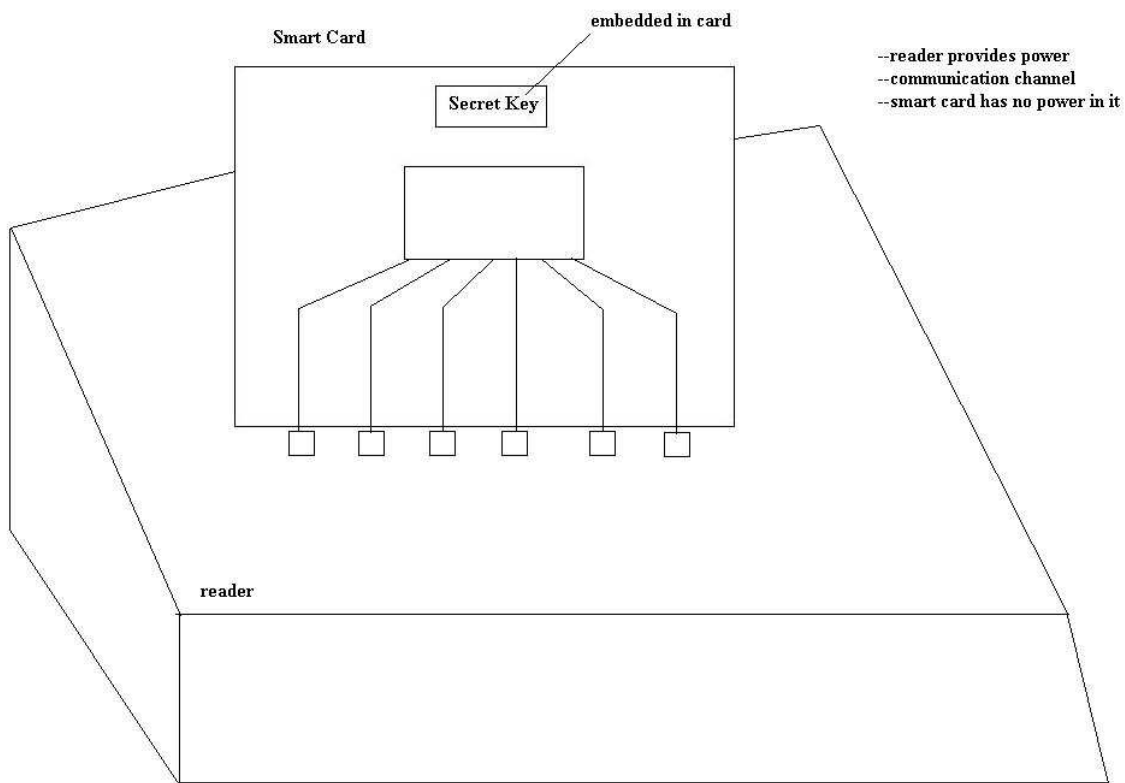
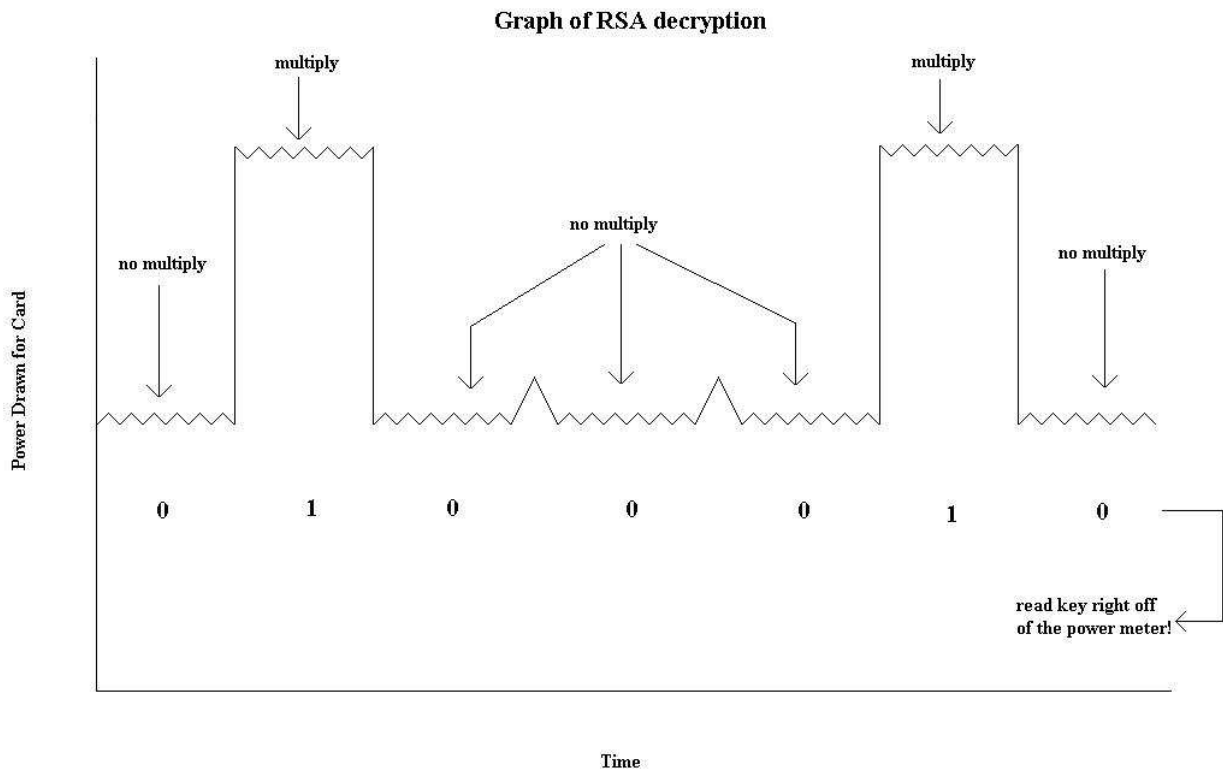


