

i Cover page

Institutt for Datateknologi og informatikk

Eksamensoppgave i IDATT2503 Programvaresikkerhet og kryptografi

Eksamensdato: 05.12.2023

Eksamenstid (fra-til): 15:00 – 19:00

Hjelpemiddelkode/Tillatte hjelpemidler: D

Det er tillatt å bruke kalkulator Casio FX- 82ES, Casio fx-570ES PLUS

Faglig kontakt under eksamen:

Donn Morrison 45548895

Ole Christian Eidheim 905 51 635

Dag Olav Kjellemo 47681639

Faglig kontakt møter i eksamenslokalet: Nei

ANNEN INFORMASJON:

Skaff deg overblikk over oppgavesettet før du begynner på besvarelsen din.

Les oppgavene nøye, gjør dine egne antagelser og presiser i besvarelsen hvilke forutsetninger du har lagt til grunn i tolkning/avgrensing av oppgaven. Faglig kontaktperson kontaktes kun dersom det er direkte feil eller mangler i oppgavesettet. Henvend deg til en eksamensvakt hvis du mistenker feil og mangler. Noter spørsmålet ditt på forhånd.

Ingen håndtegnings: Denne eksamenen tillater ikke bruk av håndtegnings. Har du likevel fått utdelt skanne-ark, er dette en feil. Arkene vil ikke bli akseptert for innlevering, og de vil derfor heller ikke sendes til sensur.

Varslinger: Hvis det oppstår behov for å gi beskjeder til kandidatene underveis i eksamen (f.eks. ved feil i oppgavesettet), vil dette bli gjort via varslinger i Inspira. Et varsel vil dukke opp som en dialogboks på skjermen. Du kan finne igjen varselet ved å klikke på bjella øverst til høyre.

Trekk fra/avbrutt eksamen: Blir du syk under eksamen, eller av andre grunner ønsker å levere blankt/avbryte eksamen, gå til "hamburgermenyen" i øvre høyre hjørne og velg «Lever blankt». Dette kan ikke angres selv om prøven fremdeles er åpen.

Tilgang til besvarelse: Etter eksamen finner du besvarelsen din i arkivet i Inspira. Merk at det kan ta én virkedag før eventuelle håndtegnings vil være tilgjengelige i arkivet.

1 302

Når en HTTP vert svarer med "HTTP/1.1 302 Found", det trengs hvilken HTTP header ved responsen?

Velg ett alternativ:

- ☐ Content-Type
- ☐ Set-Cookie
- ☐ Referer
- ☐ Location



Maks poeng: 2

2 shellcode

Et effektivt grep mot shellcode er:

Velg ett alternativ:

- ☐ shadow stack
- ☐ NX stack
- ☐ heap allocation
- ☐ setuid-root binaries



Maks poeng: 2

3 translate

Gjennom et sikkerhetstest du finner en endpoint som svarer med:

sh: 1: Syntax error: Unterminated quoted string

Etter å ha sendt følgende fuzzing payload:

http://translation.no/translate/nb/nn/hei"

Hva er den mest sannsynlig feil som du har nettopp oppdaget?

Velg ett alternativ:

- ☐ SQL injection
- ☐ Command injection
- ☐ Insecure deserialisation
- ☐ Server-side template injection



Maks poeng: 2

4 stack

Hvilken av de følgende elementer kan ikke overskrevet når det skjer en stack buffer overflow?

Velg ett alternativ:

- ☐ lagret RBP
- ☐ andre stack variabler
- ☐ return address
- ☐ RAX/EAX



Maks poeng: 2

5 Exploits

Sett den beste mulig input til riktig attack vector for å finne svakheten:

 [Hjelp](#)

