

ECE 382N-Sec (FA26):

L0: Course Overview

Neil Zhao

neil.zhao@utexas.edu

Basic Course Information

- **When and Where:** Mon & Weds, 3pm-4:30pm, BUR 134
- **Instructor:** Neil Zhao; neil.zhao@utexas.edu
 - **Rule #1** of this course: Call this guy “Neil”
 - OH: Tue 2pm-3pm, Thu 11am-12pm, or by appointment; EER 5.890
- **TA:** Zhen Tang; tang.zhen@utexas.edu
 - OH: Fri 1pm-2pm; EER 5.652 (for grading-related questions)
- **Format:** Research-oriented. No textbook. No exams. Many papers to read and discuss. One final term project
- **Course Website:** <https://ece382n-sec.github.io/>
 - Up-to-date schedule, paper reading list, and many more resources
- **Canvas:** Used for announcements and assignments
- **Ed Discussion:** Questions and finding teammates

About the Instructor

- Neil Zhao, Assistant Professor, joined our department last Fall
- **Research Focuses:**
 - How to build a secure public cloud (e.g., protect data from other tenants and cloud providers)
 - How to secure an agentic AI system
 - Develop new interesting attacks
 - Simulation
- Also a citrus tree grower



About (Security) Research

- Research is a lot like growing (citrus) trees
- Security is a kind of tax. The lower, the better
- Security is about the worst case, while performance is about the common cases

What do You Need to Know Before Taking This Course

- Good understanding of computer architecture, operating systems, and compilers
- C programming
- Background in security is recommended but not required

Course Requirements

- **Read:**
 - Lightweight reading or watching a presentation before each class
 - Read one or two research papers in-depth per week and write paper reviews
- **Present & Discuss:**
 - Present a research paper and lead the discussion
- **Participate:**
 - Come to classes and contribute to discussions
- **Term Project:**
 - Complete a semester-long **research** project related to security
 - 2~3 students per team

How to Read a Research Paper

- Try to disentangle three “tiers” of knowledge
 - **Concepts:** Most important. Understand how different concepts connect to each other
 - **Methodology:** How do experts think about a problem? How do they evaluate ideas?
 - **Domain-Specific Facts:** Easiest to retrieve (especially now with AI)
- Try to “reproduce” the paper’s key idea—e.g., explain the idea, implement the idea:
 - The difference between knowing the name of something vs. knowing the thing
 - Explained by Richard Feynman: <https://www.youtube.com/watch?v=Zjm8JeDKvdc>
- Some general advice:
 - <https://cseweb.ucsd.edu/~wgg/CSE210/howtoread.html> by William G. Griswold (UCSD)
 - <https://shd.mit.edu/2025/paperReadingGuidance.html> by Mengjia Yan (MIT)
- **Do not rely on reading papers to find new ideas**

Term Project

- 2~3 students per team. It's possible to solo a project, please talk with the instructor
- The project should be security related
- Deliverables
 - Project proposal, midpoint report, and final report
 - Final project presentation
- Timeline:
 - Form a team by Sep 5 (end of the 2nd week)
 - Submit the project proposal by Sep 25 (end of the 5th week)
 - Submit the midpoint report by Nov 6 (end of the 11th week)
 - Final presentation during the last two classes
 - Submit the final report by Dec 11
- You are encouraged to discuss your project with the instructor or TA during OH, especially before submitting your proposal

Types of Projects

- **Attack**
 - Discover new attacks/vulnerabilities
 - Improve existing attacks---e.g., relaxed assumptions, improved success rate, ...
 - Hardware reverse engineering---e.g., the LLC replacement in Intel's non-inclusive cache
 - Direct agents to reproduce known attacks and summarize your learning
- **Defense**
 - Build new defenses to existing/futuristic attacks or threats
 - Reduce the cost of existing defenses---e.g., higher performance, reduced complexity
 - Evaluate the security of existing defenses
- **Empirical measurements**
 - Evaluate the performance and efficiency overhead of different defenses
- **Systematization of Knowledge (SoK)**
 - Like a survey, but more comprehensive and provides a unified framework on a topic

Be mindful of what can realistically be done in 3 months

Hacking Ethics

- Do NOT attack other people's computers, access systems without authorization, or misuse their personal information
- Follow the standard vulnerability disclosure process (special arrangements can be made if your project involves undisclosed vulnerabilities)
- Take proper precaution if you work on an attack-focused term project
- Check with the instructor when in doubt

Grading Policy

- 10%: Class participation, including discussion and pop quizzes
- 5%: Pre-lecture reading and mini quizzes
- 10%: Post-lecture reading and pre-discussion paper reviews
- 15%: Leading paper discussion
- 60%: Term project
 - 3%: Project proposal
 - 7%: Midpoint report
 - 20%: Final presentation
 - 30%: Final report

Gen-AI Policy

Rule-of-thumb: It is acceptable to use them to retrieve and process retrieved information, but not to create original content. Do not let AI replace your thinking. That's how you lose your job to AI

Allowed:

- Find and summarize related work
- Clarify a confusing concept, while checking for hallucination or fabrication
- Proofread your draft for grammar errors
- Help write labor-intensive automation scripts

Prohibited:

- Upload course readings or questions to AI tools to complete mini quizzes or paper reviews
- Ask AI tools to write the project proposal and reports
- Vibe coding without checking its correctness

Pentagon Pizza Theory

Slice of Life: Pizza Orders Soar in D.C.