Side Channel Attacks

Side Channel – other side communication channels that exist other than the main channel (examples: electricity, amount of time of a connection, etc.)

-Remote Timing Attacks – originally for smart cards (replaces credit cards with a microchip with wires going into the edge of the card, connected to a reader box, then a wire from the reader box to the credit card company).





Threat Model – Nowadays with a credit card, it is used at a merchant, the account number (data) is read from the card, then the account is charged (suppose a waiter at a restaurant can use the card number for identity theft). Assume the merchant is bad, but we want to purchase from him, but not allow them to make any other purchases. This is a Power Analysis Attack (done a lot with remote timing attacks).

When a smart card is connected to a reader, it gives the exact amount of time it takes to make a decryption.

-Local Timing Attack – a type of Remote Timing Attack in which a custom reader is created to do thousands of encryptions/decryptions instead of only one. Create pairs (C_1, t_1) , (C_2, t_2) , (C_3, t_3) , ..., (C_n, t_n) , figure out how long it takes to compute each decryption pair, then guess each bit independently.

Attack:

```
For i = 0 to L
  Guess bit i of the key is 0.
  Divide samples into slow and fast groups.
  If timing differs
    Bit i = 0
  Else
    Bit i = 1
```

Example:

```
(C1, t1) is slow
(C2, t2) is slow
(C3, t3) is fast
...
(Cn, tn) is slow
```

If key bits are guessed correctly and divided correctly, then the key is obtained. Otherwise, the average timing of each decryption is about the same. This can be done within thousands of messages/transactions.

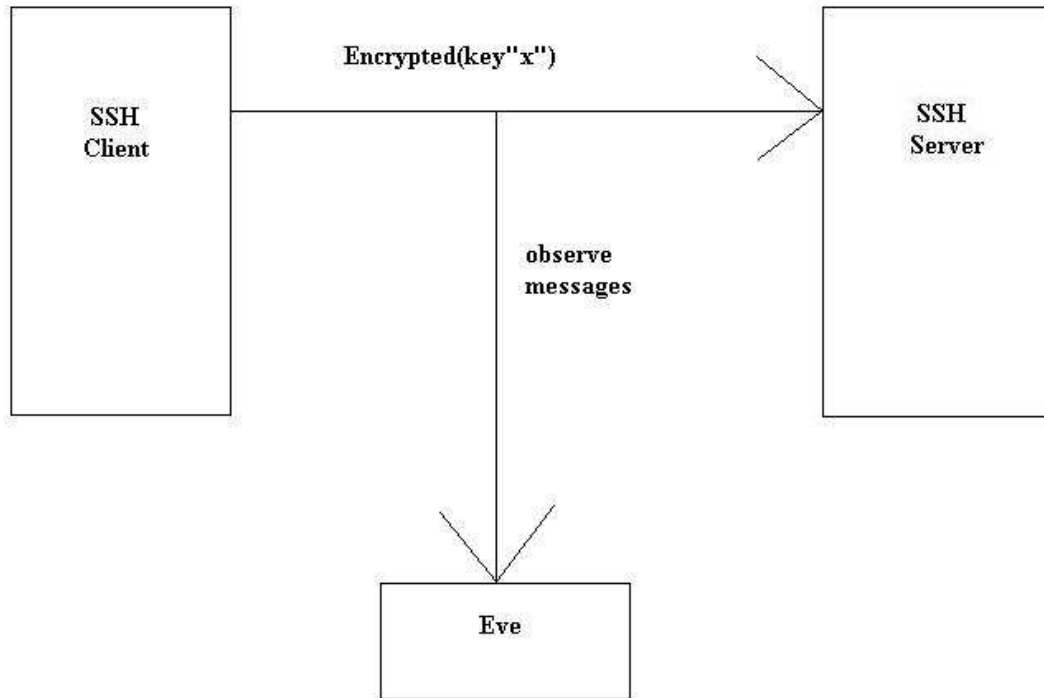
Assuming timing measures are good, and exponentiation has certain properties, if SSL does the same type of algorithm with timing, it can be broken too, but people didn't consider this possibility. However, SSL uses other algorithms besides simple multiplication and exponentiation (as opposed to the smart card).