`http://0.0.0.0:8081`

`' or sleep(10); --`

`|| id`

`">`

`{{ 2+2 }}`

`%p`

SQL injection `' or sleep(10); --` ✓

Command injection `|| id` ✓

Server-side template injection `{{ 2+2 }}` ✓

Cross-site script injection `">` ✓

Server-side request forgery `http://0.0.0.0:80` ✓

String format vulnerability `%p` ✓

Maks poeng: 12

6 Spot the vuln

```
# Whitelisted URL
whitelist_pattern = re.compile(r'^https?://safe.ntnu.no/.*$')

@app.route('/fetch_resource', methods=['GET'])
def fetch_resource():
    # Get the URL parameter from the request
    url = request.args.get('url')

    if not url:
        return jsonify({'error': 'Missing URL parameter'}), 400

    # Check if the provided URL is whitelisted
    if not whitelist_pattern.match(url):
        return jsonify({'error': 'URL is not whitelisted'}), 403

    try:
        # Make an HTTP GET request to the specified URL
        response = requests.get(url)
        response.raise_for_status() # Raise an exception for HTTP errors

        # Return the content of the response
        return jsonify({'content': response.text})

    except requests.exceptions.RequestException as e:
        return jsonify({'error': f'Request to {url} failed: {str(e)}'}), 500

if __name__ == '__main__':
    app.run(debug=True)
```

Gitt kildekoden overst, beskriv (1) feilen og (2) hvordan den kan utnyttes og (3) innvirkning (impact).

Skriv ditt svar her

Format
|
B
I
U
x₂
x²
I_x
|
📄
📋
|
↶
↷
↺
|
≡
≡
|
Ω
📱
|
✎
|
Σ
|

✖

Words: 0

Maks poeng: 10

The dots "." in the regex whitelist are not escaped, and will match any character. Thus, the whitelist filter can be bypassed by an attacker who controls the DNS entry for:

safexntnu.no

safe-ntnu.no

etc.

If the attacker controls which IP address these names resolve to, the attacker can resolve these to local/private addresses like 127.0.0.1 or 10.0.0.1 or 0.0.0.0, yielding a server-side request forgery vulnerability, where the attacker could then access unprotected services meant to be private and conduct internal port scans.

The full impact depends on the services accessed, but management consoles, services with outstanding CVEs, etc could be exposed, with the worst case leading to remote code execution.

7 Assembly

Gitt følgende Assembly kode:

```
section .data
    msg: db "Hello", 10

section .text


global _start

_start:
    mov rcx, 1
    dec rcx
    jnz exit
    mov rax, 1      ; The system call for write
    mov rdi, 1      ; File descriptor 1 - standard output
    mov rsi, msg
    mov rdx, 2
    syscall

exit:
    mov rax, 60     ; The system call for exit (sys_exit)
    mov rdi, 10
    syscall
```

a) Hva blir utskriften når programmet kjøres:

Velg ett alternativ:

- ☐ "He" 
- ☐ "" (no output)
- ☐ "Hello\n" (\n is newline)
- ☐ "Hello"

b) Hva blir `exit` status når programmet avslutter?

Velg ett alternativ☐ 16☐ 0☐ 10☐ 60☐ 1

Maks poeng: 8**8 Fuzzing**

Du fuzzer følgende funksjon med undefined sanitizer aktivert:

```
int average_byte(const char *array, size_t length) {  
    int sum = 0;  
    for (size_t i = 0; i < length; ++i)  
        sum += array[i];  
    return sum / length;  
}
```

Hvilke feil kan en da finne?

Velg ett eller flere alternativer☐ Stack overflow☐ Integer overflow☐ Buffer overflow☐ Divide by zero

Maks poeng: 6

9 OpenSSL

Hva inneholder OpenSSL?

Velg ett eller flere alternativer

- | | |
|-------------------------------------------------------|---|
| <input type="checkbox"/> Contracts | |
| <input type="checkbox"/> Symmetric encryption | ✓ |
| <input type="checkbox"/> Cryptographic protocols | ✓ |
| <input type="checkbox"/> Two-factor authentication | |
| <input type="checkbox"/> Fuzzers | |
| <input type="checkbox"/> Asymmetric encryption | ✓ |
| <input type="checkbox"/> Sanitizers | |
| <input type="checkbox"/> Cryptographic hash functions | ✓ |

Maks poeng: 4

10 System programming languages

Hvilke programmeringsspråk regnes som systemprogrammeringsspråk?








Velg ett eller flere alternativer


- | | |
|-------------------------------------|---|
| <input type="checkbox"/> Rust | ✓ |
| <input type="checkbox"/> JavaScript | |
| <input type="checkbox"/> Python | |
| <input type="checkbox"/> C | ✓ |
| <input type="checkbox"/> Java | |

Maks poeng: 2

11 Cryptography 1

Skriv ditt svar her

Format ▾ | **B** *I* U x_2 x^2 | I_x |   |    | $\frac{1}{2}$ $\frac{3}{4}$ | Ω  |  |

Σ | 

See following attachments for Cryptography 1-4.










