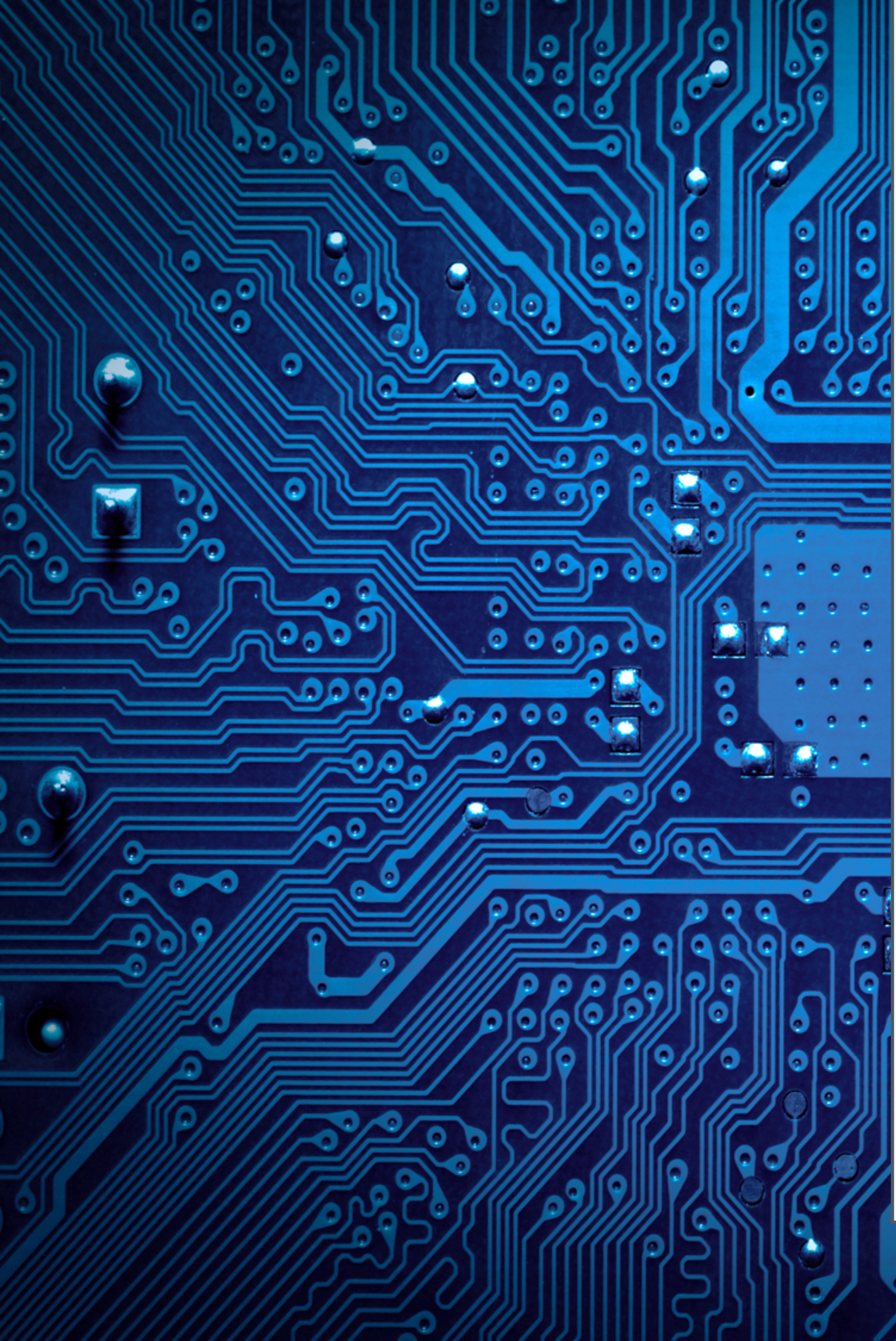


Midterm Review

ECE568 – Lecture 14
Courtney Gibson, P.Eng.
University of Toronto ECE

Block vs. Stream Ciphers

List two advantages of block ciphers over stream ciphers.