SPRINGFIELD, Va.— For a quick read on the state of world affairs, one need only look at pizza deliveries to the Pentagon, White House and CIA.

"The news media doesn't always know when something big is going to happen because they're in bed, but our deliverers are out there at 2 in the morning," said Frank Meeks, owner of the 43 Domino's outlets in the Washington area.

Since Jan. 7, late-night deliveries to the Pentagon have increased steadily, from three to 101 Tuesday night, he said. At the White House, 55 pizzas were delivered from 10 p.m. Tuesday to 2 a.m. today.

The one-night record for late-night deliveries at the CIA—21 pizzas—was set Aug. 1, the night before Iraq invaded Kuwait, Meeks said. However, deliveries after 10 p.m. have dropped since Jan. 9, when they reached 15.

"That certainly doesn't indicate that we're not keeping busy," CIA spokesman Mark Mansfield said. "I want to make clear that we're working very hard here."

—Associated Press

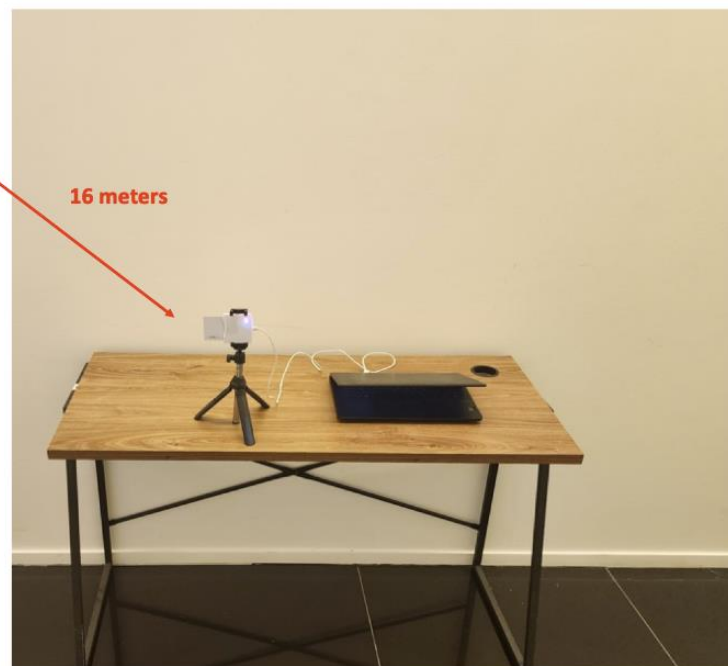
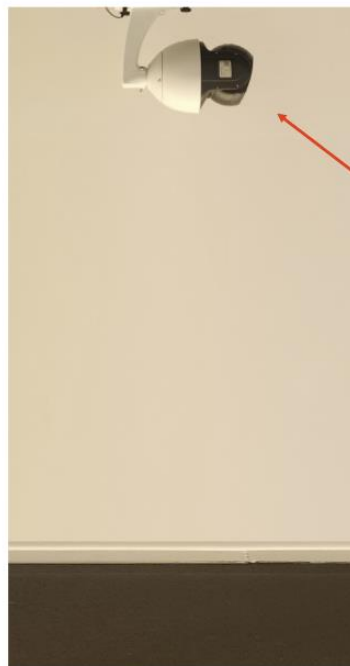
Pizza deliveries to the CIA soared the night before Iraq invaded Kuwait

Source: The LA Times
Jan 16, 1991

Side Channel Example: Video-based Cryptanalysis



ECDSA operation $\Rightarrow$ Higher power consumption
 $\Rightarrow$ LED brightness changes $\Rightarrow$ Captured by a security camera
End-to-end recovery of a 256-bit ECDSA private key from video footage



*Nassi et al., "Video-Based Cryptanalysis: Extracting Cryptographic Keys from Video Footage of a Device's Power LED Captured by Standard Video Cameras", IEEE S&P '24