Words: 0


Maks poeng: 16

12 Cryptography 2

Skriv ditt svar her

Format ▾

B *I* U x_2 x^2 I_x         

Σ 

Words: 0

Maks poeng: 18

13 Cryptograhyy 3

Identifiser og bekriv en hybrid krypteringsprotokoll i en klient-tjener setting

1. Identifikasjon:

Navngi et eksempel på en protokoll som brukes på internett i en klient-tjener setting som benytter hybrid kryptering, det vil si en kombinasjon av symmetrisk og asymmetrisk kryptering.

2. Forklaring av Bruk:

- Forklar kort hvordan protokollen utnytter både symmetrisk og asymmetrisk kryptering for å oppnå spesifikke sikkerhetsmål.
- Diskuter hvorfor hver type kryptering er valgt for sitt respektive formål i konteksten av protokollen.

3. Merk:

- Gi en kort oversikt i stedet for en detaljert beskrivelse av protokollen.
- Hvis du ikke kan huske en spesifikk protokoll, gi en generell forklaring på hvordan hybrid kryptering kan brukes til å tilrettelegge for sikker og effektiv kommunikasjon i en klient-tjener setting.

Skriv ditt svar her

Format

B

I

U

x_2

x^2

I_x

Σ

Words: 0

Maks poeng: 8

14 Cryptography 4

- a) Hva er en MAC (Message Authentication Code) og hva brukes MACer til?
- b) Hva skiller en MAC fra en kryptografisk hash funksjon?
- c) For at en MAC skal regnes som sikker, hva slags angrep skal den være motstandsdyktig mot?
- d) Hva er det som svarer til en MAC i asymmetrisk kryptografi?

Skriv ditt svar her

Format ▾ | **B** *I* U x_e x^e | I_x | | | | Ω |

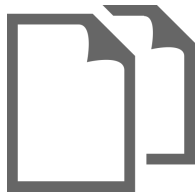
Σ |

Words: 0

Maks poeng: 8

Question 11

Attached



I denne oppgaven bruker vi det engelske alfabetet, a-z, for både klartekst og chifftertekst, men skriver chifftertekstene med store bokstaver. Mellomrom og andre tegn fjernes i krypteringen.

a) Krypter meldingen "no way" ved hjelp av et skift/rotasjonschiffer med nøkkel **y**.

b) Krypter meldingen "no way" ved hjelp av Vigenère-chifferet med nøkkel **aye**.

c) Dekrypter chiffterteksten "GMSD" med samme nøkkel **aye**.

Vi koder bokstavene som tall modulo $n = 26$, der 0 tilsvarer "a", osv.

d) Skriv en matematisk formel for skiftchifferet med nøkkel **y**, gitt denne kodingen.

Alice og Bob bestemmer seg for å bruke skiftchifferet i CBC-modus, med nøkkel **y**. Vi erstatter den vanlige "XOR"-operasjonen definert ved vanlig addisjon modulo 26.

Eksempler: $B \oplus C = D$, og $K \oplus R = C$

e) Krypter "yes" i CBC-modus ved hjelp av nøkkelen **y** og initialvektor $IV = \mathbf{b}$

f) Eva har avlyttet en lang chifftertekst hun vet er kryptert med et skiftchiffer. Hun merker seg at **T** er den mest vanlige bokstaven, men vet ikke noe mer om fordelingen av tegn i chiffterteksten. Anta at den opprinnelige meldingen var i vanlig hverdagslig engelsk. Senere ser hun meldingen "BHV" kryptert med samme nøkkel.

Kan hun dekryptere den? Forklar.

