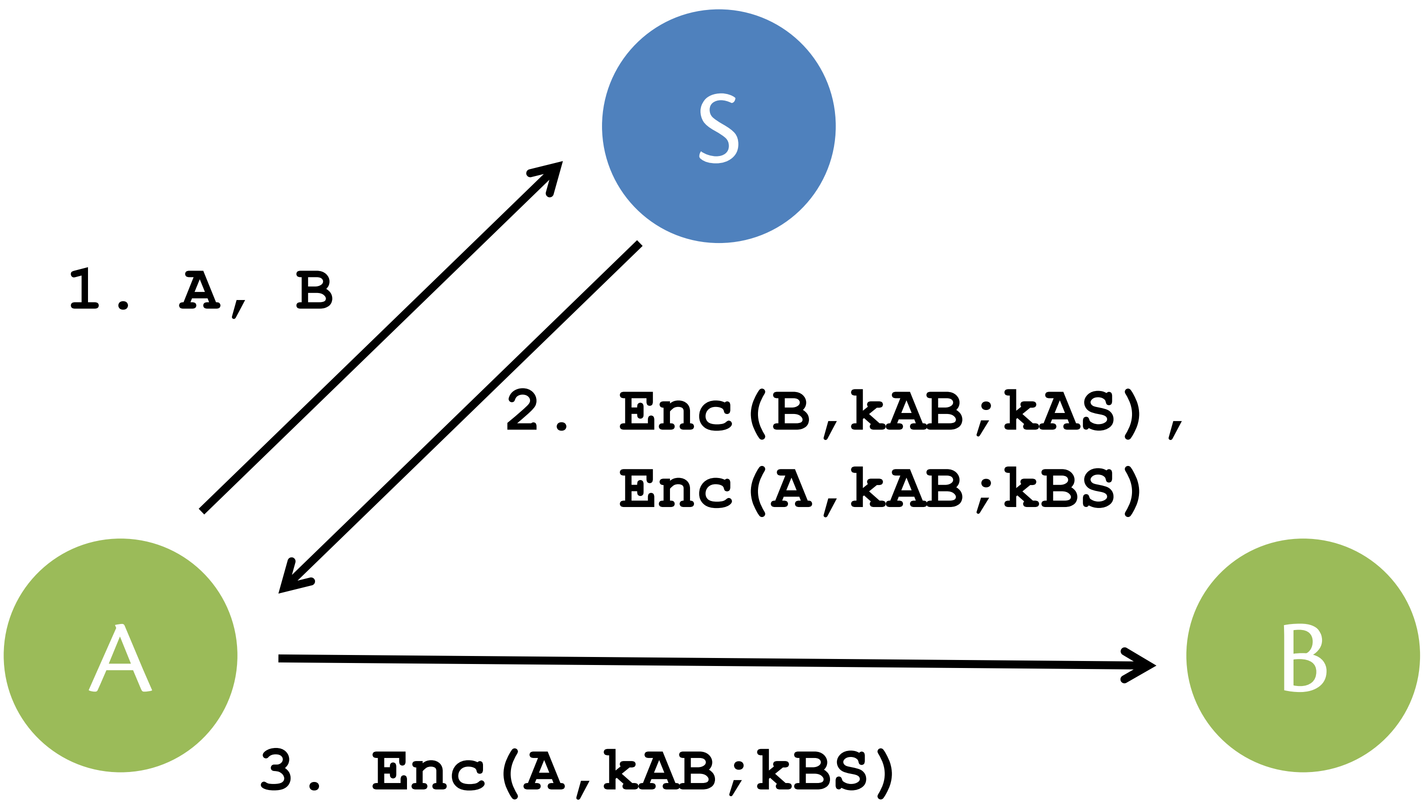
Problem 1: M knows shared key (violates conf. goal)

Problem 2: A believes key is shared with B rather than M (violates int. goal)



