Practical Cryptographic Systems

Protocols II

Instructor: Matthew Green

Housekeeping

- Midterm on Monday
- This class will be review

News?


1:38
Signal
5G
Post
Elon Musk
@elonmusk
Follow
Good
Inevitable West @Inevitab... · 20h
🚨 BREAKING: Apple to take legal action against the UK government after they demanded to view encrypted user data

Review

- Overview of materials
  - Everything we've discussed in class (slides and notes)
  - Readings through today
  - Major papers (to be discussed in a second)

Classical cryptography

- Substitution ciphers
- Vigenere
- Weaknesses and attacks
- One-Time Pad (variable length one time pad too!)

Symmetric encryption

- Definitions of security
- Block ciphers (e.g., AES, DES, no you don’t need to know their design)
- Stream ciphers (e.g., ChaCha)
- MACs
- Authenticated encryption
- Hash functions (definitions of security, what they do)

Basic number theory

- What is a cyclic group
  - What is the group Z*p and how is it constructed
  - What’s a generator, what’s the order of a group
  - What’s the Discrete Log and the Diffie-Hellman “problem”
- Basics of RSA setting

Asymmetric Cryptography

- Key exchanges
  - Diffie-Hellman key exchange
  - MITM attacks
- Public key encryption
  - DH / Elgamal
  - RSA
- Signatures & Certificates

Protocols

- What is a protocol?
- What is SSL/TLS
  - What does it do, how does it work
  - How does negotiation work
  - Overview of the protocol from negotiation to symmetric encryption

Papers I care a lot about

- Mining your Ps and Qs
- Wagner/Schneier on TLS/SSL
- Imperfect Forward Secrecy (just the ideas!)
- Twenty Years of attacks on RSA

Case study: Apple iMessage

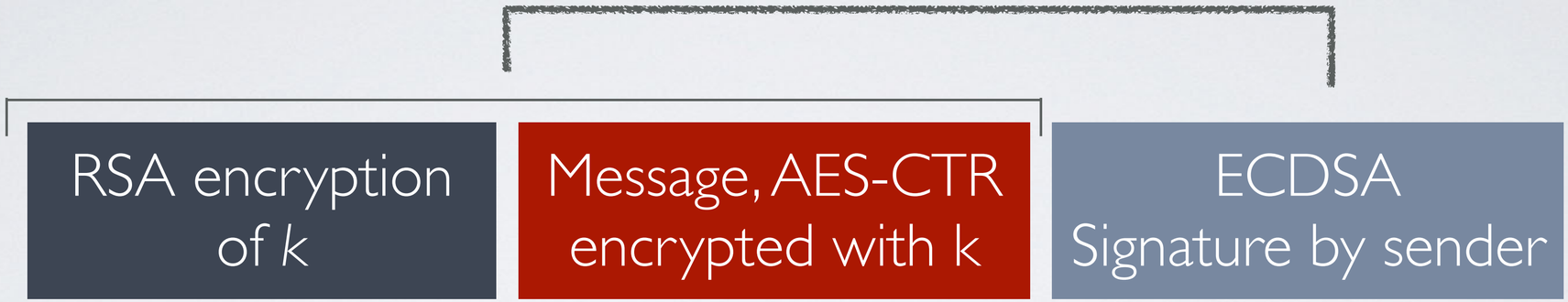
- **<u>Not</u> the most important security protocols on the Internet**
  - But pretty important to real people
  - Once you have messaging, you can build inter-device communications…

