Lecture #12

Specialized Applications and the Future of Mobile Security

SMA 2025

Title Slide

Specialized Applications and the Future of Mobile Security

From Police Body Cams to Quantum Resistance

Today's Agenda

- **Part 1: Case Study: Law Enforcement** - Mobile devices on the front lines.
- **Part 2: Specialized Verticals** - mHealth, Mobile Payments, and Connected Vehicles.
- **Part 3: The 5G Revolution** - Speed, Slicing, and Security.
- **Part 4: AI & Machine Learning** - The new arms race in defense and offense.
- **Part 5: Post-Quantum Cryptography** - Preparing for the "Cryptopocalypse".
- **Part 6: Course Wrap-Up** - Next steps for projects and exams.

Recap from Lecture 11

- **Mobile-IoT Nexus:** The phone is the control plane for the physical world.
- **REST & JSON:** The standard language of IoT.
- **Authentication:** OAuth 2.0 and JWTs are critical but fragile.
- **Threats:** Pivot attacks, side-channels, and the importance of network isolation.

Today's Link: In Lecture 11, we saw how consumer IoT works. Today, we see what happens when "IoT" means a police officer's body camera or a diabetic's insulin pump. The stakes go from "my lights won't turn on" to "life and death."

Part 1: Case Study - Mobile in Law Enforcement

The "Connected Officer"

The Operational Landscape

- **Mobile CAD (Computer Aided Dispatch):** Officers receive 911 calls and dispatch details directly on their phones.
- **Evidence Collection:** Taking photos of crime scenes, recording witness statements.
- **Identity Verification:** Scanning driver's licenses or fingerprints in the field.
- **Body-Worn Cameras (BWC):** Often controlled and reviewed via a paired mobile app.

The Security Challenge: CJIS

- **CJIS (Criminal Justice Information Services):** In the US, the FBI sets strict standards for accessing criminal databases (NCIC).
- **Compliance Requirements:**
- **Advanced Authentication:** Passwords are not enough. MFA is mandatory.
- **Encryption:** FIPS 140-2 validated encryption for data at rest and in transit.
- **Remote Wipe:** Instant capability to kill a lost device.
- **Screen Locks:** Aggressive timeouts (e.g., lock after 1 minute of inactivity).

Solution: Derived Credentials

- **The Problem:** Typing a complex password on a phone screen while chasing a suspect is impossible.
- **The Solution:** Smart Cards + NFC.
- Officer has a physical Smart Card (PIV/CAC) on their vest.
- They tap the card to the phone (NFC) and enter a short PIN.
- The phone derives a temporary credential from the card.

Threat Model: The Lost Device

- **Scenario:** An officer drops their phone during a foot pursuit. A suspect picks it up.
- **Risk:** Access to the entire police database, location of other officers, and active dispatch calls.
- **Defense:**
  - **Geofencing:** The app only works within the jurisdiction.
  - **Context-Aware Auth:** If the device moves too fast (driving) or leaves a zone, require re-authentication.
  - **"Duress" PIN:** Entering a specific PIN wipes the device immediately.

Part 2: Smart Cities & Critical Infrastructure

When the City Itself is the Target

The Smart City Architecture

- **Sensors Everywhere:** Smart streetlights, waste bins, parking meters, and traffic cameras.
- **Connectivity:** LoRaWAN, NB-IoT, and 5G.
- **The Threat:** Ransomware at city scale.
  - Imagine an attacker turning all traffic lights green simultaneously.
  - Imagine shutting down the water treatment plant.

Case Study: The Dallas Siren Hack

- **The Incident:** In 2017, every emergency siren in Dallas, Texas, started blaring at midnight.
- **The Cause:** A "Replay Attack" via unencrypted radio signals.
- **The Fix:** Encryption and Message Integrity (HMAC) on the control signals.
- **Lesson:** Critical infrastructure often relies on legacy, insecure protocols.

Industrial IoT (IIoT) & OT Security

- **IT vs. OT:**
- **IT (Information Technology):** Data, Email, Web. Priority = Confidentiality.
- **OT (Operational Technology):** Robots, Pumps, Centrifuges. Priority = Safety & Availability.

The Purdue Model & Zero Trust

- **Purdue Model:** A layered architecture separating the Enterprise (Level 4) from the Physical Process (Level 0).
- **The Mobile Risk:** A tablet used by a maintenance technician bridges these levels.
- **Defense:**
  - **Micro-segmentation:** The tablet can talk to *this* specific robot, but not the whole factory floor.
  - **Ephemeral Access:** Access is granted only for the duration of the maintenance ticket.