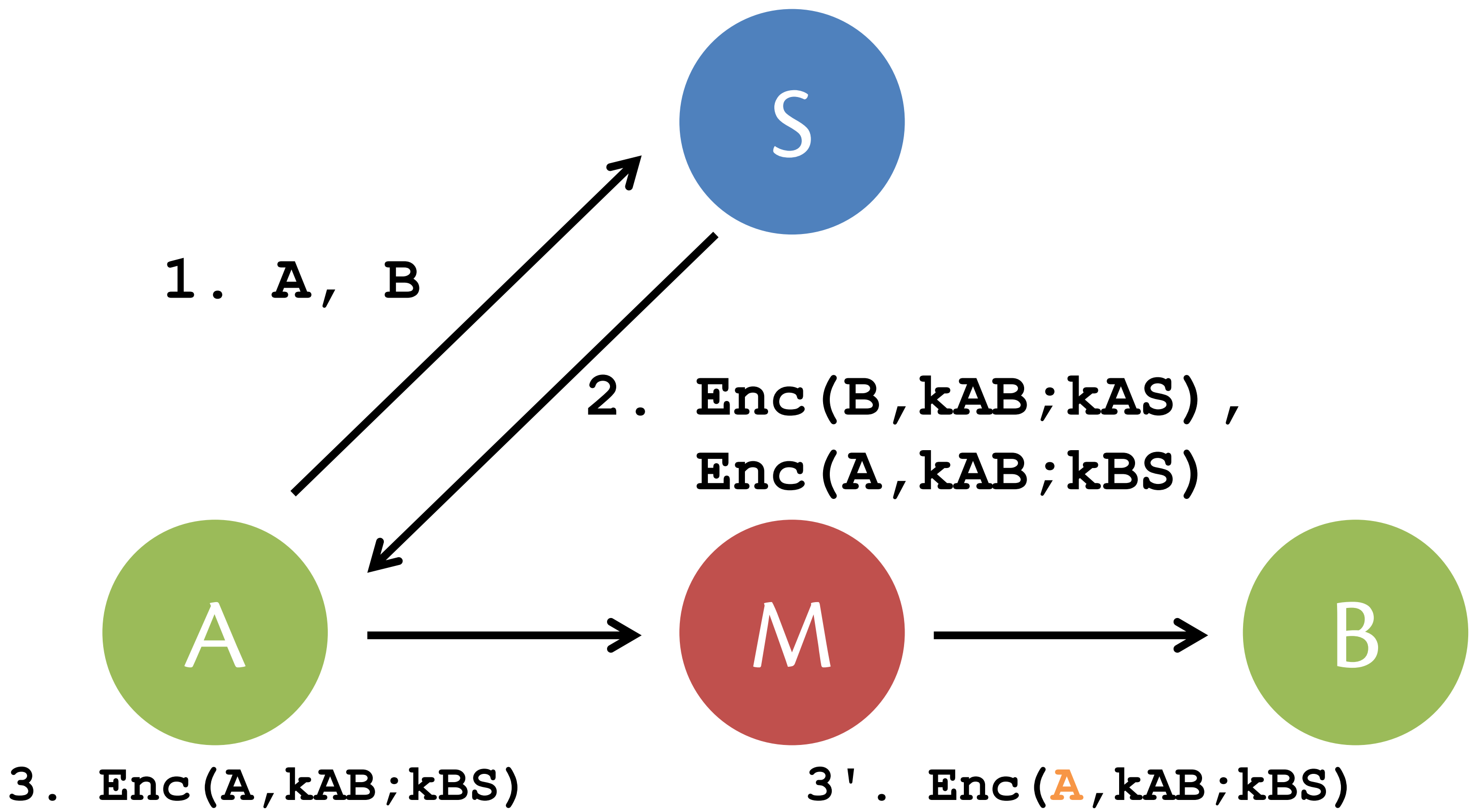
ATTEMPT #3

Countermeasure: Names



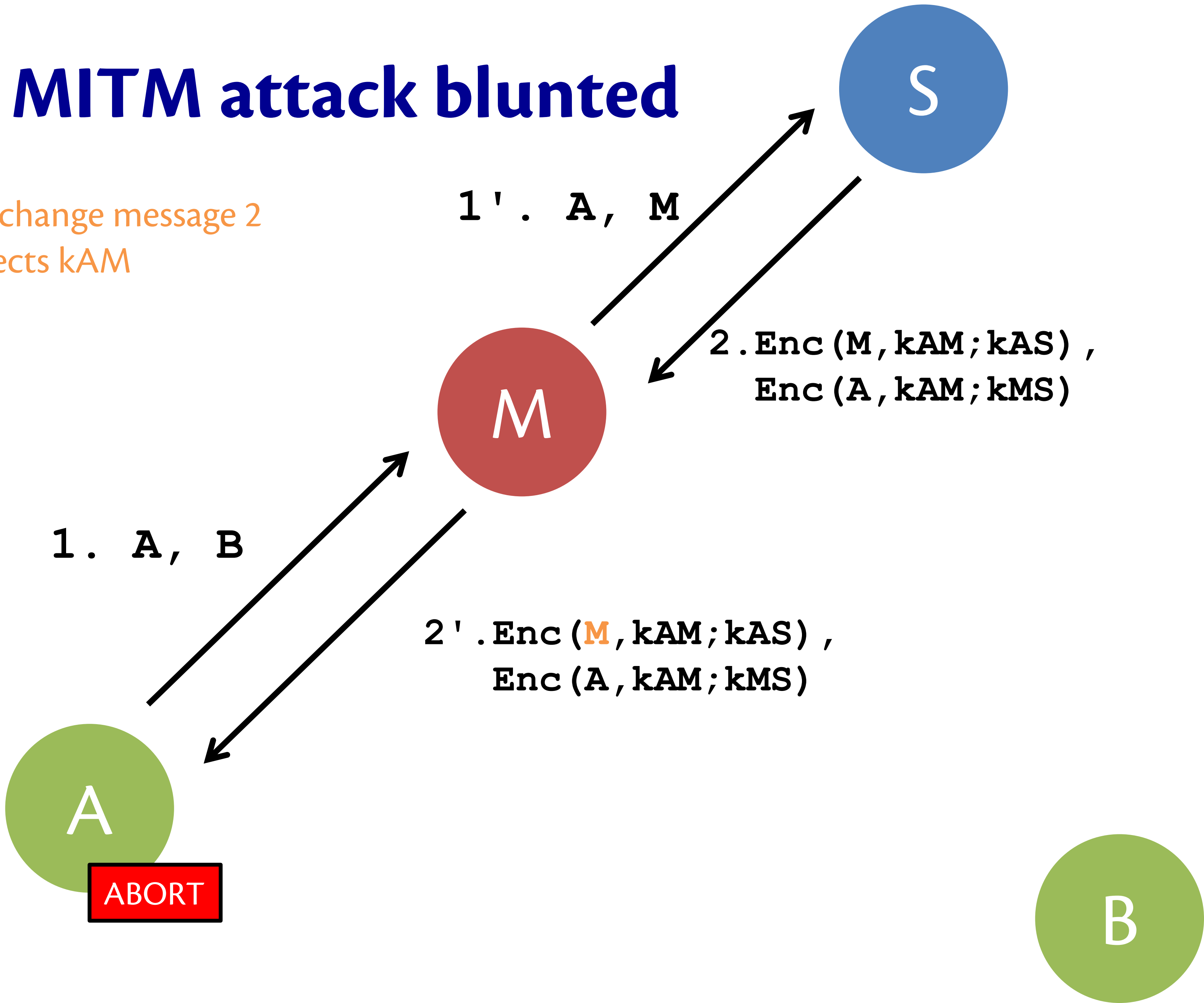
1st MITM attack blunted

M can't change name in message 3
So B correctly believes key is shared with A



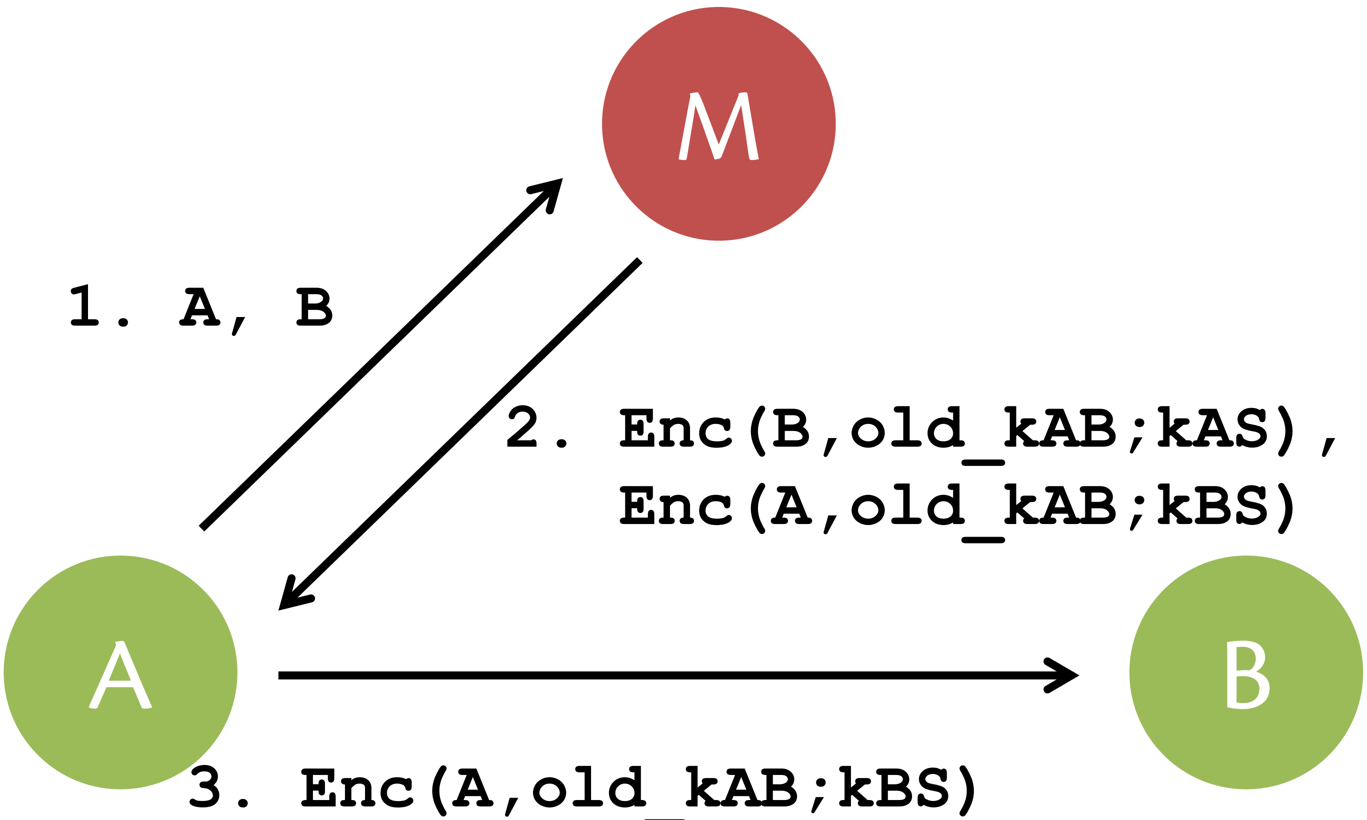
2nd MITM attack blunted

M can’t change message 2
So A rejects kAM



Replay attack

Goal: 3. the session key is *fresh* (integrity)



Secrets do leak

- "Truth will out" –Shakespeare, *Merchant of Venice*
- "For nothing is hidden that will not be disclosed, nor is anything secret that will not become known and come to light." –Luke 8:17
- **Goal 4:** protect new messages from disclosure if old session key does become known to adversary (conf.)
