

Exam Briefing

- CA would be uploaded to Canvas. There would be a deadline (to be decided later) to settle discrepancy. Wait for canvas announcement.
- Exam:
 - 24 MCQs. 3 short questions.
 - More weightage on topics 4 onward (topics not covered in midterm).

Exam:

Topics *excluded*:

- Detailed mechanism of padding oracle.
- SetUID and Unix file permission (some past-year papers had that).
- Firewall rules that appeared in some past-year papers.

Keywords that *might appear*:

- Hash, Encryption, password file, mac, signature, PKI, Forward secrecy
- DH key-exchange, authenticated key-exchange
- DNS, ARP, Re-negotiation attack
- DMZ, Intrusion detection system.
- Alice, Bob, Café owner, VPN
- SQL injection, “UPDATE” statement in SQL.
- Cookies, XSS, CSRF, same-origin policy
- Compartmentalization, Role-based access control, Principle of Least Privilege, Defense in Depth, Swiss Cheese model, Controlled invocation, elevated privilege, privilege escalation.
- Biba vs Bell-LaPaula

Endterm Test

2025/2026 Semester 1

27 November 2025

Time allowed: 2 hour

Instructions:

1. This is a **CLOSED** book assessment. You are allowed to bring **TWO** double-sided A4 sheets of notes for this assessment.
2. The assessment paper contains **TWENTY SEVEN (27) questions** and comprises **SIXTEEN (16) pages** including this cover page.
3. The time allowed is **2 hour**.
4. Use of calculators is allowed.

Instructions for Answer Sheet:

1. The answer sheet comprises 2 pages.
2. Write your student number on the answer sheet and shade the corresponding circles using a dark pen or pencil. Ensure that the bubbles are completely filled, and do not use ticks or crosses to mark your answers.
3. The answer sheet consists of **circular bubbles corresponding to the multiple choice questions**. Please ensure to shade in the circles corresponding to the correct answers. **Tick marks will not be accepted as valid answers**.
4. Do not write your name.
5. You must submit only the answer sheet and no other documents. The question set may be used as scratch paper.
6. You are allowed to use pencils, ball-pens or fountain pens. No red color.
7. Marks may be deducted for unrecognizable handwriting. All corrections should be cleanly erased or covered with correction fluid or tape.
8. All answers must be written within the corresponding box provided. Anything written outside the answer box will not be accepted.

References

1. Guideline on key strength in this assessment:
 - (a) Exhaustive search up till 100-bit keys is feasible. That is, 2^{100} number of cryptographic operations is feasible.
 - (b) Exhaustive search on 128-bit keys is infeasible.
 - (c) To secure against online dictionary attacks, password entropy of 49 bits is sufficient.
 - (d) To secure against offline dictionary attacks, password entropy of 128 bits is sufficient.
2. Output of SHA3() is 256 bits.
3. Description of `strncpy`.

NAME
`stpcpy, stpncpy, strcpy, strncpy -- copy strings`

LIBRARY
Standard C Library (libc, -lc)

SYNOPSIS
`#include <string.h>`

```
char *
stpcpy(char *dst, const char *src);
char *
stpncpy(char *dst, const char *src, size_t len);
char *
strcpy(char *dst, const char *src);
char *
strncpy(char *dst, const char *src, size_t len);
```

DESCRIPTION
The `stpcpy()` and `strcpy()` functions copy the string `src` to `dst` (including the terminating `\'0' character.)

The `stpncpy()` and `strncpy()` functions copy at most `len` characters from `src` into `dst`. If `src` is less than `len` characters long, the remainder of `dst` is filled with `\'0' characters. Otherwise, `dst` is not terminated.

The source and destination strings should not overlap, as the behavior is undefined.

Multiple Choice Questions [72 marks]

Each question is worth 3 marks. Each question has only one correct answer. No marks will be awarded if more than one answer is selected. Some questions may be awarded partial marks for answers that are close but incorrect. Ensure that the corresponding bubbles are shaded properly in the answer sheet.

- 2 answer pages in one sheet. (2nd page not shown here).
 - The hardcopy not in high resolution, like our Midterm. To confirm the correct bubbles, count them.
 - Completely cover the bubbles. No tick.