Key Space

The Solitaire cipher featured in Neal Stephenson's *Cryptonomicon* uses a shuffled deck of cards as the key. What is the key space for this cipher? Is it bigger or smaller than the key space for AES-128?

SSL / MACs

The SSL protocol uses a Message Authentication Code (MAC) to authenticate data being sent over SSL. List two advantages of using a MAC rather than a digital signature for authentication?

Vulnerabilities

```
#define BUFLEN 128
void get_file(char *dst, char *src, char len) {
  int pos;
  char n;

  for (n = 0; n < len; n++) {
    if (src[n] == ‘/’) pos = n;
  }
  for (n = pos; n < len; n++) dst[n] = src[n];
}
int main(int argc, char **argv){
  int len;
  char buffer[BUFLEN];
  char file[BUFLEN+4];

  /*strnlen(s, maxlen) returns the min of maxlen and strlen(s)*/
  len = strnlen(argv[1], BUFLEN);
  get_file(buffer, argv[1], len);
  sprintf(file, "%d%s", len, buffer);
  printf(file);
  return 0;
}
```

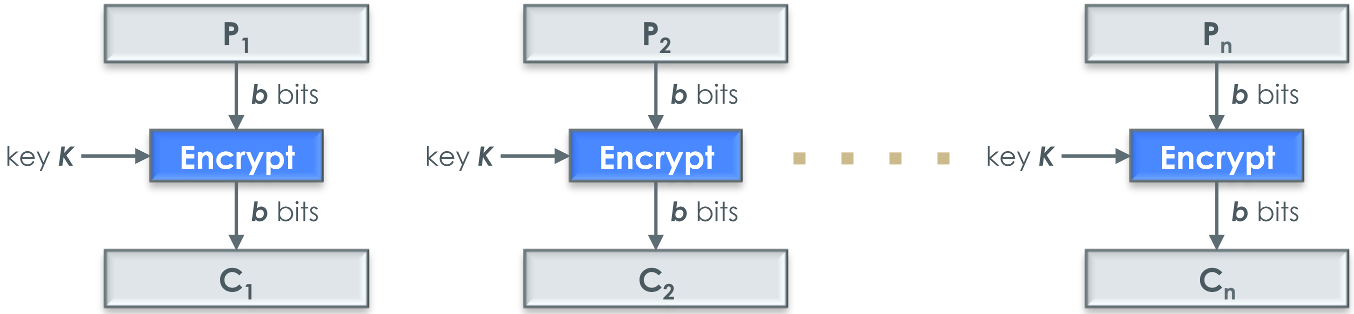
Encryption Modes

Mode	Encrypt	Decrypt
ECB	$C_i = E(K,M_i)$, $i = 1\ldots n$	$M_i = D(K, C_i)$, $i = 1\ldots n$
CBC	$C_1 = E(K, M_1 \oplus IV)$ $C_i = E(K, M_i \oplus C_{i-1})$, $i = 2\ldots n$	$M_1 = D(K, C_1) \oplus IV$ $M_i = D(K, C_i) \oplus C_{i-1}$, $i = 2\ldots n$
CFB		
OFB		

Electronic Codebook (ECB)

ECB is the simplest mode:

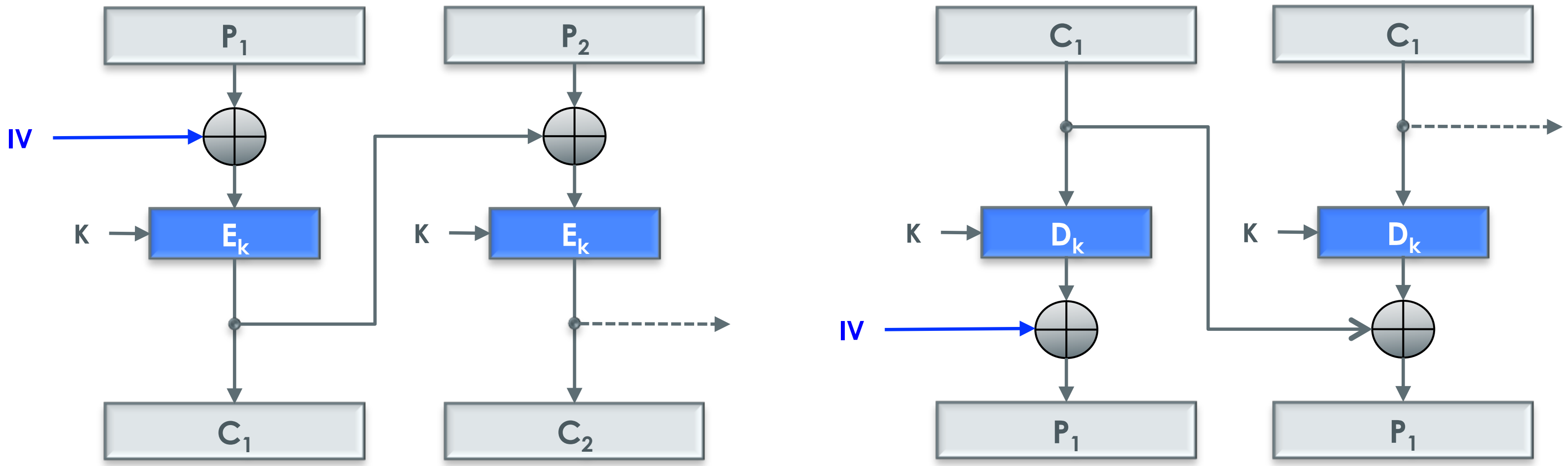
- The message is broken into block-sized chunks
- Padding is added to the last block
- Each chunk is encrypted independently



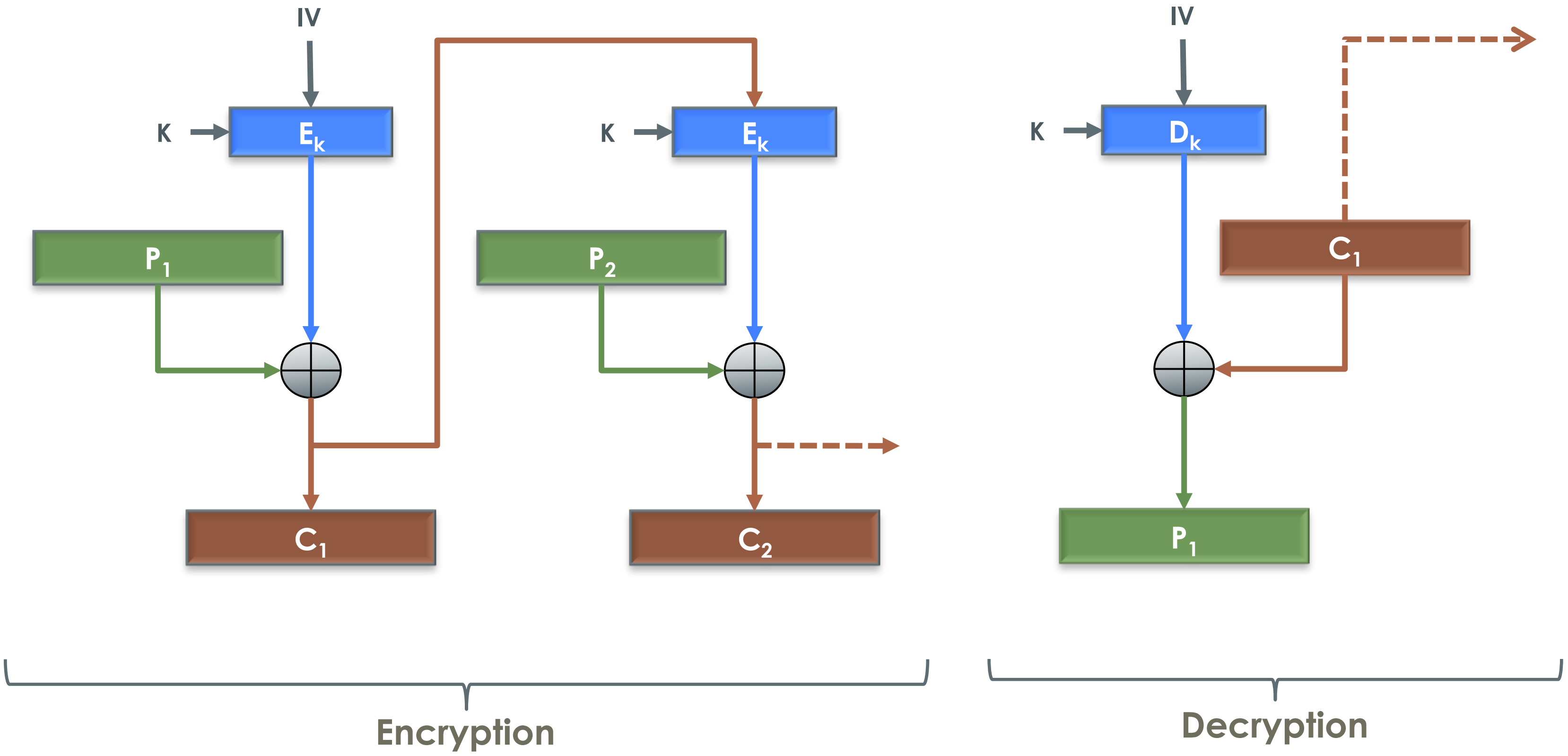
Cipher Block Chaining (CBC)