Drone Security (The Flying IoT)

- **UAVs (Unmanned Aerial Vehicles):** Used for delivery, surveillance, and inspection.
- **Remote ID:** The "Digital License Plate" for drones.
- Drones must broadcast their location and pilot ID via Bluetooth/Wi-Fi.
- **GPS Spoofing:** Tricking the drone into flying to a different location.
- **Signal Jamming:** Severing the link between pilot and drone.

Part 3: Specialized Verticals (Deep Dive)

mHealth, Payments, and Auto

mHealth (Mobile Health)

- **IoMT (Internet of Medical Things):** Insulin pumps, pacemakers, continuous glucose monitors (CGM).
- **The Mobile Role:** The phone is the gateway and display for these sensors.
- **Regulation:** HIPAA (USA), GDPR (Europe).
- **Critical Risk:**
  - **Privacy:** Leaking HIV status or mental health records.
  - **Safety:** A hacked insulin pump could deliver a lethal dose.

Code Sample: Reading HealthKit Data (Swift)

```swift
import HealthKit

let healthStore = HKHealthStore()
let heartRateType = HKQuantityType.quantityType(forIdentifier: .heartRate)!

// 1. Request Permission (Explicit User Consent)
healthStore.requestAuthorization(toShare: nil, read: [heartRateType]) { (success, error) in
    if success {
        // 2. Query the Data
        let query = HKSampleQuery(sampleType: heartRateType, predicate: nil, limit: 10, sortDescriptors: nil) { (query, results, error) in
            guard let samples = results as? [HKQuantitySample] else { return }
            for sample in samples {
                print("Heart Rate: \(sample.quantity)")
            }
        }
        healthStore.execute(query)
    }
}
```

Medical Device Security Standards

- **FDA Guidance:** "Postmarket Management of Cybersecurity in Medical Devices."
- **Key Requirement:** SBOM (Software Bill of Materials).
- **Why?** If a library like OpenSSL has a bug, hospitals need to know which pacemakers are affected *immediately*.
- **Code Signing:** Pacemakers must only accept firmware signed by the manufacturer's private key.

Mobile Payments: SoftPOS

- **Evolution:**
- **NFC Payments:** Apple Pay / Google Pay (Phone acts as Card).
- **SoftPOS:** Phone acts as the **Terminal**.

Connected Vehicles (V2X)

- **Phone-as-a-Key (PaaK):** Using Bluetooth (BLE) or Ultra-Wideband (UWB) to unlock and start cars.
- **UWB Security:**
  - **Time-of-Flight:** UWB measures how long the signal takes to travel.
  - **Anti-Relay:** It can detect if the signal is being relayed (amplified) from far away because the time-of-flight will be too long.

Android Automotive OS (AAOS)

- **Concept:** Android is the OS *of the car*, not just running on the phone (Android Auto).
- **Vehicle HAL (Hardware Abstraction Layer):** The bridge between Android apps and the CAN Bus.
- **Permission Model:** Strict permissions like *CAR_SPEED*, *CAR_ENERGY*.
- **Isolation:** Entertainment apps (Spotify) cannot talk to Safety systems (Brakes).

Code Sample: Reading Car Properties

```
import android.car.Car
import android.car.hardware.CarPropertyValue
import android.car.hardware.property.CarPropertyManager

// Connect to the Car Service
val car = Car.createCar(context)
val propertyManager = car.getCarManager(Car.PROPERTY_SERVICE) as CarPropertyManager

// Read the current speed (Requires PERMISSION_CAR_SPEED)
if (context.checkSelfPermission(Car.PERMISSION_SPEED) == PERMISSION_GRANTED) {
    val speed = propertyManager.getProperty<Float>(
        CarPropertyManager.SENSOR_RATE_NORMAL,
        VehiclePropertyIds.PERF_VEHICLE_SPEED,
        CarPropertyManager.AREA_GLOBAL
    )
    Log.d("CarSec", "Current Speed: $speed m/s")
}
```

Part 4: The 5G Revolution

More Than Just Speed

5G Architecture Changes

- **Software Defined Networking (SDN):** The "core" of the network is now software running on standard servers, not specialized hardware.
- **Network Slicing:** The ability to create virtual, isolated networks on the same physical infrastructure.
- **Slice A:** High-speed video for consumers.
- **Slice B:** Ultra-low latency for autonomous cars.
- **Slice C:** High-security slice for government use.