  - Old messages will be disclosed
  - New messages need not be
- Is it likely that adversary learns session key kAB but not any long-term shared keys?
  - Session keys typically stored only in memory
  - Long-term keys might be stored elsewhere

Implementing key erasure

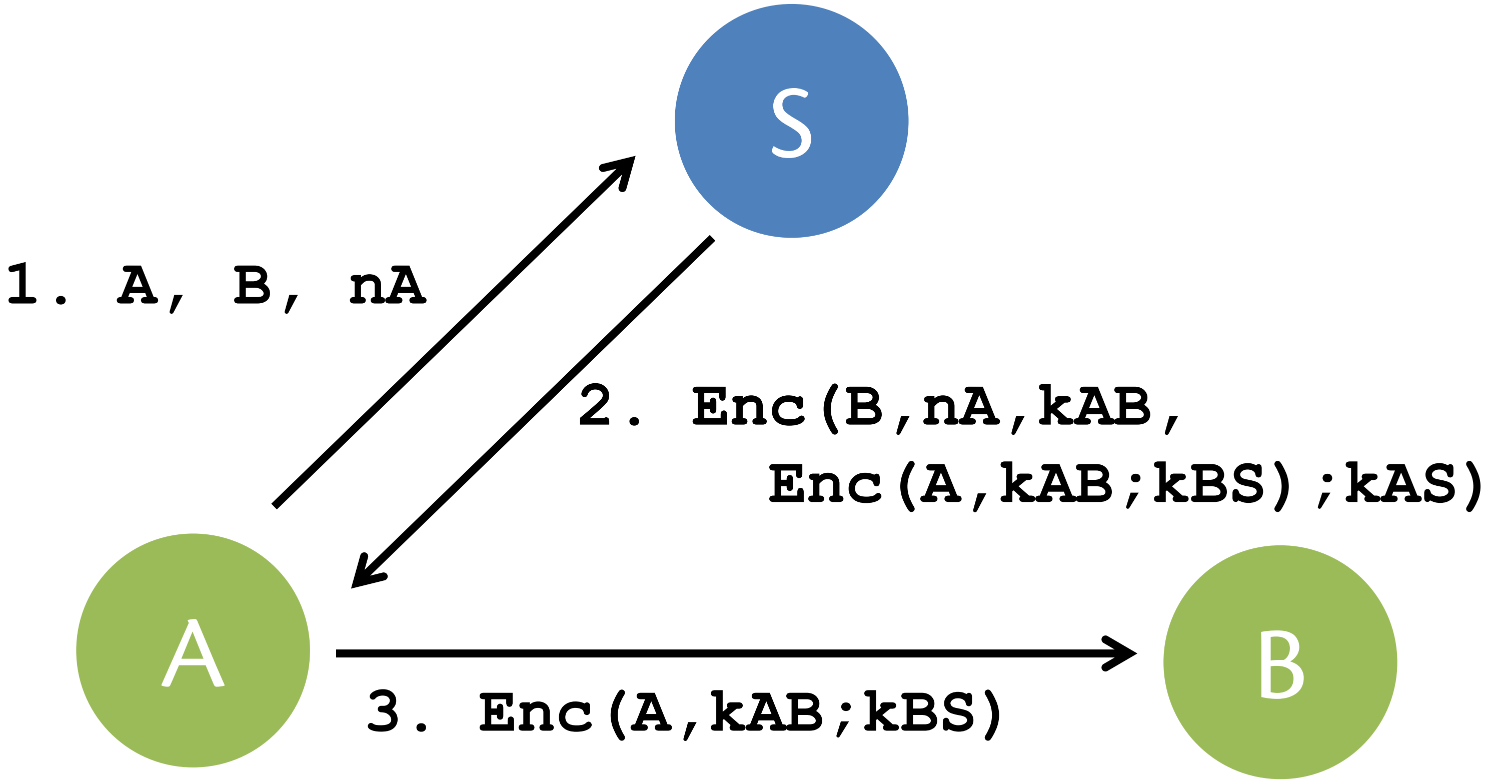
- Never assume that deallocation or garbage collector will make keys inaccessible
- Zero out arrays containing keys, passwords, other secrets; if you can!
  - High-level languages make it quite hard
  - Compilers might optimize away
  - Registers and memory can end up in swap files on disk
  - DRAM can be cooled, physically extracted, and read

