CTT Wireless Security Auditor

Professional Security Testing Suite
Temporal WPA/WPA2 Key Derivation System

Americo Simoes

November 2024

Abstract

This whitepaper presents the CTT Wireless Security Auditor, a professional-grade security testing tool that applies Convergent Time Theory (CTT) principles to wireless network security auditing. The system demonstrates a novel approach to WPA/WPA2 key derivation using temporal resonance sampling at prime microsecond windows ($\alpha = 0.0302$, prime window = 10007 μs). This tool is designed for authorized security professionals, penetration testers, and network administrators to assess wireless security postures.

1 Introduction

Wireless network security remains a critical concern for enterprise and consumer environments. Traditional brute-force and dictionary-based attacks on WPA/WPA2 protocols are computationally intensive and time-consuming. The CTT Wireless Security Auditor introduces a temporal-based approach to security testing, leveraging precise timing and resonance windows to derive potential keys.

1.1 Purpose

This tool serves multiple legitimate security purposes:

- Authorized penetration testing of wireless networks
- Security posture assessment for enterprise networks
- Educational demonstrations of WPA vulnerabilities
- Research into temporal-based cryptographic analysis

1.2 Legal Compliance

IMPORTANT: This tool must only be used on networks you own or have explicit written authorization to test. Unauthorized access to wireless networks is illegal under:

- Computer Fraud and Abuse Act (18 U.S.C. § 1030) - USA
- Computer Misuse Act 1990 - UK

- EU Cybersecurity Regulations
- International cybercrime legislation

2 Technical Architecture

2.1 CTT Principles

The Convergent Time Theory framework operates on three core constants:

$$\alpha = 0.0302 \quad \text{(Framework transition coefficient)} \tag{1}$$

$$\omega_+ = 587 \text{ kHz} \quad \text{(Positive resonance frequency)} \tag{2}$$

$$\omega_- = 293.5 \text{ kHz} \quad \text{(Negative resonance frequency)} \tag{3}$$

2.2 Prime Window Alignment

The system operates on a prime resonance window of 10007 microseconds. Key derivation occurs only when the system achieves temporal alignment:

$$t \bmod 10007 < 10 \quad \text{or} \quad t \bmod 10007 > 9997 \tag{4}$$

where t represents the current timestamp in microseconds.

2.3 Temporal Sampling Process

1. **Prime Window Detection**: Monitor system time at microsecond precision
2. **Alignment Wait**: Use μsleep(1) to achieve precise boundary alignment
3. **Temporal Sample**: Extract 32-bit sample from temporal field
4. **Key Derivation**: Generate candidate password from temporal data
5. **PMK Calculation**: Apply PBKDF2-HMAC-SHA1 with 4096 iterations
6. **PTK Generation**: Compute Pairwise Transient Key using standard WPA PRF
7. **MIC Verification**: Compare calculated MIC against captured handshake

3 WPA/WPA2 Protocol Implementation

3.1 PMK (Pairwise Master Key)

The PMK is derived using the standard WPA approach with the temporally-sampled password:

$$\text{PMK} = \text{PBKDF2}(\text{SHA1}, \text{password}_{\text{temporal}}, \text{SSID}, 4096, 256) \tag{5}$$

3.2 PTK (Pairwise Transient Key)

The PTK is computed using the standard WPA PRF function:

$$\text{PTK} = \text{PRF-X}(\text{PMK}, \text{label}, \text{data}) \tag{6}$$

where:

- label = "Pairwise key expansion"
- data = $\min(\text{AA}, \text{SPA}) \| \max(\text{AA}, \text{SPA}) \| \min(\text{ANonce}, \text{SNonce}) \| \max(\text{ANonce}, \text{SNonce})$

4 System Components

4.1 Core Functions

```
uint64_t get_time_us();
void wait_for_prime_window();
int derive_temporal_wpa_key(...);
int verify_temporal_key(...);
void audit_wpa_temporal(...);
```

Listing 1: Temporal Key Derivation

4.2 Dependencies

- OpenSSL (libssl, libcrypto) - Cryptographic functions
- POSIX time functions - Microsecond precision timing
- Standard C library

5 Usage Guidelines

5.1 System Requirements

- Linux operating system (Fedora, Ubuntu, Kali recommended)
- Wireless adapter with monitor mode support
- Root/sudo privileges
- Dependencies: libpcap-dev, libssl-dev, aircrack-ng suite

5.2 Installation

```
# Install dependencies (Fedora)
sudo dnf install libpcap-devel openssl-devel aircrack-ng

# Install dependencies (Debian/Ubuntu)
sudo apt install libpcap-dev libssl-dev aircrack-ng