SSL has been broken simply by doing a handshake and watching the timing of RSA decryptions. It was also done behind wiring and an entire network! It took roughly between 400 and 800 messages per bit of the key to be guessed, and since the key was about a thousand bits, it took roughly one million messages to accomplish this!

This type of remote timing attack extracted the key from a real SSL server with one million messages. Thus, the attacker was now able to impersonate the server with the SSL key!

Another Side Channel Attack

- Keystroke Timing
 - From Packet Timings,
 - One can infer keystroke timing
 - There is bias in interkey typing (consider how you type letters on the keyboard using your hand, and the different lengths in time when typing a pair of letters such as x and y).
 - Thus, one can infer what a person is typing from interkey timings!



Model for obtaining keyboard information

1. Develop Model of Interkey Timing
Probability[KeyPair = "xy" given the interkey timing is 37 ms] (conditional probability)
2. Viterbi's Algorithm: Takes observed timings and outputs the most likely key sequence!

If there is a microphone in the room, one not only knows the timing, but even the frequency/sound of each unique key. Then, figure out each unique keyboard sound (based on the type of keyboard), and infer what a user is typing!

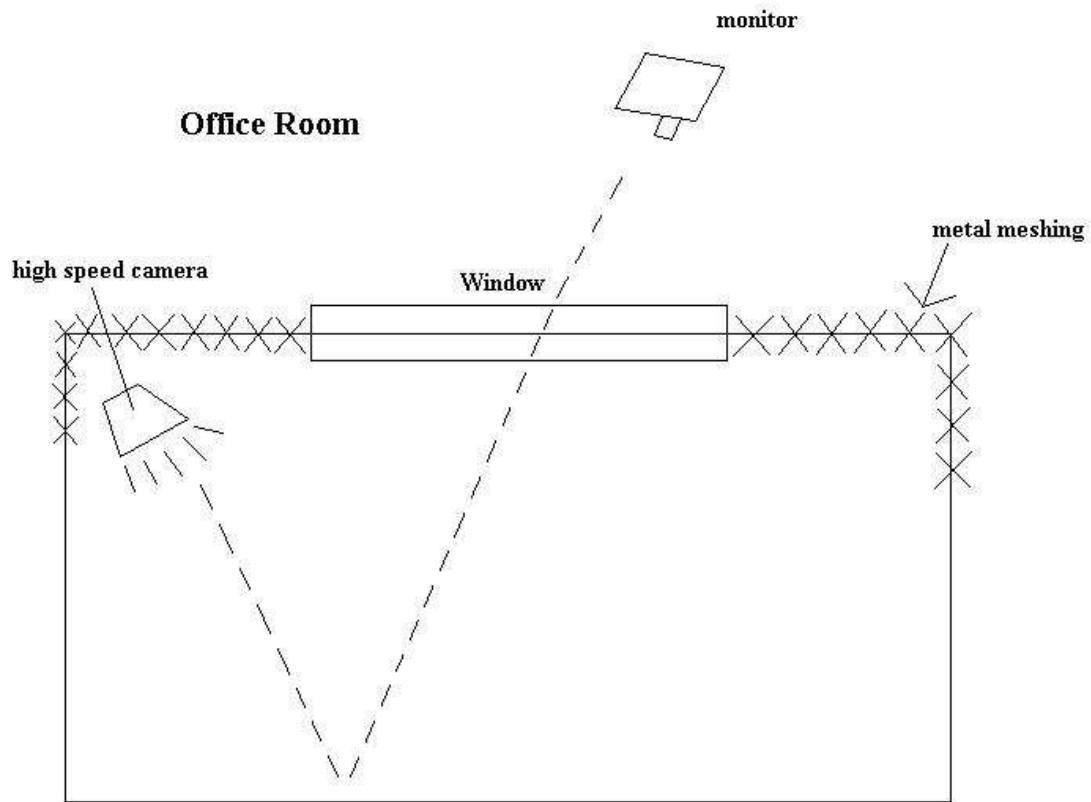
There is another type of side attack, but the technology that this is based on is a CRT monitor: an Optical-Domain Tempest Attack

Computer Screen

1	2	3	4	5	6
7	8	9	10	11	12

← pixels numbered

One pixel lights up at a time if it needs to be on or not. If so, an electron is shot at the pixel to turn it on, but this is done so fast for the entire screen, the human eye only sees the end result. However, place a very high speed camera at the screen and one can obtain the information. This is particularly useful if the monitor is facing a window, and an attacker is not at the same desk or even outside in a different building! Thus, agencies such as NSA force monitors to point away from windows.