- Clearly choose only one single choice per question.
 - Suggestion: Use 2B pencil and bring good eraser.

1. ○ A ○ B ○ C ○ D ○
2. ○ A ○ B ○ C ○ D ○
3. ○ A ○ B ○ C ○ D ○
4. ○ A ○ B ○ C ○ D ○
5. ○ A ○ B ○ C ○ D ○
6. ○ A ○ B ○ C ○ D ○
7. ○ A ○ B ○ C ○ D ○
8. ○ A ○ B ○ C ○ D ○
9. ○ A ○ B ○ C ○ D ○
10. ○ A ○ B ○ C ○ D ○
11. ○ A ○ B ○ C ○ D ○
12. ○ A ○ B ○ C ○ D ○
13. ○ A ○ B ○ C ○ D ○
14. ○ A ○ B ○ C ○ D ○
15. ○ A ○ B ○ C ○ D ○
16. ○ A ○ B ○ C ○ D ○
17. ○ A ○ B ○ C ○ D ○
18. ○ A ○ B ○ C ○ D ○
19. ○ A ○ B ○ C ○ D ○
20. ○ A ○ B ○ C ○ D ○
21. ○ A ○ B ○ C ○ D ○
22. A B ○ C ○ D ○
23. ○ A B ○ C ○ D ○
24. A ○ B ○ C ○ D ○

FOR EXAMINER'S USE

Question	Marks
Q1-Q24	/ 72
Q25-27	/ 28
Total	/100

} Negative e.g.

25. (a)

Topic 0: Summary & Takeaways

- Need precise formulation of “Security” for analysis.
- C-I-A requirement.
- Aware of
 - Security Trade-off (usability, cost)
 - Difficulty to achieve
 - Attackers go for the weakest point,
 - Implementation flaw,
 - legacy system, don’t-care,
 - Designers not aware of the attack scenarios (info attacker can access, attacker’s goal)
 - human error.
 - Need to be managed
 - Adversarial thinking in analysis (think like the attacker when analyzing a system)

Topic 1: Summary & takeaways (1)

- Encryption is designed for confidentiality. (not necessary provides integrity, although some method (e.g. AES GCM mode) do.)
- Threat model defines types of attacks to be considered. (threat model is useful in security analysis, not just encryption)
 - Attacker's goal: total break → distinguishability
 - Attacker's capability: ciphertext only → plaintext only → encryption oracle → decryption oracle.

Defender wants a method that is secure under most “humble” attacker’s goal, and stronger attacker’s capability. A system S_1 is more secure than S_1 wrt to the threat model, when for any attack that can be prevented in S_2 , it can also be prevented in S_1
- Notions of “Oracle”.
 - Encryption Oracle, aka CPA (chosen plaintext attack) (oracle’s output can be obtained from, e.g. smart card, probing of server)
 - Decryption Oracle. Padding Oracle Attack (know the detailed mechanism). (oracle output can be derived in many real life system)
- Key strength: Quantifying security by equivalence of best-known attack to exhaustive search. (e.g 2048-bit RSA key has key strength of ~128 bits.)
- No known efficient attacks on modern schemes (e.g. AES) under the intended threat models, but there are pitfalls
 - Implementation error: using known insecure crypto, wrong mode, wrong random sources, mishandling of IV.
 - side-channel information attack. (the intended threat model does not consider information available to the attacker that turns out to be feasible)
 - Implicitly require integrity. (the intended threat models does not consider attackers’ goal that turns out to be crucial)

Summary & takeaways (2)