Countermeasure: Challenge-Response

- (back to that replay attack with old keys)
- Challenger issues question
- Responder gives answer
- Example: *From Russia with Love*
  - Unfortunately, that *static challenge* can be replayed
- So crypto protocols use nonces
  - Principals contribute their own unique nonce to be convinced of *freshness*

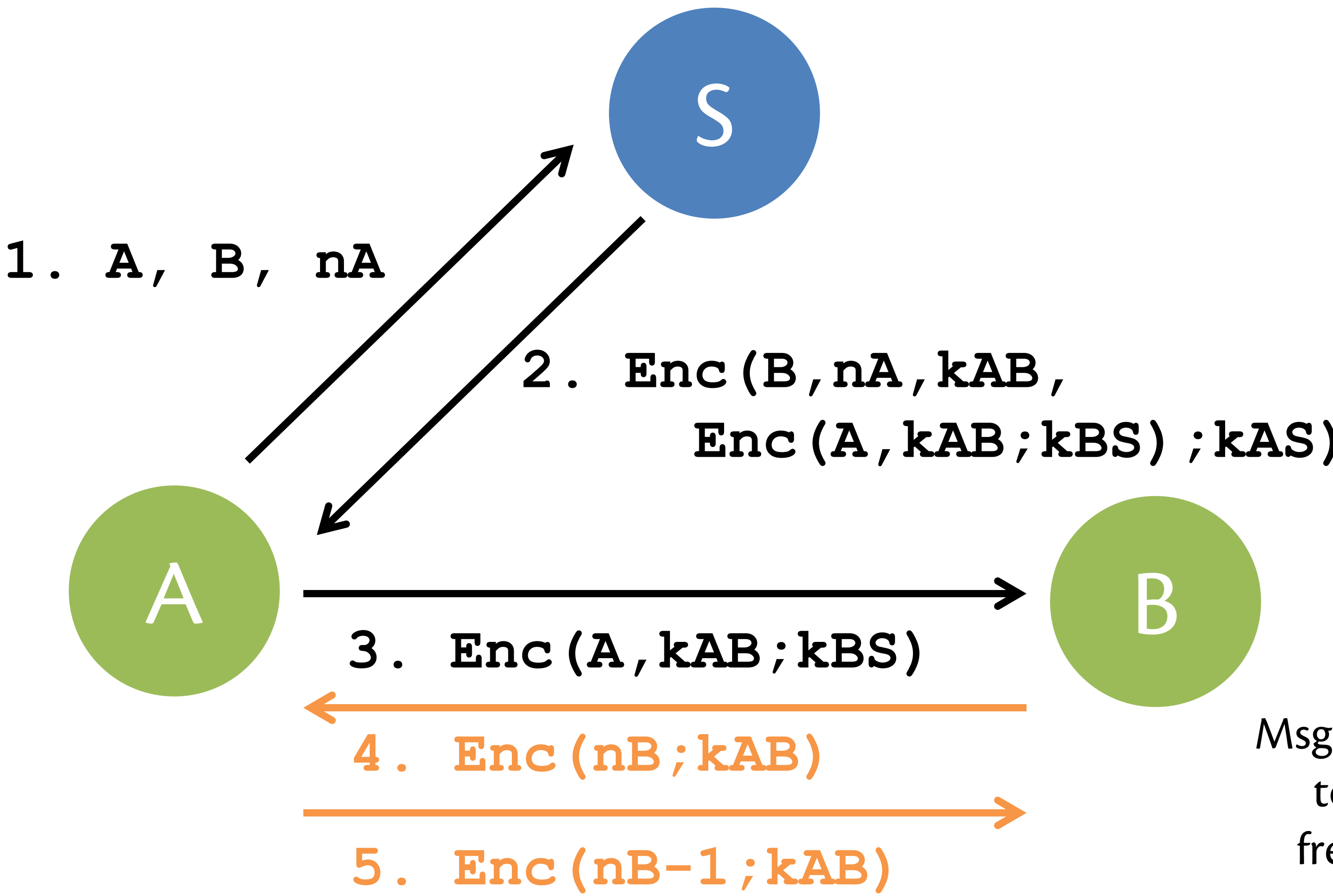
ATTEMPT #4

Countermeasure: Nonces



Convinces A that key is fresh, but not B…

Needham & Schroeder 1978

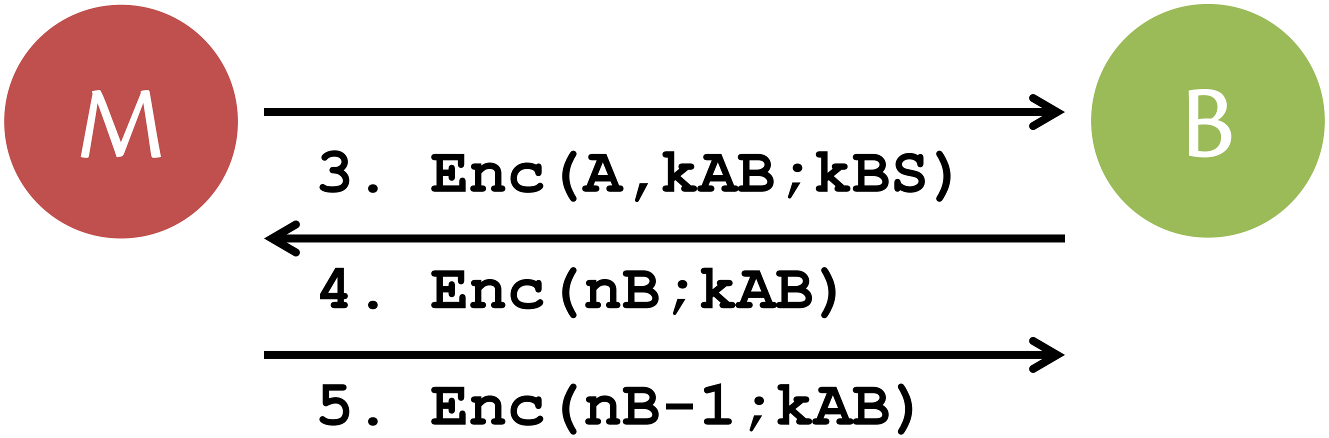


Msg 4&5 meant to guarantee freshness, but don't…

Replay attack

Assume:

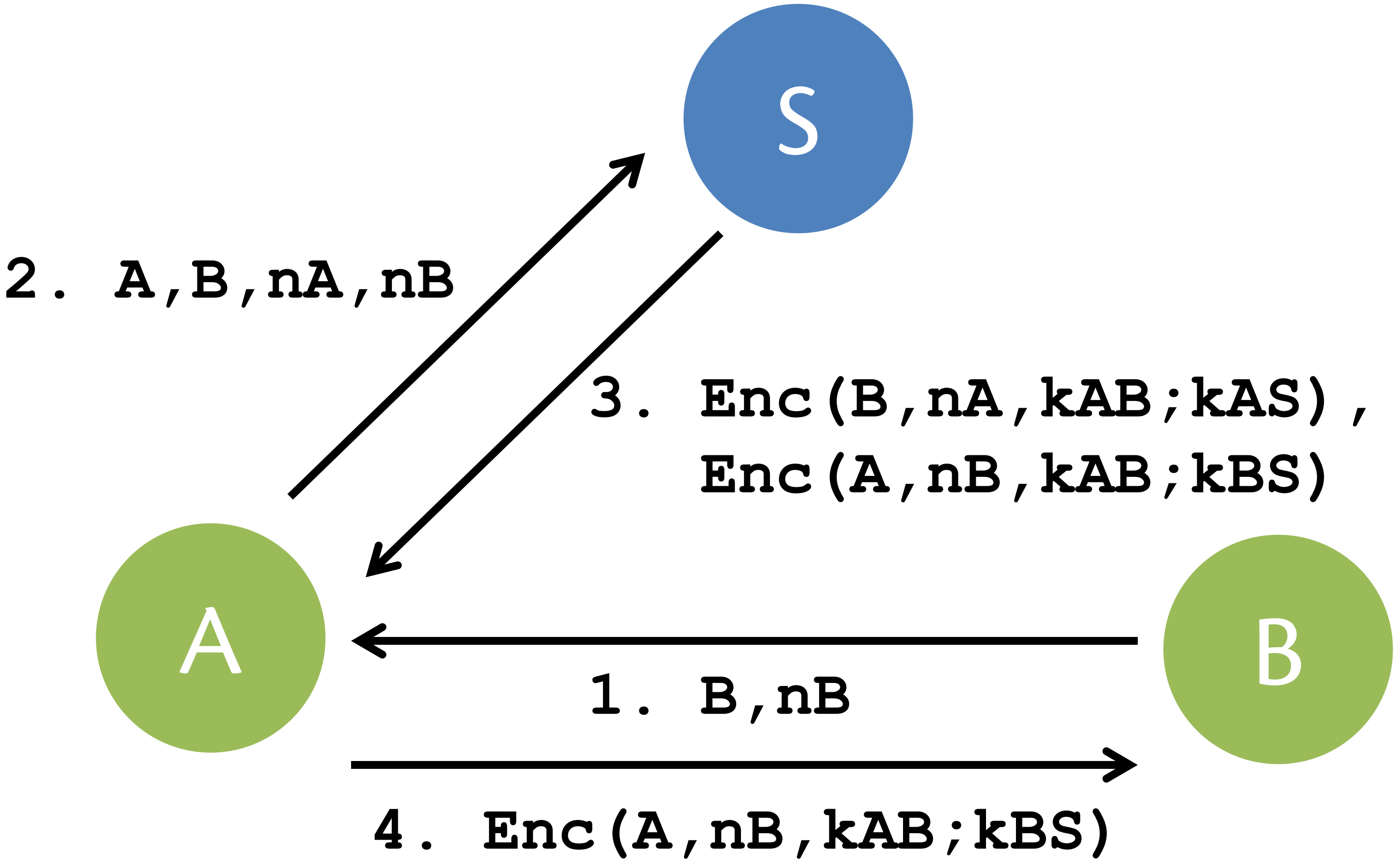
- M captures message 3, and
- M learns kAB



