

# SENG 360 - Security Engineering Distributed Systems

Jens Weber

Fall 2022



# Learning Objectives



At the end of this class you will be able to

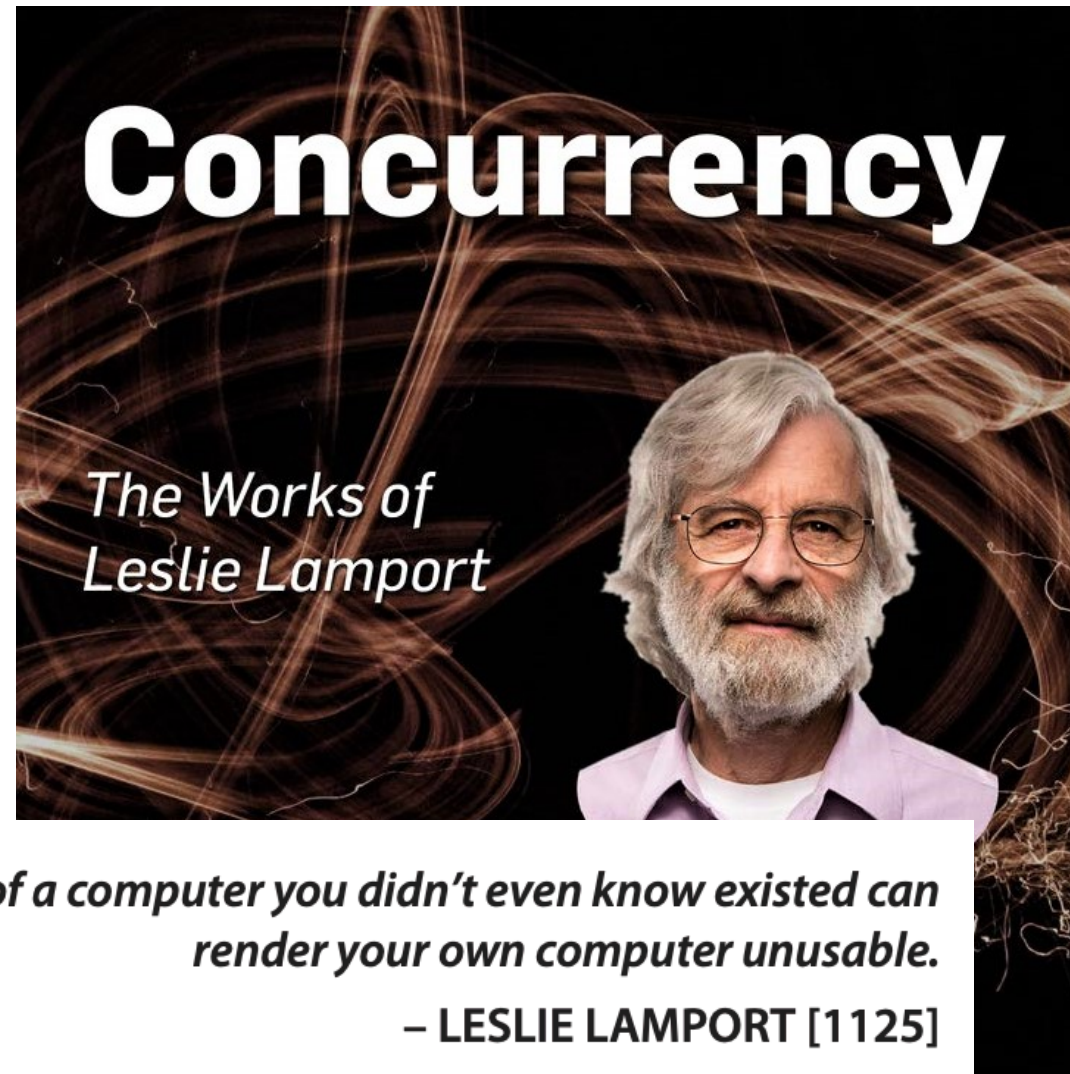
- explain why securing distributed systems is hard
- describe attacks related to distributed systems
- define challenges and principles related to designing secure distributed systems