Side Channel Example: Leaky String Comparison

A naïve implementation that compares strings s1 and s2

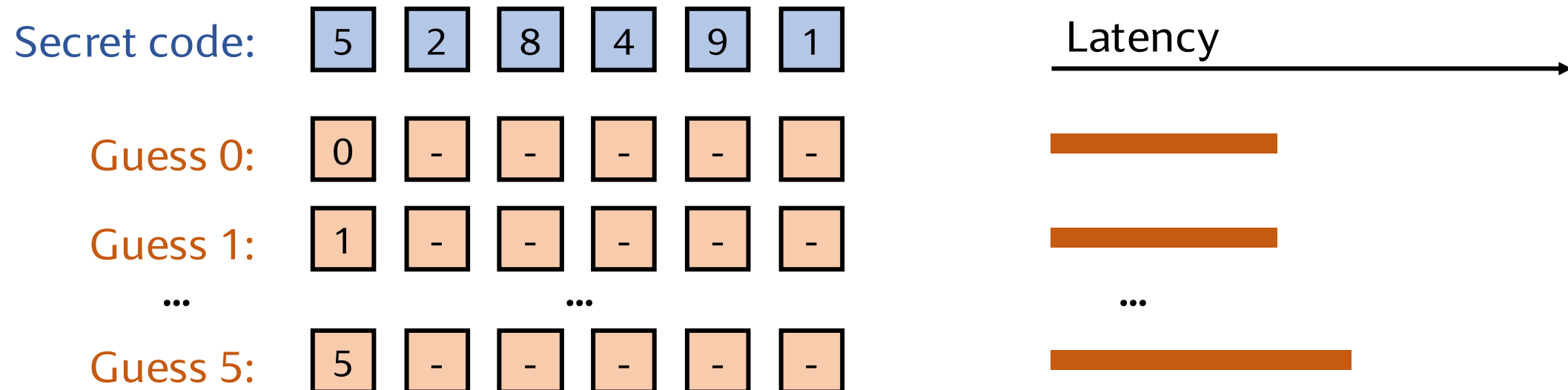
```
bool str_equal(const char *s1, const char *s2) {  
    for (; *s1 && *s2; s1++, s2++) {  
        if (*s1 != *s2) return false; // Early return  
    }  
    return *s1 == *s2;  
}
```

More number of matching bytes between s1 and s2 → Longer execution time

Side Channel Example: Leaky String Comparison

Leaking a secret access code

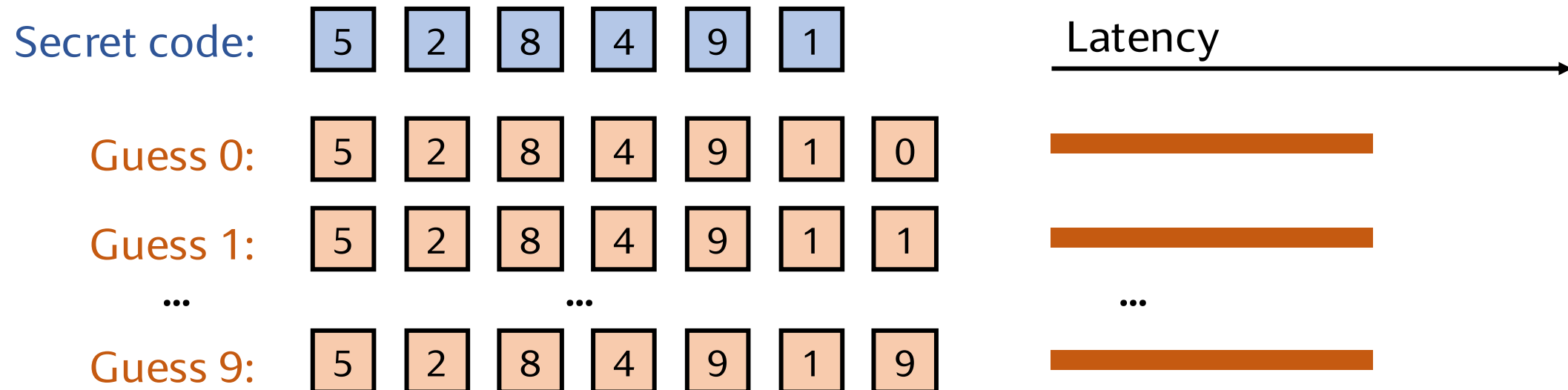
```
bool check_access_code(const char *user_input) {  
    return str_equal(user_input, SECRET);  
}
```



Side Channel Example: Leaky String Comparison

Leaking the length of the secret access code

```
bool check_access_code(const char *user_input) {  
    return str_equal(user_input, SECRET);  
}
```



Launch Code Cracking in WarGames (1983)



If the AI could validate each character of the launch code one-by-one (as shown in the movie):
36 possibilities x 10 characters => At most 360 attempts!
They won't have time to shut it down!