iMessage: Encryption

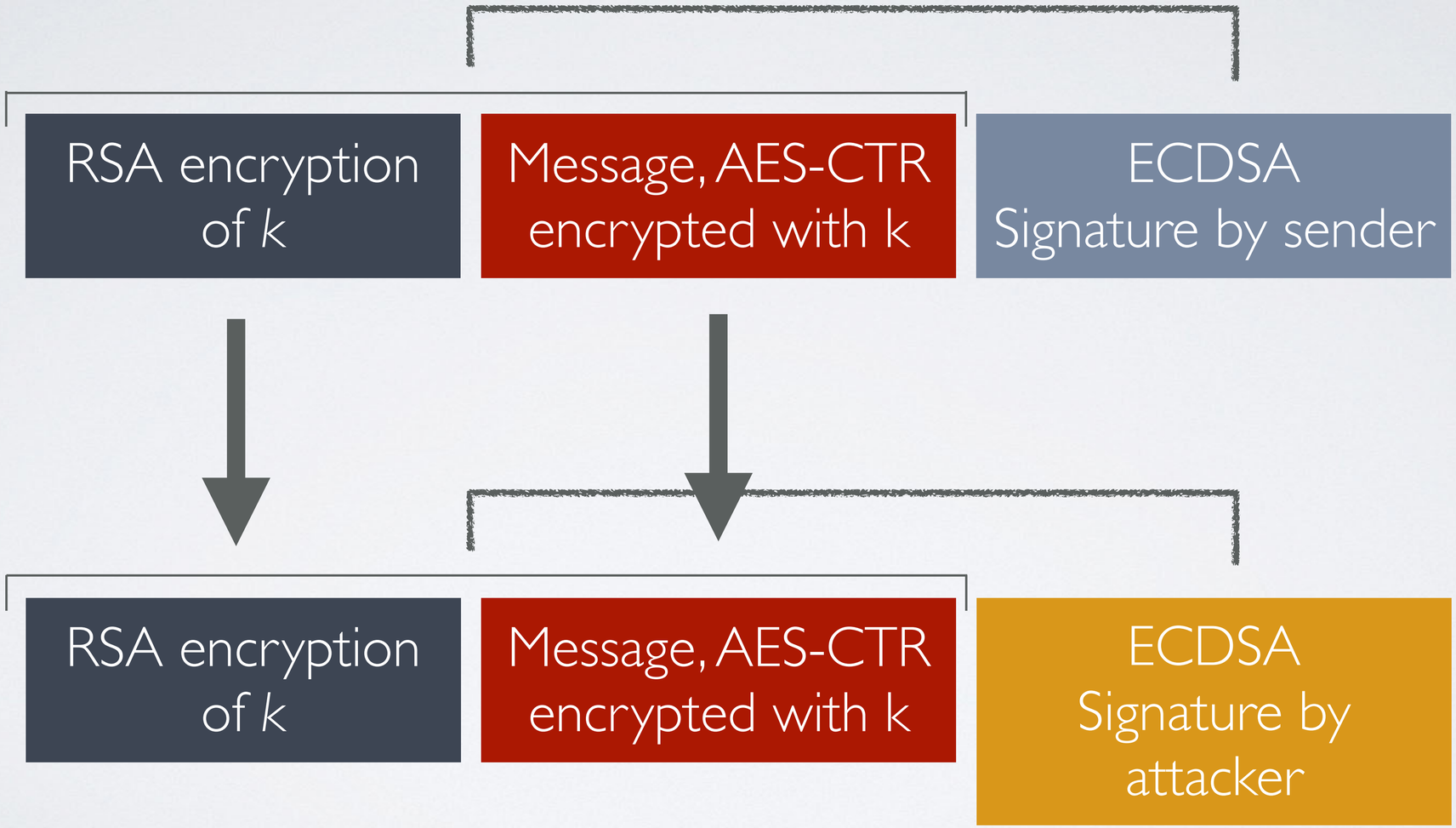
When a user turns on iMessage on a device, the device generates two pairs of keys for use with the service: an RSA 1280-bit key for encryption and an ECDSA 256-bit key on the NIST P-256 curve for signing. The private keys for both key pairs are saved in the device's keychain and the public keys are sent to Apple's directory service (IDS), where they are associated with the user's phone number or email address, along with the device's APNs address.

The user's outgoing message is individually encrypted for each of the receiver's devices. The public RSA encryption keys of the receiving devices are retrieved from IDS. For each receiving device, the sending device generates a random 128-bit key and encrypts the message with it using AES in CTR mode. This per-message AES key is encrypted using RSA-OAEP to the public key of the receiving device. The combination of the encrypted message text and the encrypted message key is then hashed with SHA-1, and the hash is signed with ECDSA using the sending device's private signing key. The