Code Sample: Checking 5G Capabilities

```
import android.telephony.TelephonyManager

val telephonyManager = context.getSystemService(Context.TELEPHONY_SERVICE) as TelephonyManager

// Check if the device is on a 5G NR (New Radio) network
val networkType = telephonyManager.dataNetworkType
if (networkType == TelephonyManager.NETWORK_TYPE_NR) {
    Log.d("G", "Connected to 5G")

    // Advanced: Check for Network Slicing support (Android 12+)
    val capabilities = telephonyManager.getNetworkCapabilities(...)
    if (capabilities.hasCapability(NetworkCapabilities.NET_CAPABILITY_ENTERPRISE)) {
        Log.d("5G", "Connected to Enterprise Slice")
    }
}
```

Fixing the IMSI Catcher (Stingray)

- **The Old Flaw (2G/3G/4G):** Your phone broadcasts its permanent ID (IMSI) in cleartext to connect to a tower.
- **The Attack:** "Stingrays" (fake towers) catch these IMSIs to track people.
- **The 5G Fix:** SUPI (Subscription Permanent Identifier) vs. SUCI (Subscription Concealed Identifier).
- The phone **encrypts** its ID using the carrier's public key *before* sending it.
- The tower sees only the encrypted SUCI. Only the home carrier can decrypt it.

Edge Computing & Security

- **MEC (Multi-access Edge Computing):** Moving servers closer to the tower.
- **Use Case:** AI processing for video surveillance, AR/VR.
- **Security Risk:** Physical Security.
- A cloud data center is a fortress.
- An Edge node might be a server rack at the base of a cell tower in a field.

Part 5: AI & Machine Learning

The New Arms Race

AI for Defense

- **On-Device Anomaly Detection:**
- Google Play Protect uses ML to scan apps for malicious *behavior*, not just known signatures.
- Example: "This calculator app is trying to access the microphone and send data to a Russian IP. That is anomalous."
- Learning how *you* hold your phone, how fast you type, and where you usually go.
- **Continuous Auth:** If the behavior changes (e.g., typing speed doubles), lock the phone.

Code Sample: On-Device Anomaly Detection (TFLite)

```
import org.tensorflow.lite.Interpreter

// Load the TFLite model (trained to recognize bad behavior)
val tflite = Interpreter(loadModelFile("behavior_model.tflite"))

// Input: [TypingSpeed, GyroscopeX, GyroscopeY, AppSwitchCount]
val inputFeatures = floatArrayOf(120.0f, 0.01f, -0.05f, 5.0f)
val outputProbability = Array(1) { FloatArray(1) }

// Run inference
tflite.run(inputFeatures, outputProbability)

// Check result
val fraudScore = outputProbability[0][0]
if (fraudScore > 0.9) {
    lockDevice() // High probability of unauthorized user
}
```

Federated Learning

- **The Privacy Problem:** To train a good AI, you need everyone's data. But you can't send everyone's private photos to the cloud.
- **The Solution:** Federated Learning.
  - Send the *Model* to the Phone.
  - Train the model locally on the user's data.
  - Send only the *Weights* (math updates) back to the cloud.

AI for Offense

- **Deepfakes:**
- **Voice Cloning:** Attackers clone a CEO's voice to authorize a wire transfer.
- **Video Injection:** Bypassing "Liveness Checks" in banking apps by injecting a deepfake video into the camera stream.
- Perfect grammar, personalized context.
- "Spear phishing" at scale.

Adversarial Machine Learning

- **Concept:** Attacking the model itself.
- **Evasion Attacks:** Modifying a malware file slightly (adding "noise") so that the AI classifier thinks it is benign.
- **Poisoning Attacks:** Injecting bad data into the training set so the model learns the wrong things.
- Example: Teaching a self-driving car that a "Stop" sign with a specific sticker on it is actually a "Speed Limit 45" sign.

Part 6: Post-Quantum Cryptography (PQC)

The Looming Cryptopocalypse

The Quantum Threat

- **Shor's Algorithm:** A quantum algorithm that can factor large prime numbers efficiently.
- **The Victim: RSA and ECC.**
- All the public key crypto we use today (HTTPS, Digital Signatures, VPNs) will be broken instantly by a sufficiently powerful quantum computer.

"Store Now, Decrypt Later" (SNDL)

- **Why worry now?** Quantum computers are years away.
- **The Attack:** Nation-states are harvesting encrypted traffic *today* and storing it in massive data centers.
- **The Payoff:** In 10-15 years, when they have a quantum computer, they will decrypt all that stored data.
- **Impact:** Long-lived secrets (intelligence, trade secrets, health records) are already at risk.