FINAL ATTEMPTS

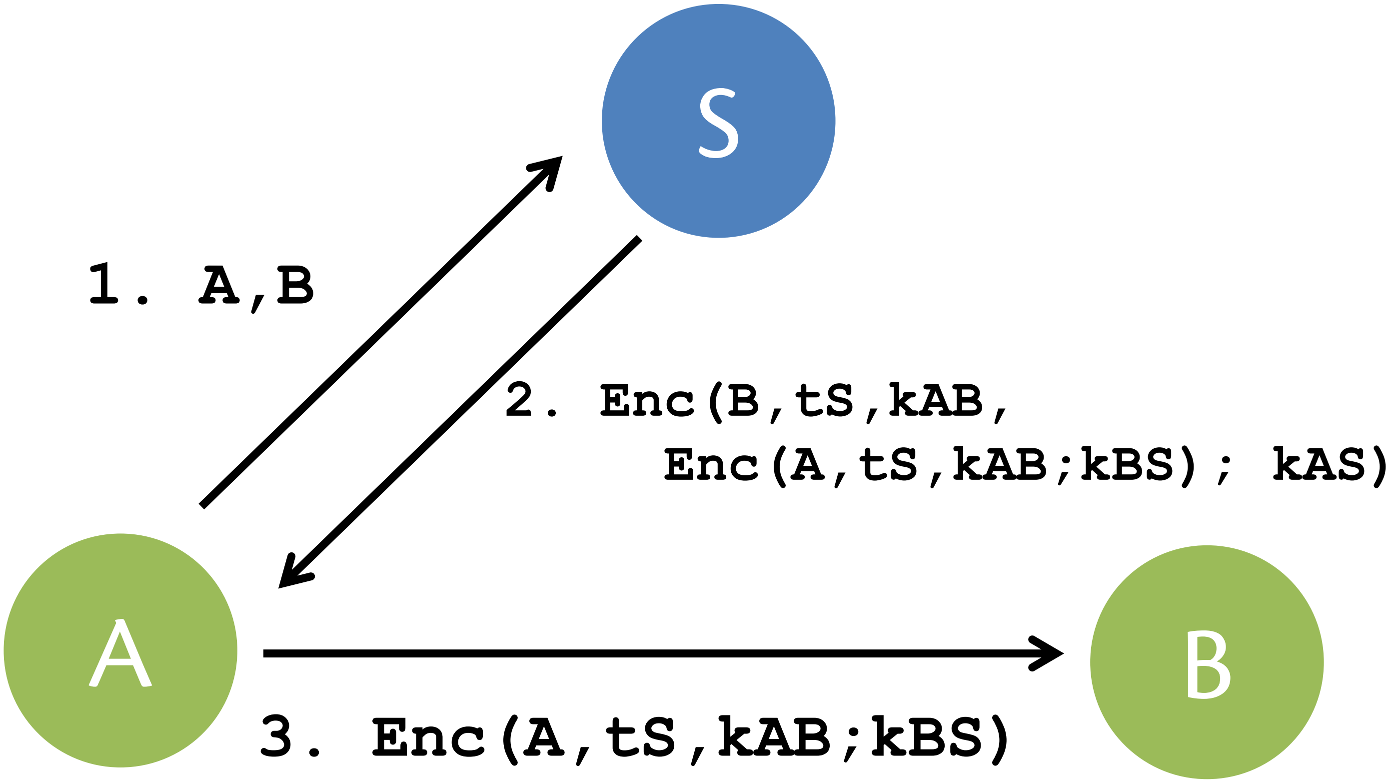
Bauer et al. 1983

Solution 1: submit nonces from both users to S



Denning & Saco 1981

Solution 2: use synchronized clocks and timestamps as nonce



tS is time at server S. A and B reject any message that is too old.

Wrapup: Secure channel

- Used authenticated encryption, message numbers, key derivation function, key establishment protocol
- Now we can have secure conversations!

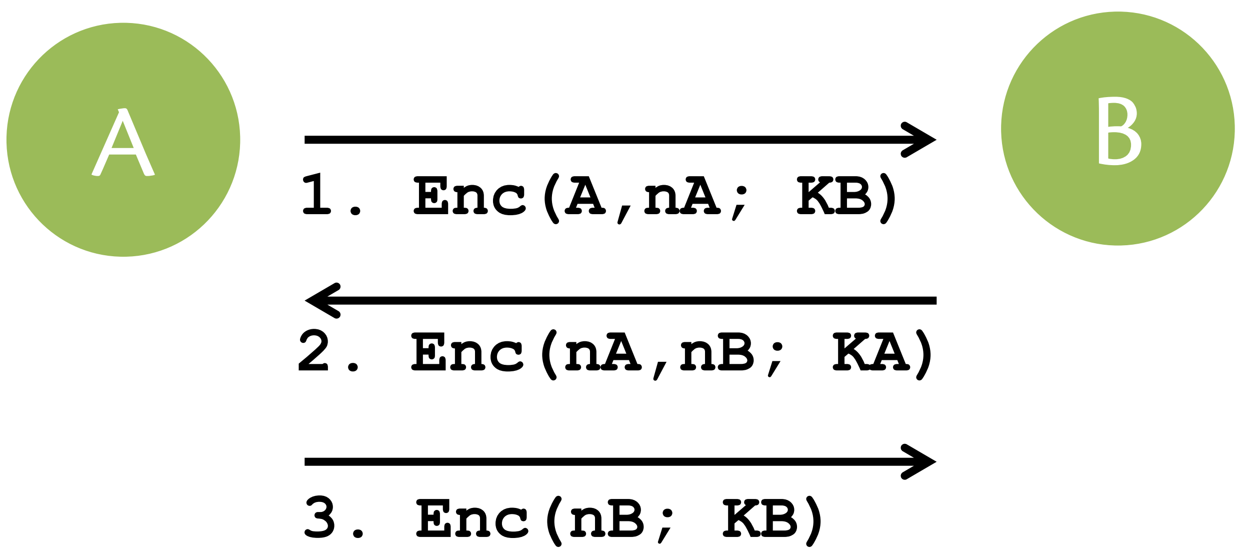
Lessons learned

- Designing simple cryptographic protocol is hard
  - Attacks aren't obvious
  - Published protocols later found to be flawed
- Goals aren't immediately obvious
  - We ended up with four
  - There are many more contemplated in literature