- Designs of various symmetric key encryption schemes
 - One-time pad. “unbreakable” even if attacker has sufficient time to exhaustively search.
 - Stream Cipher. xor’ing with a “pseudo-random” string.
 - Block Cipher. Mode of operations.
 - CBC: provides some form of integrity. (Secure against CPA. vulnerable to padding oracle attack, BEAST attack, might achieve some forms of integrity. To secure against BEAST, IV needed to be unpredictable (random is more general and will do).) *USE THIS WITH CAUTION*
 - ECB: flexible but leak info. Deterministic (no IV involved). *DO NOT USE*
 - CTR: stream cipher. *USE THIS WITH CAUTION*
 - Secure against CPA; vulnerable to padding oracle attack if padded. No “integrity” at all and easily change (aka malleable). If IV of two ciphertext is the same, leak significant information (in contrast, if IV of two ciphertext under CBC is the same, there are some leakage but not as bad).
 - GCM: Authenticated-Encryption (AE). Achieved both integrity & confidentiality. Secure against Decryption Oracle. Only standardized quite recently and thus not in some legacy systems. *USE THIS*
- Crucial role of IV. (need randomness to have indistinguishability)
 - Why? Make the encryption probabilistic. Eg when no IV.
 - Proper implementation

Topic 2: Summary and takeaways

- Data origin vs Communication Entity Authentication.
- Role of authentication credential. Something (data, device, etc) held by entity for authenticity verification. E.g. password, smart card, biometric. The entity who knows (what you know), holds (what you have), being (who are you) the credential is deemed to be authentic.
- Password strength
 - Online vs offline dictionary attack.
- Attacks on password.
 - Phishing. Bootstrap. Default password. *Phishing* is very effective.
- 2-factor vs 2-steps verification.
 - 2-factor/2-steps is better than single factor/step. E.g. online banking.
 - Compare different combinations.
 - Examples.

Topic 3: Summary & takeaways

- Public Key Encryption.
 - RSA (based on integer factorization. Only integer). ElGamal (based on discrete log of “Algebraic group”. Many choices: ECC).
 - Differences with symmetric key. Implications:
 - Symmetric key requires a *secure channel* to distribute key.
 - Public key requires a *secure broadcast channel* to distribute key.
 - Post-Quantum Crypto (implication after quantum computer become available)
 - Pitfall: using RSA in symmetric key setting. Using textbook RSA.
- Authentication primitives: digest, mac, signature.
 - Security models and application scenarios (Summary Slide 73, 74, 75)
 - Digest
 - No key or other secret information in hash/digest.
 - Hash requirement: Collision resistant. Collision resistant vs 2nd pre-image attack.
 - All hash subjected to Birthday attacks.
 - Signature vs mac
 - (advantage) only require secure broadcast channel; non-repudiation.
 - (disadvantages): efficiency.
- Achieve Confidentiality $\not\Rightarrow$ Achieve Authenticity (if a scheme preserves confidentiality, it might not preserve authenticity)
- Construction:
 - hash: SHA
 - Mac: CBC-mac, HMAC
 - Signature: DSA, hash-and-sign. Special property of RSA for signature (hash-and-encrypt).
- Time-memory-tradeoff in inverting hash

Topic 4: Summary & takeaways

Part 1: PKI

- PKC requires a “secure broadcast channel” to distribute the public keys: PKI.
(consequence of having the wrong public key).
- Certificate: a piece of document that binds a “name” to a “public key” & certified by an authority, called CA. A Certificate contains:
 - ***name, public key, expiry date,***
 - meta info: usages, type of crypto, name of CA, etc,
(meta info often omitted in textbooks but are crucial info, especially “usage”)
 - ***CA's signature***
- PKI: infrastructure to broadcast the key. Comprise of
 - Certificate Authority (CA)
 - The processes on issuing, verification, revocation of certificates.
 - The mechanism of chain-of-trust. (A root CA can certificate other CA. Root CA's public keys are pre-installed or “manually” installed.)
- “Public PKI” usually refers to the one for Internet (often simply called “PKI”). Public PKI’s limitations: Too many root CA’s. A “private PKI” uses a separate group of private CA, forming another chain-of-trust.