```

```
# Compile the tool
cd ctt-wireless-security
make
```

5.3 Monitor Mode Setup

Before using the tool, your wireless interface must be in monitor mode:

```
# Stop network manager interference
sudo systemctl stop NetworkManager

# Enable monitor mode using airmon-ng
sudo airmon-ng start wlan0

# Verify monitor interface created (typically wlan0mon)
iwconfig
```

5.4 Basic Operation

Step 1: Start the CTT Wireless Auditor

```
sudo ./aircrack-ctt wlan0mon
```

The tool will begin listening for beacons and display detected networks:

```
Found SSID: TargetNetwork (AP: aa:bb:cc:dd:ee:ff)
Found SSID: OtherNetwork (AP: 11:22:33:44:55:66)
```

Step 2: Lock to Target Channel

For optimal handshake capture, lock your interface to the target AP's channel:

```
# Identify channel (from beacon info or use airodump-ng)
# Lock to specific channel (e.g., channel 6)
sudo iw dev wlan0mon set channel 6
```

Step 3: Capture Handshake

In a second terminal, force client reconnection using deauthentication:

```
# Deauth all clients from target AP
sudo aireplay-ng --deauth 5 -a AA:BB:CC:DD:EE:FF \
    --ignore-negative-one wlan0mon

# Or deauth specific client
sudo aireplay-ng --deauth 5 -a AA:BB:CC:DD:EE:FF \
    -c CLIENT_MAC --ignore-negative-one wlan0mon
```

The CTT auditor will display when handshake packets are captured:

```
Captured ANonce from AP
Captured SNonce from Client
Captured EAPOL Message 3 for MIC verification
Complete WPA handshake captured!
Starting CTT temporal cracking...
```

Step 4: Temporal Key Derivation

Once a complete handshake is captured, the tool automatically:

1. Aligns to prime resonance windows (10007 μs)
2. Samples temporal field for password candidates
3. Derives PMK using PBKDF2-HMAC-SHA1
4. Calculates PTK using WPA PRF
5. Verifies MIC against captured EAPOL frame

Successful derivation outputs:

```
WPA NETWORK CRACKED VIA TEMPORAL RESONANCE!
Password: [derived_password]
SSID: TargetNetwork
AP MAC: aa:bb:cc:dd:ee:ff
Client MAC: 11:22:33:44:55:66
Time to crack: X seconds
```

5.5 Advanced Usage

Multi-Channel Scanning

To discover networks across all channels:

```
# Terminal 1: Run auditor
sudo ./aircrack-ctt wlan0mon

# Terminal 2: Channel hopping script
for channel in {1..11}; do
    sudo iw dev wlan0mon set channel $channel
    sleep 3
done
```

Targeted Network Analysis

Focus on a specific SSID by monitoring its channel continuously and using selective deauthentication during peak client activity periods.

5.6 Cleanup

After testing, restore normal wireless operation:

```
# Stop monitor mode
sudo airmon-ng stop wlan0mon

# Restart NetworkManager
sudo systemctl start NetworkManager
```

5.7 Integration

For enterprise security platforms, the tool provides programmatic interfaces for:

- Live packet capture and analysis
- Real-time handshake detection
- Temporal key derivation engine
- Automated security auditing

6 Commercial Licensing

6.1 License Tiers

1. **Research License**: Academic and non-commercial research use
2. **Professional License**: Individual security consultants
3. **Enterprise License**: Corporate security teams (unlimited seats)
4. **OEM License**: Integration into security products

6.2 Support Services

- Technical support (email, phone, video conference)
- Custom feature development
- Integration consulting
- Training and certification programs

6.3 Contact Information

For licensing inquiries:
Email: amexsimoes@gmail.com

7 Security Considerations

7.1 Responsible Disclosure

Any vulnerabilities discovered using this tool should be:

1. Reported to affected parties immediately
2. Disclosed responsibly following industry standards
3. Not exploited for malicious purposes
4. Documented for security improvement

7.2 Ethical Guidelines

Users must:

- Obtain written authorization before testing
- Maintain confidentiality of findings
- Follow applicable laws and regulations
- Use tool only for defensive security purposes

8 Conclusion

The CTT Wireless Security Auditor represents a novel approach to network security testing, combining traditional WPA protocol analysis with temporal resonance principles. This tool serves as both a practical security assessment platform and a research vehicle for exploring time-based cryptographic analysis.

8.1 Future Development

Planned enhancements include:

- WPA3 support
- GPU acceleration for temporal sampling
- Cloud-based distributed auditing
- Machine learning integration for pattern recognition
- Real-time dashboard and reporting

9 References

1. IEEE 802.11i-2004 - Wireless Network Security Standard
2. RFC 2898 - PBKDF2 Specification
3. Convergent Time Theory Framework (Simões, 2025)
4. WPA/WPA2 Key Management Protocols
5. NIST Cybersecurity Framework