KEY ESTABLISHMENT WITH PUBLIC KEYS

Needham & Schroeder 1978

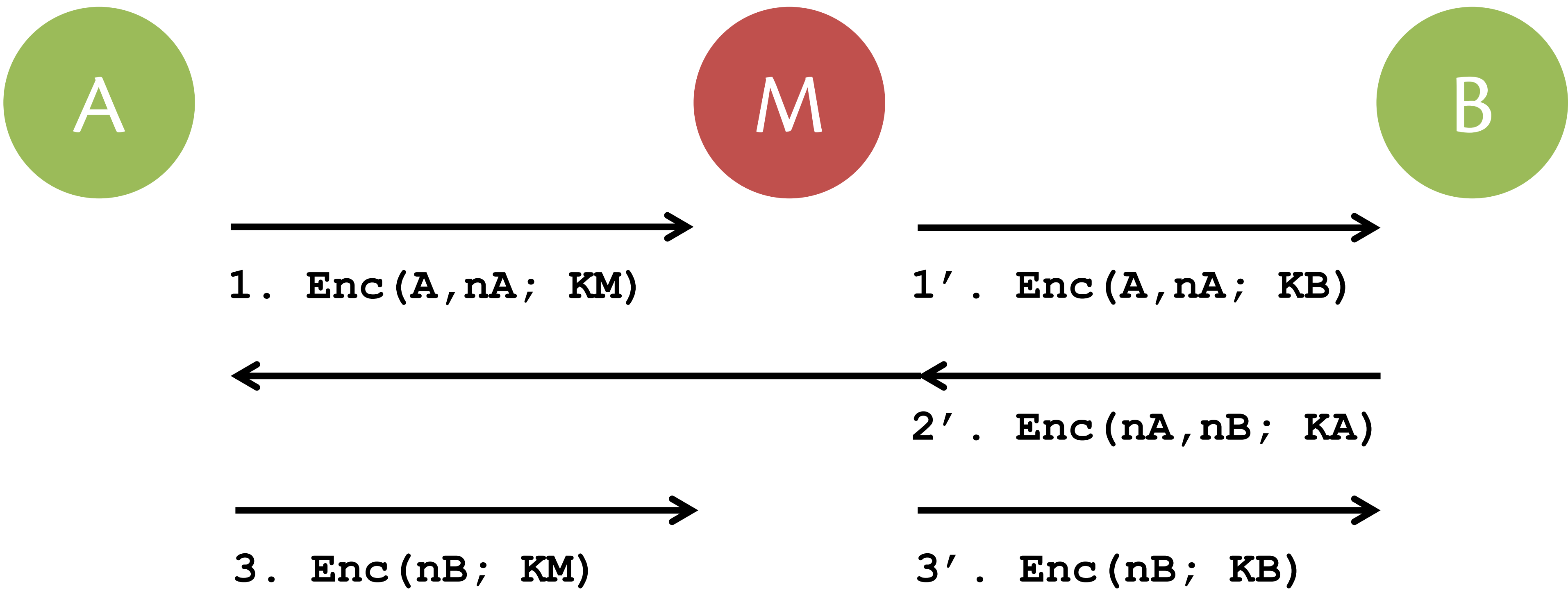
Assume: A and B already have key pairs (KA,kA), (KB,kB),
And public keys are already known to both



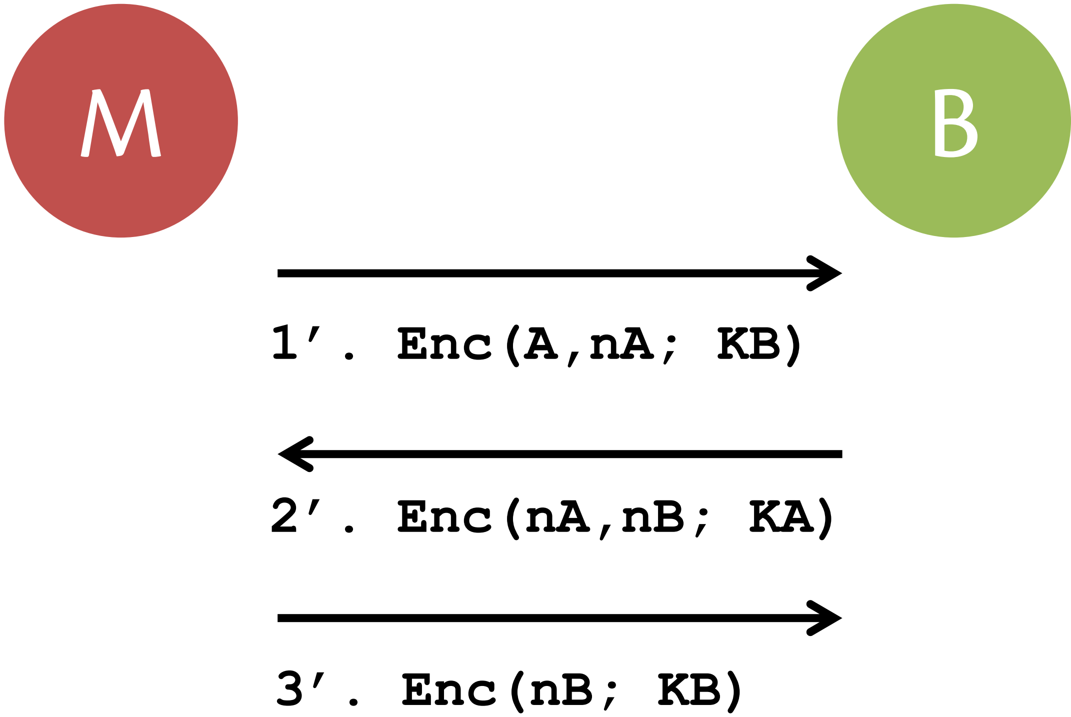
From **`nA`** and **`nB`** derive a key, e.g., **`H(nA,nB)`**