Topic 4: Summary & takeaways

Part 2: Channel security

- Protocols: (basic) Authentication, Key Exchange, Authenticated Key Exchange.
 - Authentication. *Adversary extract information and later impersonate.* (Assuming each entity remains the same throughout each session). Unilateral, Mutual.
 - Key exchange: *Adversary sniffs and wants to steal the session key.* No authentication. (PKC, DH)
 - Authenticated key exchange. *Adversary is Mallory* (can sniff, spoof, modify, and thus can take over session) and wants to impersonate and/or steal the session key.
- Putting all together. With crypto primitives, we can obtain a secure (*w.r.t. authenticity & confidentiality, and against Mallory*) channel on top of an underlying unsecure public channel.
 - Method:
 - (1) Use **long-term key** in **Authenticated key exchange** to get a fresh **session key**
 - (2) Use **session key** to protect confidentiality (encrypt) & integrity (mac) of subsequent messages via **authenticated encryption**.
 - Why not just using the long-term key in step (2)?
 - More efficient.
 - Forward secrecy.

Topic 5: Summary & takeaways

- Crypto + PKI can establish end-2-end security (w.r.t. Confidentiality & authenticity) even in the presence of MITM. However, there are other issues in Networking:
 - **Availability** not addressed.
 - Want to mitigate MITM as much as possible (crypto not “perfect”: concerns on implementation flaws, side-channel leakage, can’t hide the existence of interactions, some channels are unprotected or by weak crypto). This means *routing* information need to be protected.
- **Routing.**
 - **Layering.** Intermediate nodes need to see and modify routing info at different “layers”.
 - Protection at different layers. MITM in different layers.
- Monitoring and segmentation. Firewall, Intrusion Detection, DMZ, Server’s zone.
- Some specific attacks:
 - Name resolution attacks (poisoning, spoofing, ARP, DNS).
 - Flaw in protocol (e.g. TLS renegotiation attack).
 - Port Scanning.
 - DDOS, botnets.
- Popular protocols: SSL/TLS (application), IPSEC (network), WPA (link)
 - Which layer to employ e-2-e encryption.
 - Anonymity: TOR, VPN
- What does padlock, https in browser mean? What can a MITM (in link layer, IP layer) get? What can an attacker obtain via DNS spoofing?
- Tools: Wireshark, nmap, nslookup.

Topic 6: Summary & takeaway

- Web (HTTP) is in the “application” layer. It is a popular platform with many services built on top of it.
- HTTPS = HTTP on top of TLS. (inherit strength & weakness of TLS).
- The mechanism of cookies.
 - Cookies might contain authentication credential (authentication) and personal information (privacy).
 - Browser interacts with multiple sites, each site with its cookies. Need separation among different sites. Access control boundary between sites based on *same-source-origin*. Unfortunately, turn out that the separation is weak (ambiguity, difficult for users to visually double-check, etc).
- XSS. XSRF.
- Script injection.
- Human-in-the-loop. (confusing URL and address bar).

Topic 7: Summary and takeaways

- Demonstrate how process integrity can be compromised by modifying values in memory.
- ***Call stack*** facilitates function calls by maintaining and keeping track of ***runtime environments***.
- An element in the stack is a “stack frame”. It contains runtime info such as control flow info (return address), local variables, and parameters. Malicious modification of those info would have significant consequence (next lecture on stack smash + buffer overflow).
- ***Stack Smack***: Due to how OS/compiler maintain the runtime environments, there are opportunities for an attacker to compromise its victim’s call stack (e.g. buffer overflow of local variables into return address).

Topic 8: Summary and takeaway

- Writing a program securely. (common mistake make by programmers).
- Buffer overflow.
- Data representation.
- TOCTOU.
- Code injection.
- ...

Topic 9: Summary & takeaways

- How access control is specified: operation, objects, subjects (principle).

Access Control Matrix, Intermediate control to simplify representation. Examples of intermediate control.
Representation: ACL vs Capability.

- Guideline.

Security perimeter, security boundary, principle of least privilege, segregation, compartment, segmentation, firewall.

- How to share info across security perimeter: “Bridge” and “privilege elevation” (aka “privilege escalation” in the context of attack).

Example in Unix.

- File system: ACL. Intermediate control (user, group, world).
- Objects: program, resources (e.g keyboard, display, network) treated as files.
- Subjects: processes. Each process has (1) **real uid**, specifying its owner (principle) (2) **effective uid** that is used by the reference monitor to check permission.
- Operations: read, write, execute.
- Ring: root (0) vs user(1).
- Example of “bridge”.

Example: Apps in Android vs Users in Linux/Window