Calculators may be used in this examination provided they are not capable of being used to store alphabetical information other than hexadecimal numbers

UNIVERSITY OF BIRMINGHAM

School of Computer Science

Security of Real-World Systems

Main Summer Examinations 2024

Time allowed: 2 hours

[Answer all questions]

Note

Answer ALL questions. Each question will be marked out of 20. The paper will be marked out of 60, which will be rescaled to a mark out of 100.

Question 1

Imagine that we have a protocol for a client (C) and server (S) to set up a secure key that runs as follows.

The client has a long term private key ltc and the server knows the corresponding public key g^{ltc}.

The server has a long term private key lts and the client knows the corresponding public key g^{lts}.

To establish a secure channel the client generates a new random value ec and the server generates a new random value es and they send the following values to each other:

$$C->S : g^{ltc}, g^{ec}$$
$$S->C : g^{lts}, g^{es}$$

The server then calculates the key as $KDF((g^{ltc})^{lts}|(g^{ec})^{es})$ and the client uses $KDF((g^{lts})^{ltc}|(g^{es})^{ec})$.

A Key Compromise Impersonation (KCI) attack is an attack in which the attacker learns one parties secret key and then uses it to run the protocol with that party, and tricks that party into believing the attacker is someone else.

(a) Describe a KCI attack against the protocol described above, include the attack written in Alice and Bob notation. **[7 marks]**

(b) Describe the smallest change you could make to the protocol to stop this attack. Write the new protocol in Alice and Bob notation and clearly explain why the attack is stopped. **[7 marks]**

(c) Are there any real world scenarios in which a KCI attack would be a threat? If so describe two scenarios. If not explain why not. **[6 marks]**

Question 2

For an x86-64 crypto library, there are two variants of AES-implementations: (i) using Intel's AES-NI instruction set extension (which purely operates on `xmm` registers), and (ii) a pure software implementation that includes the following implementation of the `xtime()` function:

```
uint8_t xtime(uint8_t x) {
   uint16_t v = ((uint16_t)x) << 1;
   if(v & 0x100 == 0x100) {
      v ^= 0x1B; // real XOR
   }
   else {
      v ^= 0x00; // dummy XOR
   }
   return v & 0xFF;
}
```

(a) Is one (or are both) of the implementation variants vulnerable to a timing attack? Justify your answer and clearly indicate where any timing leakage(s) might arise. **[8 marks]**

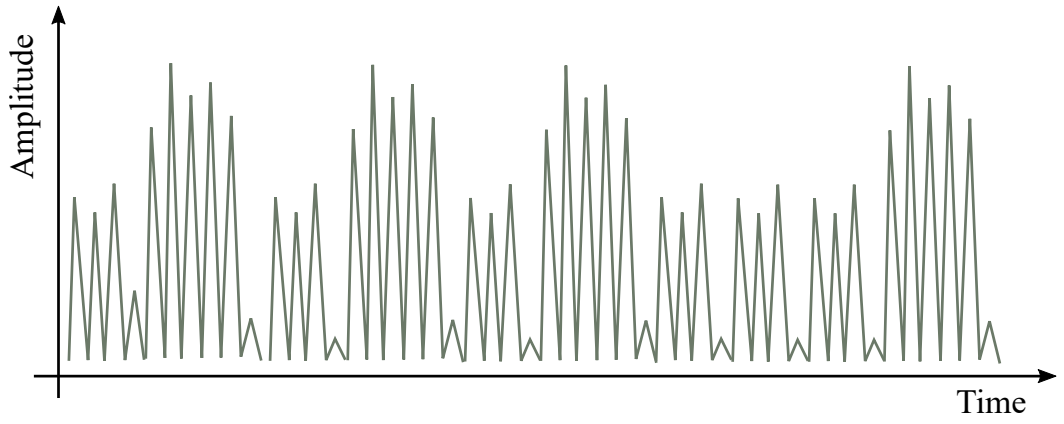
(b) Someone suggests to use a Rowhammer attack to inject a fault into the AES implementations and then use differential fault analysis to recover the key. For both implementation variants, indicate if this is possible or not. Justify your answers briefly. **[6 marks]**

(c) Assume an attacker can physically inject faults, resulting in bit flips in cache and registers, through voltage glitching of an Intel CPU. In that scenario, is one (or are both) of the implementation variants vulnerable to a differential fault attack? Justify your answer. **[4 marks]**

(d) Assume an attacker can record many power traces for implementation variant (i), and assume that the circuitry for AES-NI is not specially protected against power analysis. Is it then in principle possible to recover the key with differential power analysis? Justify your answer. **[2 marks]**

Question 3

An Elliptic Curve signature scheme is implemented in an ARM64 crypto library (as a Linux `.so` shared library) running under Linux on an embedded processor. The ARM64 architecture has a method to flush cachelines from userspace. The crypto library uses the left-to-right double-and-add algorithm explained in the lecture. With an oscilloscope, you observe the power trace during the scalar multiplication $Y = d \cdot X$, where d is the secret scalar:



The pseudocode for the scalar multiplication is as follows (this is *not* valid C code):

```
1 Y = identity_point;
2 for(int i = 256 - 1; i >= 0; i--) {
3     Y = double(Y);
4     if(bit i of d == 1)
5       Y = add(Y, X);
6 }
7 return Y;
```

(a) From the power trace, write down the sequence of executed double (D) and add (A) operations. Based on that, write down the respective bits of d. **[6 marks]**

(b) Describe one way how the power side-channel attack from a) can be mitigated (or at least made more difficult) through a software modification of the scalar multiplication. Give the new pseudocode. **[6 marks]**

(c) Assume an adversary has a user account on the processor, but cannot directly read-/write the data memory of the crypto library used by a process under a different user account. Would they be able to recover the full secret scalar with a cache attack? Justify your answer and describe the main steps of a potential attack in bullet points. **[8 marks]**

End of Paper

This page intentionally left blank.

Do not complete the attendance slip, fill in the front of the answer book or turn over the question paper until you are told to do so

Important Reminders

- Coats/outwear should be placed in the designated area.
- Unauthorised materials (e.g. notes or Tippex) must be placed in the designated area.
- Check that you do not have any unauthorised materials with you (e.g. in your pockets, pencil case).
- Mobile phones and smart watches **must** be switched off and placed in the designated area or under your desk. They must not be left on your person or in your pockets.
- You are not permitted to use a mobile phone as a clock. If you have difficulty seeing a clock, please alert an Invigilator.
- You are not permitted to have writing on your hand, arm or other body part.
- Check that you do not have writing on your hand, arm or other body part – if you do, you must inform an Invigilator immediately
- Alert an Invigilator immediately if you find any unauthorised item upon you during the examination.

Any students found with non-permitted items upon their person during the examination, or who fail to comply with Examination rules may be subject to Student Conduct procedures.