Question 12

Attached



Alice har $n = 943$ og $e = 9$ som offentlig nøkkel for sin RSA-baserte kryptering.

a) Finn Alices private nøkkel fra likningen $4 \cdot 880 - 391 \cdot 9 = 1$. Litt informasjon: $943 = 23 \cdot 41$.

b) Hvorfor kan ikke $e = 5$ brukes som del av offentlig nøkkel?

c) Krypter $m = 3$ med Alices offentlige nøkkel. Vis hvordan du gjør dette effektivt med få multiplikasjoner.

d) Mallory får tak i den krypterte meldingen C og endrer den til C' . Alice mottar C' og dekrypterer den til meldingen $2m$, mens dekrypteringen av opprinnelig melding er m . Hvordan konstruert Mallory C' fra C , uten å vite m ?

Alice signerer meldingen 10 med sin private nøkkel, beregner $10^d \equiv 939 \pmod{n}$, og sender $(10, 939)$ til Bob.

e) Hvordan kan Eva skape en eksistensiell forfalskning når Alice signerer på denne måten?

f) Hvordan kan Alice gjøre for å unngå denne typen angrep? (Igjen, vi ignorerer at nøklene er usikre fordi de er veldig små.)

Cryptography 1

In this question, we use the English alphabet, with letters a-z, for both plain texts and cipher texts, but write cipher texts in capitals. Also, spaces and punctuation is left out in the encryption.

- a) Encrypt the message "no way" using a shift/rotation cipher with key **y**.
- b) Encrypt the message "no way" using the Vigenere chipher with key **aye**
- c) Decrypt the ciphertext "GMSD" with the same key **aye**.
- d) We encode the letters as numbers modulo $n = 26$, with 0 corresponding to "a", etc. Write a mathematical formula for the shift cipher with key **y** using the above encoding.
- e) Alice and Bob decide to use CBC-mode of operation with the shift cipher with their key **y**. The "XOR" operation is replaced by addition modulo 26.
For example, $B \oplus C = D$, and $K \oplus R = B$ (incorrectly given as C in exam)
Encrypt "yes" in CBC-mode using the key **y** and initial vector $IV = \mathbf{b}$
- f) Eve has eavesdropped a long ciphertext she knows is encrypted with a shift cipher. She notices that **T** is the most common letter, but does get any other statistics. Assume that the plain text message was normal English. Later she sees the message "BHV" encrypted with the same key. Can she decrypt it? Explain.

Answer

- a) The letter **y** means shifting the alphabet by 24 letters, giving the substitutions

plain:	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
cipher:	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X

This give the cipher text **LMUYW**

- b) The first, fourth, seventh etc letter is encrypted using key **a**, the second, fifth, eighth etc letter is encrypted using **y**, and the rest using **e**. This gives **NMAAW**
- c) **good**
- d) $e(x) = x + 24 \mod 26$, $d(x) = x - 24 \mod 26 = x + 2 \mod 26$

- e) Letting x_k be the k -th letter of plain text, and y_k the k -th letter of cipher text, with $c_0 = \mathbf{b} = 1$, corresponding to the letter \mathbf{b} , we get

$$y_1 = e(x_1) = (x_1 + 1) + 24 \mod 26 = 24 + 1 + 24 \mod 26 = 23 = \mathbf{X} \quad (1)$$

$$y_2 = x_2 + y_1 + 24 \mod 26 = 4 + 23 + 24 \mod 26 = 25 = \mathbf{Z} \quad (2)$$

$$y_3 = x_3 + y_2 + 24 \mod 26 = 18 + 25 + 24 \mod 26 = 15 = \mathbf{P} \quad (3)$$

- f) Yes, she can, if we assume the encryption was in ECB-mode. The key can is determined by knowing just one pair of plain text and cipher text, consisting of only a single letter. We can assume that \mathbf{e} is the most common plain text letter, and that it corresponds to the cipher text \mathbf{T} , giving a shift of 15 letters.

Decrypting \mathbf{BHV} then gives the plaintext \mathbf{msg} .

If another mode like CBC is used, she cannot infer that the key is as above.

Cryptography 2

Alice has $n = 943$ and $e = 9$ as the public key for her RSA based encryption.

- Find Alice's private key d from the equation $4 \cdot 880 - 391 \cdot 9 = 1$. For your information, $943 = 23 \cdot 41$. The key should be a positive integer.
- Why cannot $e = 5$ be used as part of the public key?
- Encrypt $m = 3$ with Alice's public key. Show how to do this efficiently with few multiplications.
- Mallory intercepts an encrypted message C , and changes it to C' . Alice receives C' and decrypts it to $2m$, while the untampered C decrypts to m . How did Mallory construct C' from C , without knowing m ?
- Alice signs the message 10 with her private key, calculating $10^d \equiv 939 \mod n$, and sends $(10, 939)$ to Bob. How can Eve create an existential forgery when Alice signs in this way?
- What can Alice do to avoid this type of attack (again, we ignore that the keys are insecure by being very small)?