# Concurrency

Constantly increasing

Assuring correctness is hard



*A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable.*

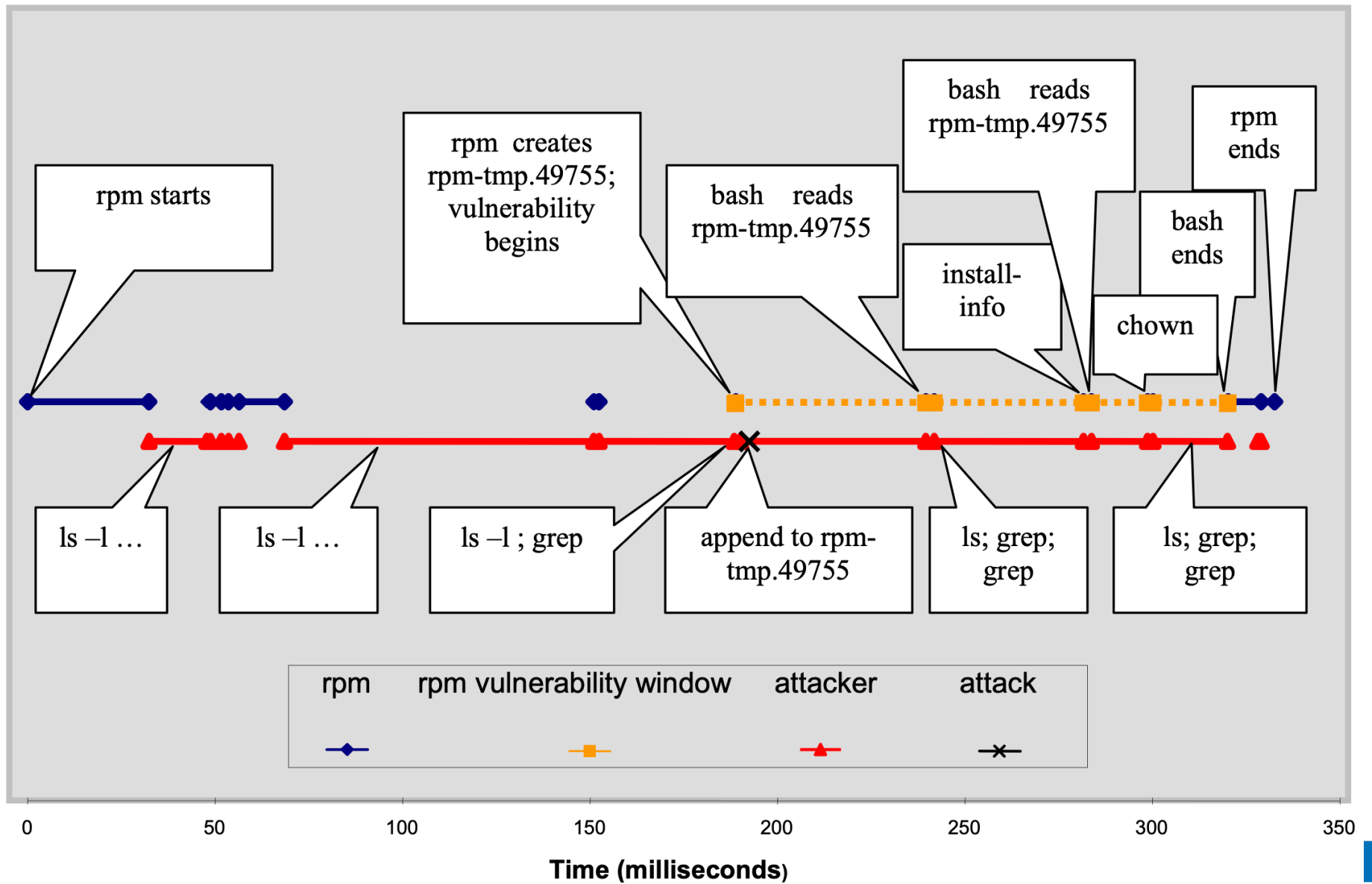
– LESLIE LAMPORT [1125]

# Using old data vs. paying to propagate state

TOCTOU (time -of-check-to-time-of-use) attacks exploit race conditions

Victim	Attacker
<pre>if (access("file", W_OK) != 0) {     exit(1); }  fd = open("file", O_WRONLY); // Actually writing over /etc/passwd write(fd, buffer, sizeof(buffer));</pre>	<pre>// // // After the access check symlink("/etc/passwd", "file"); // Before the open, "file" points to the password database // //</pre>





# Another example: Certificate Revocation

[Services ▼](#)[Solutions ▼](#)[News](#)[Company ▼](#)[Reso](#)

## How certificate revocation (doesn't) work in practice

13th May, 2013

Certificate revocation is intended to convey a complete withdrawal of trust in an SSL certificate and thereby protect the people using a site against fraud, eavesdropping, and theft. However, some contemporary browsers handle certificate revocation so carelessly that the most frequent users of a site and even its administrators can continue using an revoked certificate for weeks or months without knowing anything is amiss. Recently, this situation was clearly illustrated when a busy e-commerce site was still using an intermediate certificate more than a week after its revocation.