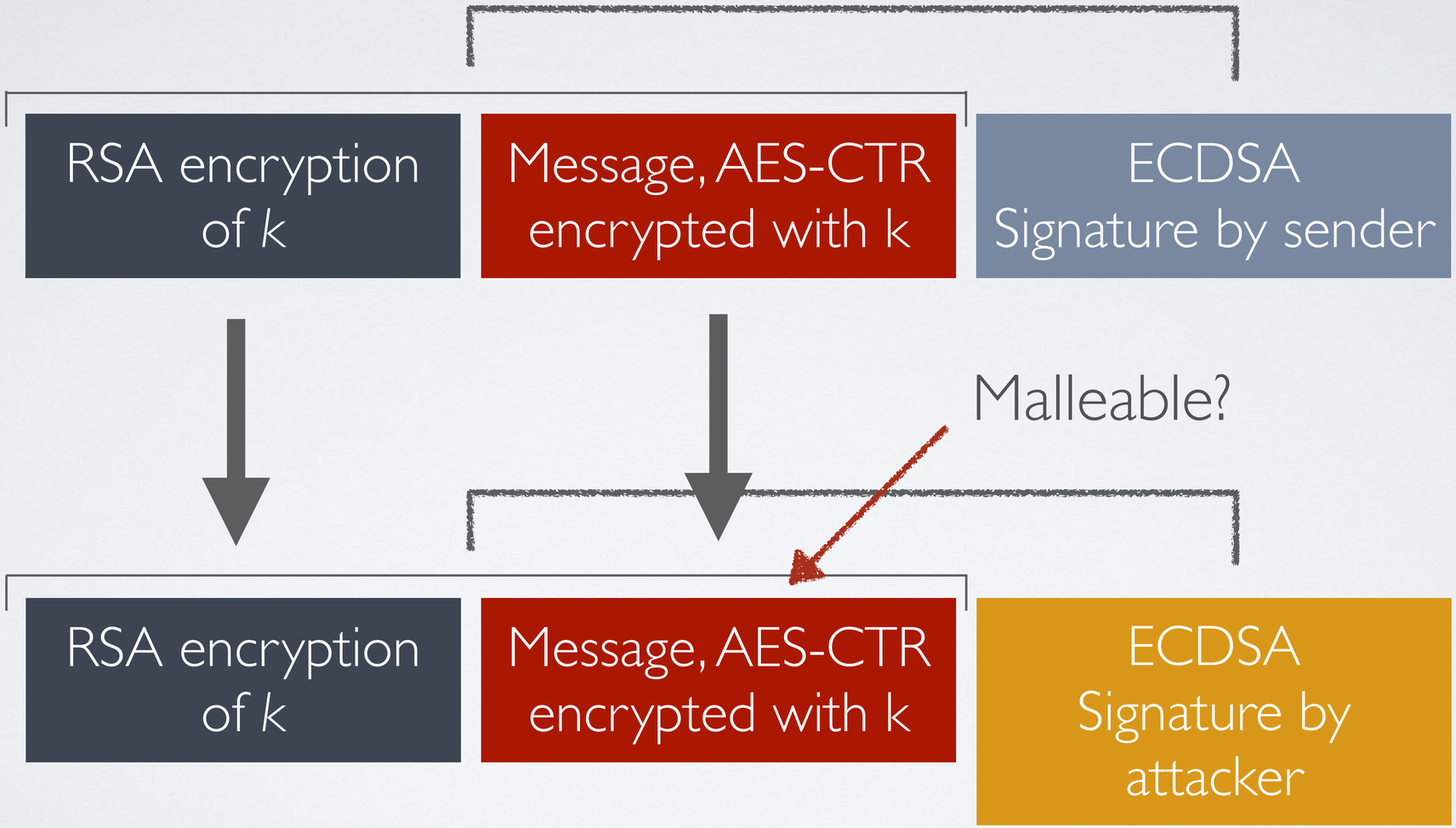
iMessage: Encryption



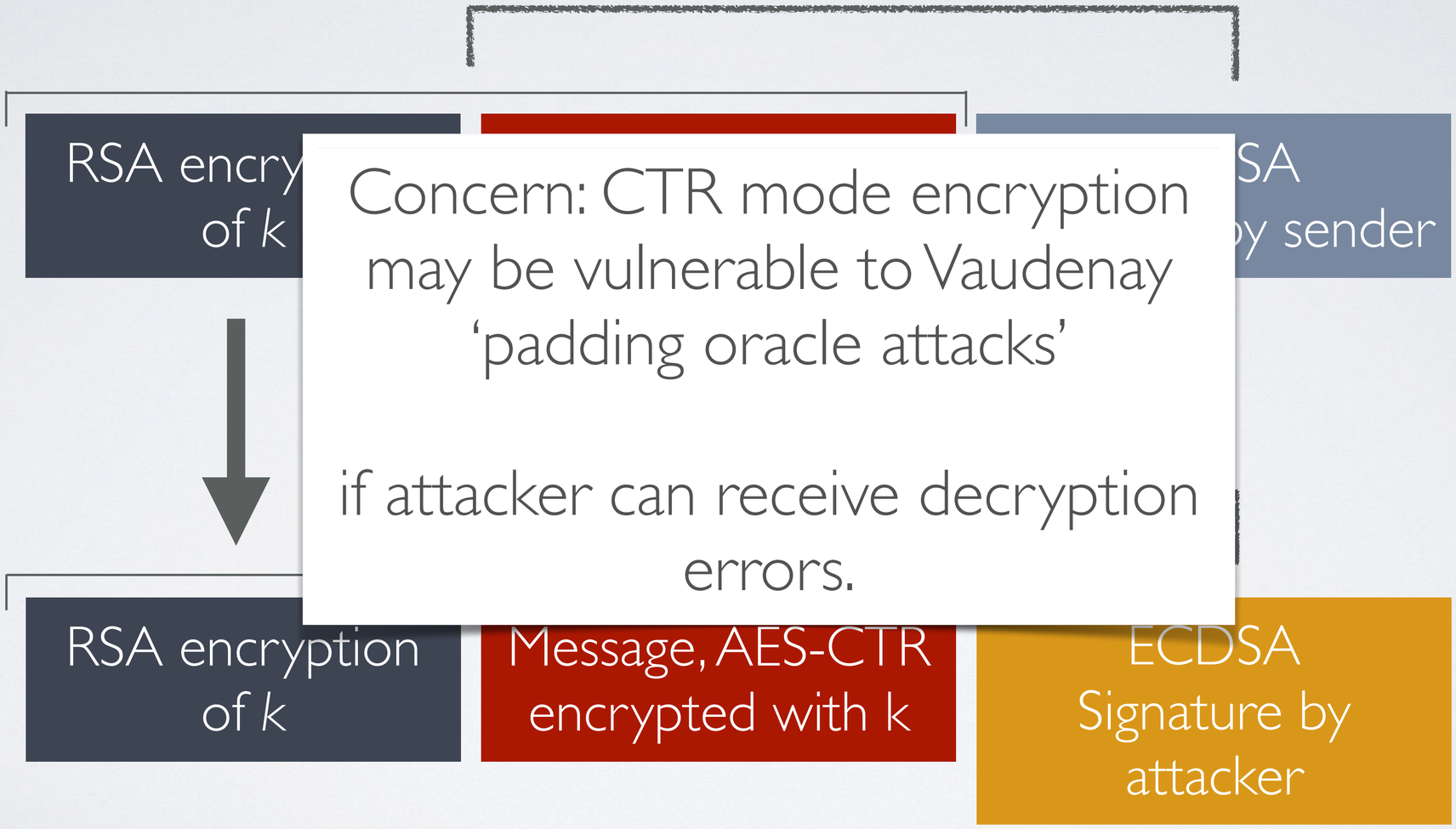
iMessage: Encryption



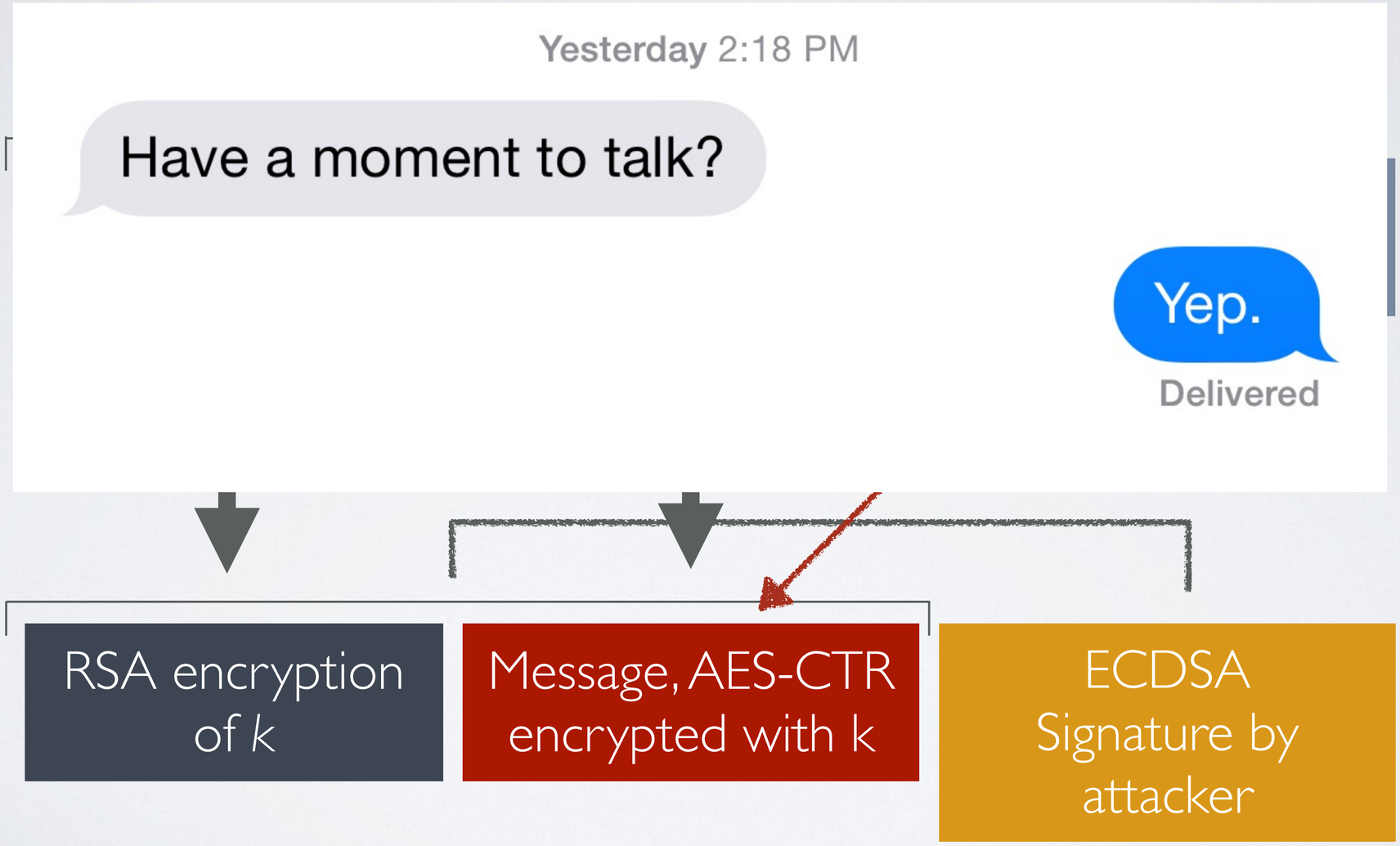
iMessage: Encryption



iMessage: Encryption



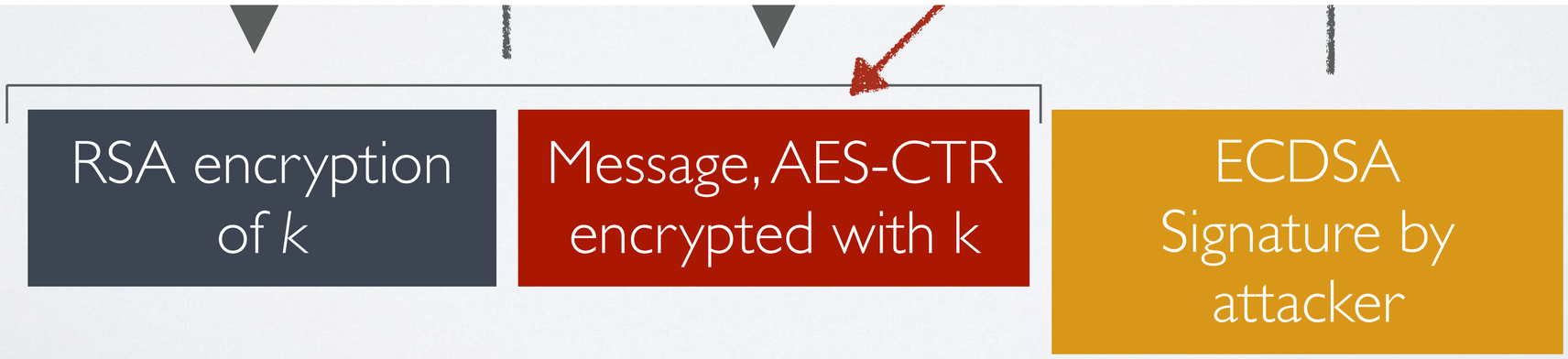
iMessage: Encryption



iMessage: Encryption

Here is an example of such a bplist:

```
D:   True
E:   'pair'
P:   <variable length binary data> (iMessage payload, deflate compressed)
U:   <128bit binary data> (iMessage UID)
c:   100
i:   <32bit integer> (messageId, same as in PUSH header)
sP:  mailto:tim_c@icloud.com  (sender URI)
t:   <256bit binary data> (sender Push-Token)
tP:  mailto:mark_z@facebook.com (receiver URI)
ua:  [Mac OS X,10.8.5,12F37,MacBookPro10,2] (sender OS and hardware version)
v:   1
```



Conclusion

- Cryptography is challenging!
  - We fail to push best practices down to the engineering community
  - They fail to pull best practices from the literature, even years after vulnerabilities are known
  - Cryptosystems continue to become more complex and vulnerable
  - This process is not really tolerable anymore