Answer

- 23 and 41 are prime numbers, so these are part of the private key data. From this we can calculate $\phi(943) = (23 - 1)(41 - 1) = 880$. The private key d is the multiplicative inverse of 9 modulo 880. This we get from the given diophantine equation:

$$4 \cdot 880 - 391 \cdot 9 = 1$$

$$-391 \cdot 9 \equiv 1 \mod 880$$

$$9^{-1} \equiv -391 \equiv 880 - 391 \equiv 489 \mod 880$$

- b) 5 is not relatively prime to $\phi(943) = 880$, so it does not have a multiplicative inverse.
- c) Using repeated squaring, all modulo calculations done modulo 943:

$$3^1 = 3$$

$$3^2 = 9$$

$$3^4 = 9^2 = 81$$

$$3^8 = 81^2 = 6561 \equiv 903$$

$$3^9 = 3 \cdot 3^8 \equiv 2709 \equiv 823$$

- d) Encryption and decryption in RSA is *multiplicative*, so Mallory needs to encrypt 2 with Alice's public key, $e(2) = 512$, and multiply this to C to get C' .

$$d(C') = d(e(2) \cdot C) = 2 \cdot d(C) = 2 \cdot m$$

- e) An existential forgery is any valid message-signature pair that Eve has not seen before. One way is to swap message and signature: She chooses any s for signature, and calculates the "message" s^e . (s^e, s) is a valid signature pair.

Another way is from a valid pair (m, s) create new pairs (m^k, s^k) for integers k : if s is a valid signature for m , then $m^e \equiv s$, but then $(m^k)^e \equiv (m^e)^k \equiv s^k$.

- f) Alice can first use a cryptographic hash function to compute a hash for the message. She then signs the hash.

Cryptography 3

Identify and Briefly Explain a Hybrid Encryption Protocol in a Client-Server Setting

1. **Identification:** Name an example of a protocol used on the internet in a client-server setting that employs hybrid encryption, i.e., a combination of symmetric and asymmetric encryption.
2. **Explanation of Usage:** Briefly explain how the protocol utilizes both symmetric and asymmetric encryption to achieve specific security objectives. Discuss why each type of encryption is chosen for its respective purpose in the context of the protocol.
3. **Note:** Provide a concise overview rather than a detailed description of the protocol. If you cannot recall a specific protocol, give a general explanation of how hybrid encryption can be used to facilitate secure and efficient communication in a client-server setting.

Solution:

1. TLS is a widely used protocol, used for example in HTTPS
2. Asymmetric encryption is used in the initial handshake phase to exchange (session) keys. Server will usually send the client a certificate, containing its public key, which *authenticates* the server. Asymmetric encryption is used so that any client can connect to server.
3. Data transmission uses symmetric encryption using the shared keys established in the handshake. Symmetric encryption is used since its much more efficient than asymmetric. MACs are also computed for each chunk of data. This ensures confidentiality, integrity and authentication.

Cryptography 4

- a) What is a MAC (Message Authentication Code) and what is it used for?
- b) What differentiates a MAC from a cryptographic hash function?
- c) For a MAC to be secure, what type of attack does it need to be resistant against?
- d) What corresponds to a MAC in asymmetric cryptography?

Solution

- a) A MAC is a (short) piece of information used to authenticate (verify sender) and ensure integrity (not altered) of a message, and sent with the message. It is calculated from the message and a secret key. A receiver must have the same key, and can recompute the MAC, and can verify integrity and authenticity by comparing it with the one sent with the message.
- b) A cryptographic hash function does not use a key, and provides only integrity of a message.
- c) It needs to be resistant to existential forgeries under chosen or known message attacks. I.e. an adversary should not be able to create a MAC for messages, even if they have other MACs for known or chosen messages.
- d) Digital signatures. One also obtains non-repudiation, in addition to authenticity and integrity.