# Locking to prevent inconsistent updates

For example:

- preauthorization of credit card (locks \$500)
- billed later

AVIS

Welcome, DONNA

1	2	3 Rental Options	4
<b>Pick-Up</b> Friedman Memorial Airport, SUN ⓘ Wed, Jul 24, 2:30 PM	<b>Return</b> Friedman Memorial Airport, SUN ⓘ Wed, Aug 21, 11:00 AM	Base Rate Mileage: Unlimited Rental Options Discount Codes Fees & Taxes <b>Estimated Total</b> Amount Prepaid (USD)	\$627.30 \$0.00 \$155.93 <b>\$783.23</b> 783.23
 <b>Economy</b> Ford Fiesta or similar ⓘ Automatic Transmission		<p><b>NOT the total price. That will only show up after you charge your credit card, and it will be \$100 more, at least.</b></p>	

# Order of updates

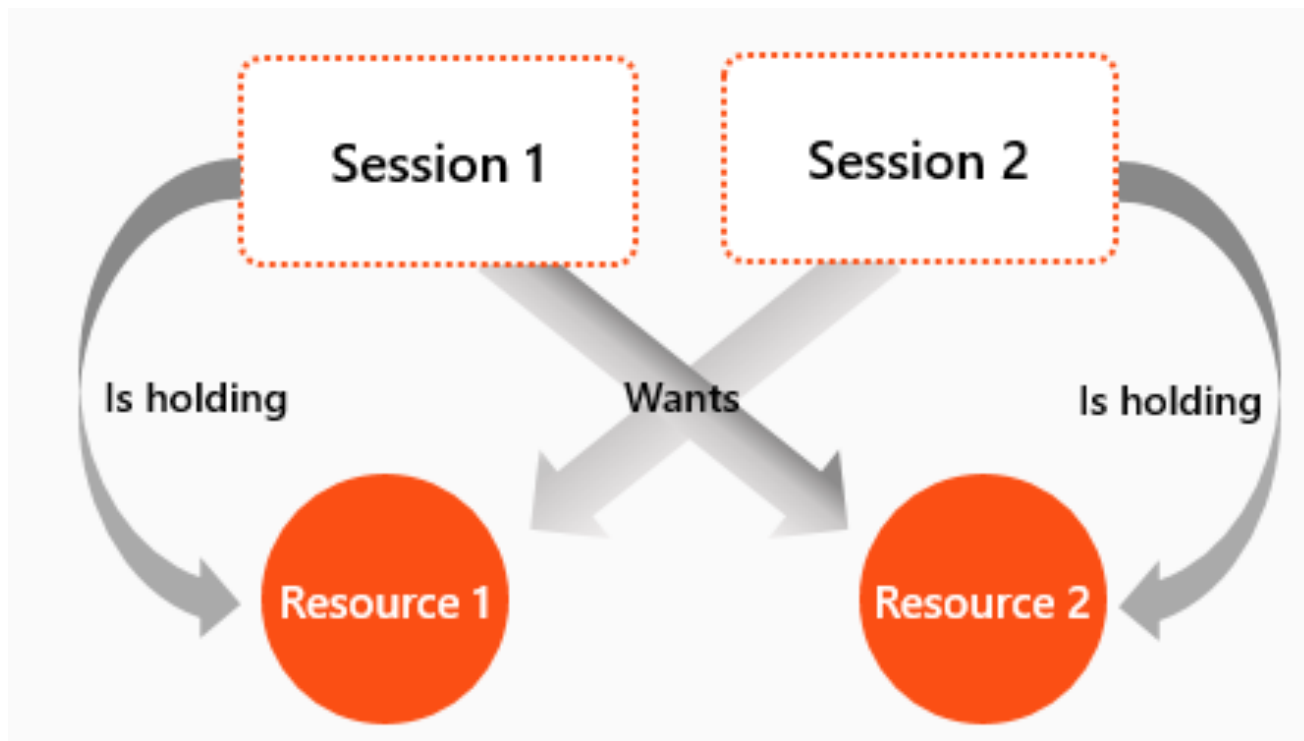
For example:

1. credit \$4000
2. debit \$3000

or the other way around makes a difference.



# Deadlocks



# Non-convergent state

ACID transactions do not scale to distributed systems

Distributed systems often use BASE (Basically Available, Soft state, Eventual consistency) semantics

State is expected to *converge* eventually (when new events taper off)

# Secure time

Time plays important role in distributed systems

- e.g., timed-credentials (e.g., Kerberos tickets)
- event order time, etc.