An alternate mode, CBC makes every blocks' input dependent on the cipher text (output) of the previous block

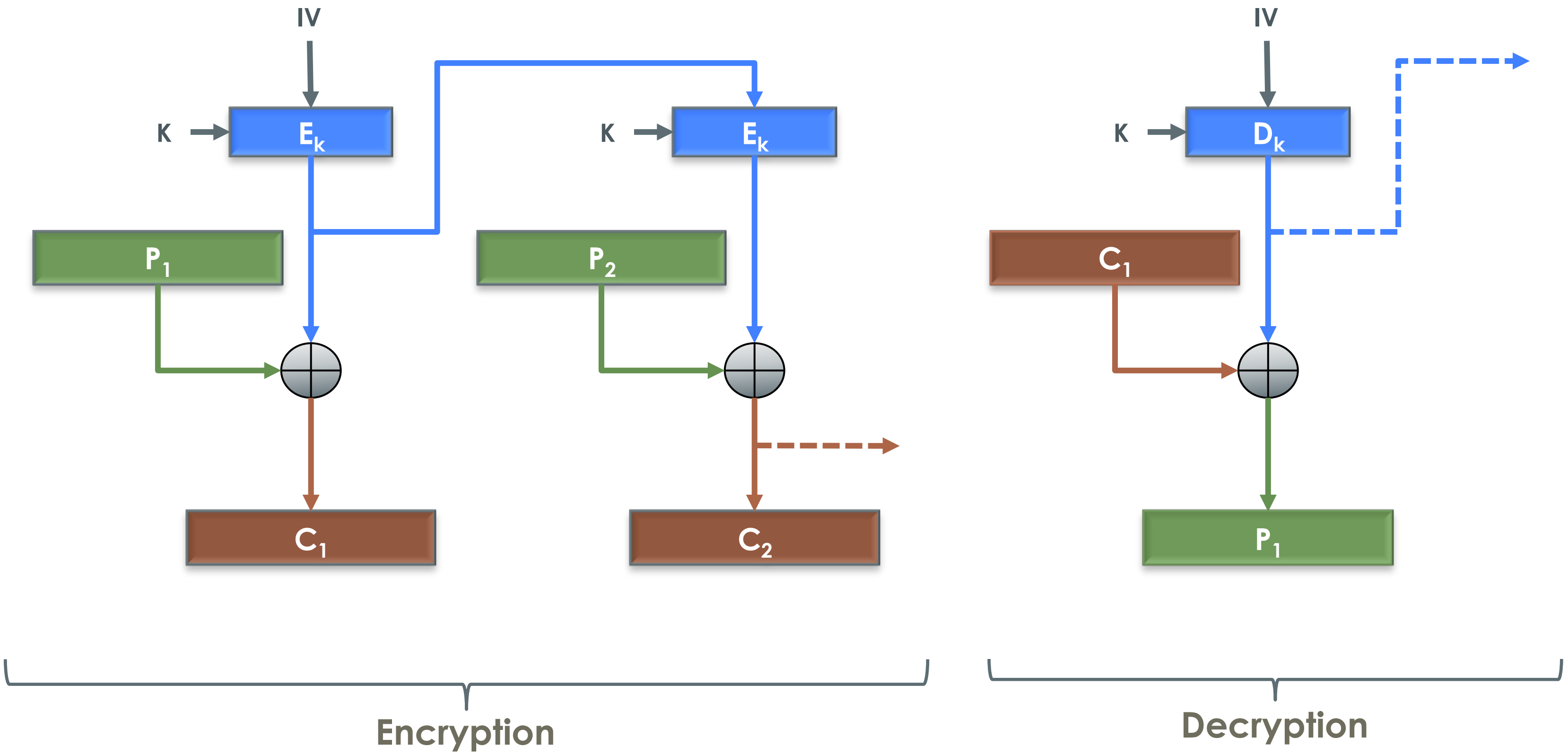
- Initial Value (IV) does not have to be secret, but shouldn't be reused for multiple messages