* Still need non-malleable encryption not plain-old encryption

MITM attack



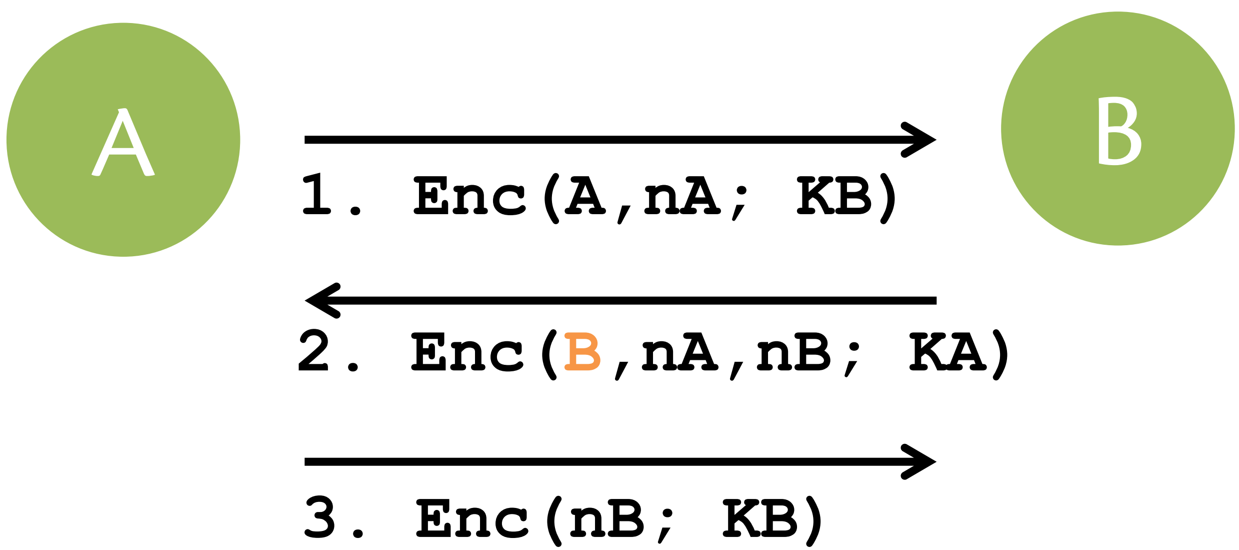
MITM attack



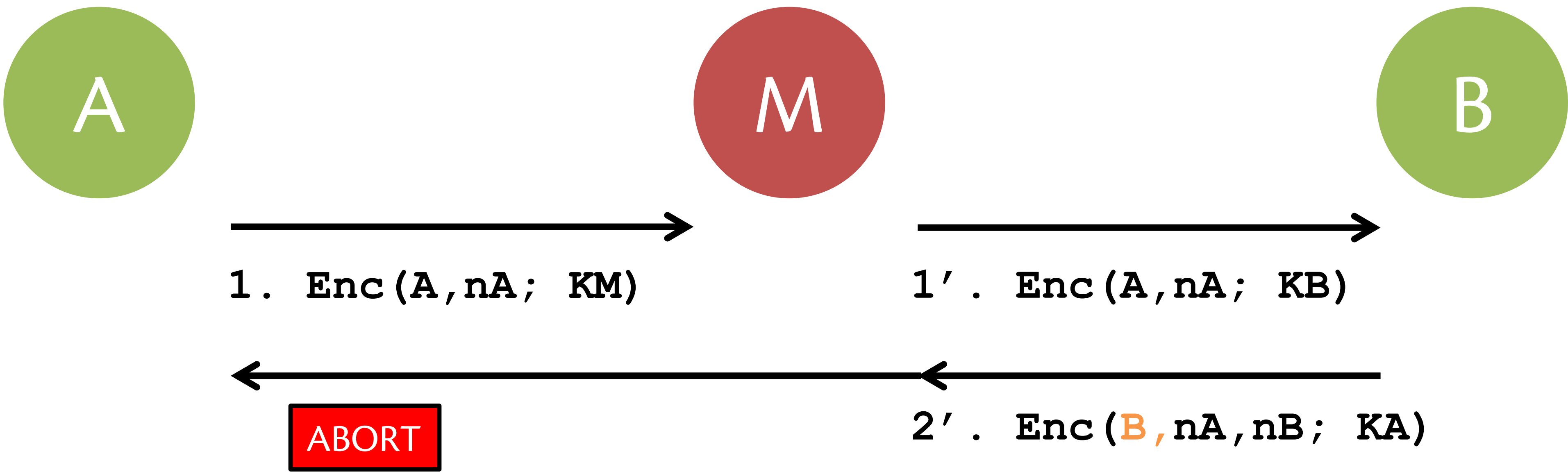
M just impersonated A!

Countermeasure: Names

Attack and fix published in [Lowe 1996]
Fixed protocol known as Needham-Schroeder-Lowe



MITM attack blunted



KEY ESTABLISHMENT FROM NOTHING

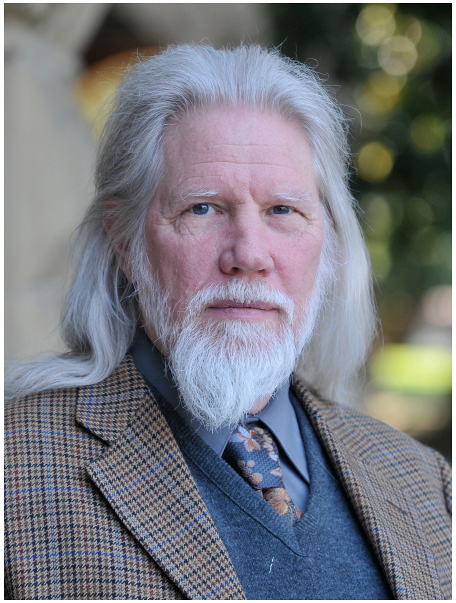
Diffie-Hellman(-Merkle)

- Key agreement protocol [1976]
  - Basis of many later protocols
  - Still available in SSL
  - No free lunch: establishes key but without any authentication of principals
  - Like having a secure telephone line to an unknown person
- Metaphor based on colors:

https://www.youtube.com/watch?v=YEBfamv-_do&feature=youtu.be&t=138

Whitfield Diffie and Martin Hellman

2015 Turing Award Winners



b. 1944

b. 1945

For critical contributions to modern cryptography.

The ability for two parties to communicate privately over a secure channel is fundamental for billions of people around the world. On a daily basis, individuals establish secure online connections with banks, e-commerce sites, email servers and the cloud. Diffie and Hellman's groundbreaking 1976 paper, "New Directions in Cryptography," introduced the ideas of public-key cryptography and digital signatures, which are the foundation for most regularly-used security protocols on the Internet today.

Upcoming events

- [today] A2 due, A3 out

You can't always get what you want.

But…sometimes you get what you need.

–The Rolling Stones