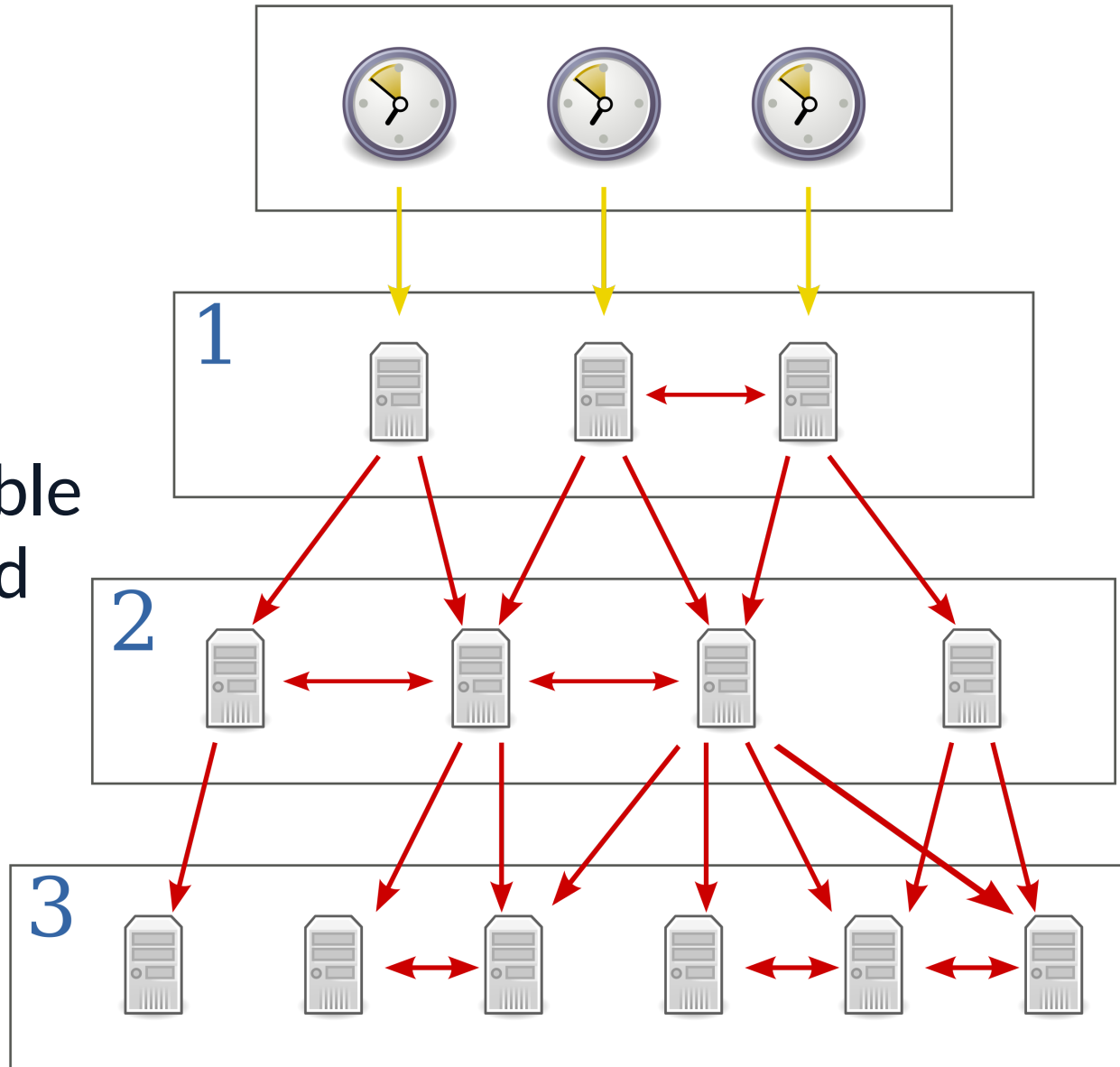
Attacks on time:

- cinderella attack (wind clock)
- desynchronization attack

Network Time Protocol (NTP)

# NTP

- Hierarchical
- Voting
- MITM attack possible unless crypt. signed
- but then DoS amplifier