Cipher Feedback (CFB)



Output Feedback (OFB)



Encryption Modes

Mode	Encrypt	Decrypt
ECB	$C_i = E(K,M_i), i = 1\ldots n$	$M_i = D(K, C_i), i = 1\ldots n$
CBC	$C_1 = E(K, M_1 \oplus IV)$ $C_i = E(K, M_i \oplus C_{i-1}), i = 2\ldots n$	$M_1 = D(K, C_1) \oplus IV$ $M_i = D(K, C_i) \oplus C_{i-1}, i = 2\ldots n$
CFB	$C_0 = IV$ $C_i = M_i \oplus E(K, C_i-1), i = 1\ldots n$	$C_0 = IV$ $M_i = C_i \oplus D(K, C_i-1), i = 1\ldots n$
OFB	$O_0 = IV$ $O_i = E(K, O_{i-1}), i = 1\ldots n$ $C_i = M_i \oplus O_i, i = 1\ldots n$	$O_0 = IV$ $O_i = D(K, O_{i-1}), i = 1\ldots n$ $M_i = C_i \oplus O_i, i = 1\ldots n$

Needham-Schroeder Protocol

1. **A → T :** { A, B, N_A }
2. **T → A :** { N_A, K_{AB}, B, {K_{AB}, A}$_{KB}$ }$_{KA}$
3. **A → B :** { K_{AB}, A }$_{KB}$
4. **B → A :** { N_B }$_{KAB}$
5. **A → B :** { (N_B – 1) }$_{KAB}$

- What attack is possible if "B" in step 2 was omitted from the protocol?
- What attack is possible if "N_A" in step 2 was omitted from the protocol?
- What attack is possible if "(N_B – 1)" in step 5 was omitted from the protocol?

MACs and Signatures

Assume that Mallory is able to observe all messages sent between Alice and Bob. Mallory has no knowledge of any keys but the public keys used for digital signatures. State whether MAC and digital signatures protect against the four attacks described below. Given message X, the value of Auth(X) is computed using a MAC or a signature.

1) Alice sends a message X="Transfer $1000 to Mark" in the clear and Auth(X) to Bob. Mallory intercepts the message and replaces "Mark" with "Mallory". Will Bob detect it?