The Solution: NIST PQC Standards

- **NIST Competition:** A multi-year global competition to find new algorithms resistant to quantum attacks.
- **The Winners (2022/2024):**
  - **Key Encapsulation (Encryption):** CRYSTALS-Kyber (ML-KEM).
  - **Digital Signatures:** CRYSTALS-Dilithium (ML-DSA), FALCON, SPHINCS+.

The Migration Challenge on Mobile

- **Key Sizes:** PQC keys are much larger than ECC keys.
- ECC Public Key: ~32 bytes.
- Kyber Public Key: ~1,000 bytes.
- Dilithium Signature: ~2,500 bytes.
- If Kyber turns out to have a bug, ECC protects us.
- If Quantum arrives, Kyber protects us.

Code Sample: Hybrid Key Exchange (Conceptual)

```
// Hybrid Key Exchange: ECC + Kyber

// 1. Generate ECC Keypair (Classical)
KeyPair eccPair = generateECCKeyPair();

// 2. Generate Kyber Keypair (Post-Quantum)
KyberKeyPair pqcPair = Kyber.generateKeyPair();

// 3. Combine Public Keys
byte[] hybridPublicKey = combine(eccPair.getPublic(), pqcPair.getPublic());

// 4. Send to Server...
// Server encapsulates shared secret using BOTH keys.

// 5. Decrypt Shared Secret
byte[] sharedSecretECC = eccDecrypt(serverResponse.eccPart, eccPair.getPrivate());
byte[] sharedSecretPQC = kyberDecrypt(serverResponse.pqcPart, pqcPair.getPrivate());

// 6. Derive Final Session Key
byte[] sessionKey = KDF(sharedSecretECC + sharedSecretPQC);
```

Crypto Agility

- **The Concept:** Designing your app so you can switch cryptographic algorithms without rewriting the code.
- **How:**
  - Don't hardcode "SHA-256".
  - Use configuration files or negotiation protocols to select the algorithm.

Appendix A: 5G Network Slicing Deep Dive

- **NSSF (Network Slice Selection Function):** The core component that decides which slice a device gets.
- **URLLC (Ultra-Reliable Low Latency Communications):** The "slice" for autonomous cars and surgery.
- **mMTC (Massive Machine Type Communications):** The "slice" for billions of IoT sensors.

Appendix B: Matter Protocol Security

- **PKI (Public Key Infrastructure):** Every Matter device has a unique certificate signed by the CSA (Connectivity Standards Alliance).
- **PASE (Passcode Authenticated Session Establishment):** Secure pairing using a setup code.
- **CASE (Certificate Authenticated Session Establishment):** Secure operation after pairing.

Appendix C: Automotive CAN Bus

- **Controller Area Network:** A broadcast network. No source IP, no destination IP.
- **The Vulnerability:** If you are on the bus, you can read everything and inject anything.
- **The Fix:** Automotive Ethernet and SecOC (Secure Onboard Communication).

Appendix D: Drone Remote ID Standards

- **ASTM F3411:** The standard for Remote ID.
- **Broadcast:** Bluetooth 4.0/5.0 or Wi-Fi Beacon.
- **Data:** Lat/Long, Altitude, Velocity, Pilot ID, Timestamp.
- **Security:** The message is signed to prevent spoofing.

Appendix E: AI Model Poisoning Defense

- **Data Sanitization:** rigorously filtering training data.
- **Differential Privacy:** Adding noise to data so individual contributions cannot be identified.
- **Model Monitoring:** Continuously checking the model's accuracy in production to detect drift.

Appendix F: Recommended Reading

- **"The Tangled Web"** by Michal Zalewski.
- **"Android Hacker's Handbook"** by Drake et al.
- **"iOS Hacker's Handbook"** by Miller et al.
- **"Smart Cities: Big Data, Civic Hackers, and the Quest for a New Utopia"** by Anthony Townsend.

Appendix G: Tools for 5G Security

- **SCAT (Signaling Collection and Analysis Tool):** For analyzing cellular traffic.
- **srsRAN:** Open source 4G/5G software radio suite.
- **Wireshark:** Now supports 5G protocols (NGAP, PFCP).

The Journey We've Taken

- **Foundations:** Linux kernel, Sandboxing, Permissions.
- **Offense:** Malware, Phishing, Network Attacks.
- **Defense:** Secure Coding, Cryptography, IAM.
- **Management:** MDM, MAM, Corporate Strategy.
- **Future:** IoT, AI, Quantum.

Final Thought

"Security is not a destination, it is a journey."

The technology we discussed today (5G, AI) will be obsolete in 5 years. The specific tools will change. But the **mindset** ”adversarial thinking, defense in depth, least privilege” will remain constant.

Q&A

Questions?

Thank You!

Have a wonderful Christmas Break!