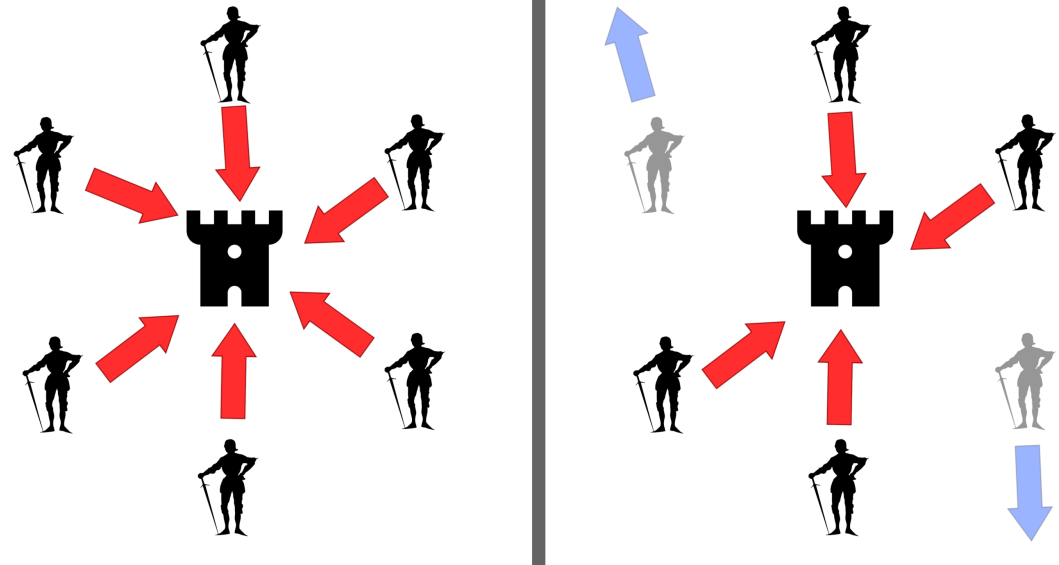


# Fault Tolerance and Failure Recovery

Failure model: **Byzantine failure**

Given  $n$  generals, how many conspiring generals  $t$  can be tolerated? ( $3t+1 \leq n$ )

- > signing helps
- > silence is better than lies



# Redundancy and fail-stop processes

**Redundancy** makes systems more resilient (availability) but also more complex, and more vulnerable (confidentiality)

**Fail stop processors** stop operation when failure is detected

Can be combined

# Redundancy at different levels

- **Hardware** (e.g., RAID, CPUs)
- **Process group** (e.g., multiple systems processing requests and comparing results) (aka hot redundancy)
- **Backup** (e.g., copy of system taken at “checkpoints” that can take over) (aka cold redundancy)
- Note: Distinguish between *failover* and *fallback* (the latter implies a service degradation)



# Fallbacks and Security

more availability

but at a price  
(integrity, confidentiality)

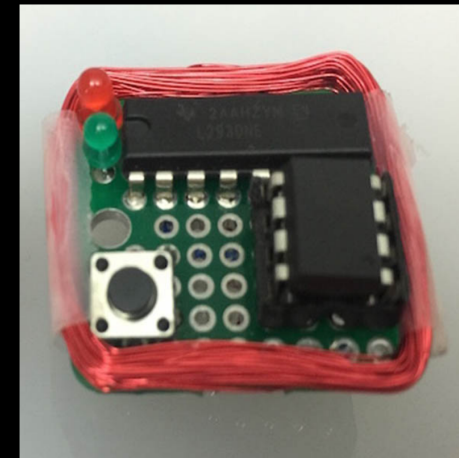
## DEFEATING CHIP AND PIN WITH BITS OF WIRE

November 25, 2015 by [Brian Benchoff](#)

120 Comments

One of many ways that Americans are ridiculed by the rest of the world is that they don't have chip and PIN on their credit cards yet; US credit card companies have been slow to bring this technology to millions of POS terminals across the country. Making the transition isn't easy because until the transition is complete, the machines have to accept both magnetic stripes and chip and PIN.

**This device can disable chip and PIN**, wirelessly, by forcing the downgrade to magstripe. [Samy Kamkar] created the MagSpoofer to explore the binary patterns on the magnetic stripe of his AmEx card, and in the process also created a device that works with drivers licenses, hotel room keys, and parking meters.



The electronics for the MagSpoofer are incredibly simple. Of course a small microcontroller is necessary for this build, and for the MagSpoofer, [Samy] used the ATtiny85 for the 'larger' version (still less than an inch square). A smaller, credit card-sized version used an ATtiny10. The rest of the schematic is just an H-bridge and a coil of magnet wire – easy enough for anyone with a soldering iron to put together on some perfboard.

By pulsing the H-bridge and energizing the coil of wire, the MagSpoofer emulates the swipe of a credit card – it's all just magnetic fields reversing direction in a very particular pattern. Since the magnetic pattern on any credit card can be easily read, and [Samy]