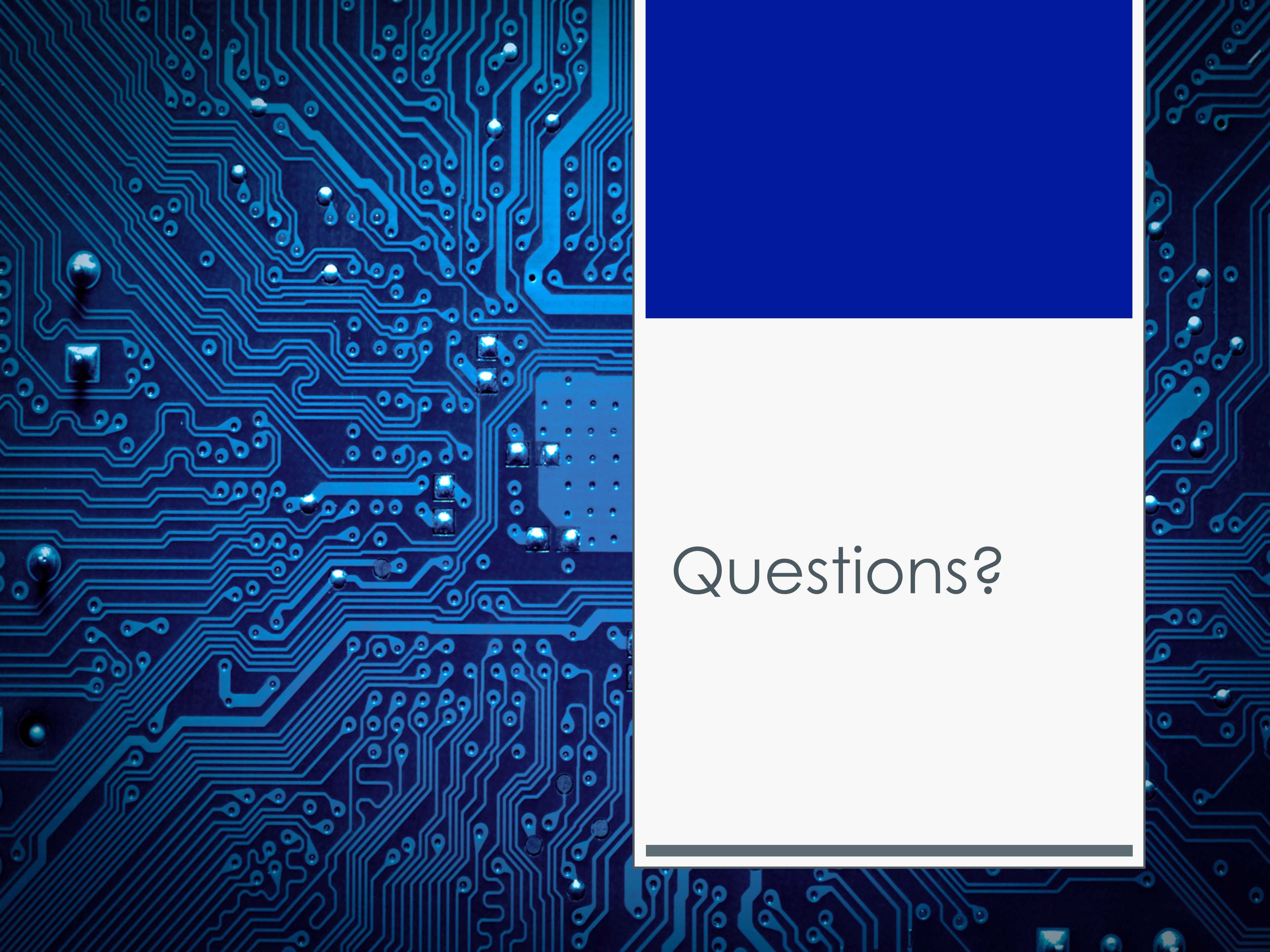
2) Alice sends a message X="Transfer $1000 to Mark" in the clear and Auth(X) to Bob. Mallory observes both messages and sends them to Bob a second time. Will Bob detect it?

3) Mallory and Alice both claim that they sent some message X with a valid Auth(X) to Bob. Can Bob know which of Alice or Mallory sent the message?

4) Bob claims that he received a message X with a valid Auth(X) from Alice (e.g., "Transfer $1000 from Alice to Bob") but Alice claims that she never sent it. Can Alice's claim be checked?

Buffer Overflow

Typical buffer overflow attacks use the fact that the return address is at a **higher** memory address than the local variables (*i.e.*, the stack grows towards an address of 0). Explain how to carry out a buffer overflow attack if the stack layout is reversed (*i.e.*, the return address is at a lower memory address than the local variables).



Questions?