Shine a high speed camera at the wall, and when the monitor shines each pixel, it is reflected on the wall and caught by the camera.

LCD monitors do not allow this! Another option that agencies use is a Ferraday Cage, which blocks electromagnetic rays by using metal meshing around the walls and windows of high security places (see the diagram with the office).

CAPTCHA's



These are used to make sure that humans were at the keyboard and not a script (example: Yahoo's free email accounts).

Yahoo needed

-puzzle

-humans can pass with extremely high probability (but later on, this was a problem for illiterate or blind people, foreigners using different alphabets and characters, and even kids).

-computers cannot pass (only passed with a very low probability)

-computer can create the puzzle

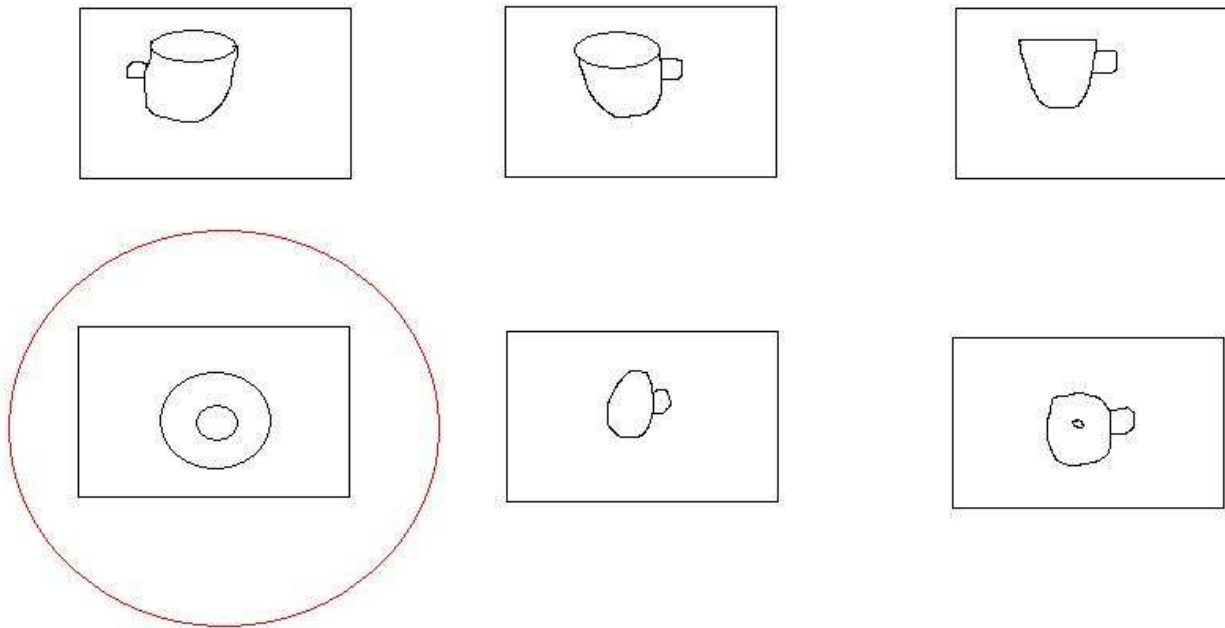
-computer can grade the puzzle (the computer picks a word from the dictionary or a set of characters, then puts them into a image distorter, then checks if the user's input matches the word used to create it).

Attempts to break this lead to:

Fundamental AI problems

-OCR (attacks: better OCR algorithms than used in the past)

-Image recognition (example: two words are picked from the dictionary, use Google image searches on the two words, then picks 5 images of one word and one image from the other word. Pick the odd picture).



Proxy Attack

One can run their own website that everyone wants to visit. Then, require visitors to the website to solve a CAPTCHA. This CAPTCHA is obtained by running a script against Yahoo, then the CAPTCHA is sent from Yahoo to the website, let the visitors solve the CAPTCHAs, and then send their exact response to Yahoo.

The CAPTCHA goal to break this: Humans are the cheapest solution.