Salt

- Hash password with **salt**
- Choose random salt s and compute

  $$y = h(\text{password}, s)$$

  and store (s, y) in the password file
- Note: The salt s is **not** secret
- Easy to verify salted password
- But Eve must re-compute dictionary hashes **for each user**
  - Lots more work for Eve!

Salting

- Have a set of n hash functions
  - Randomly select one function when registering new authentication info
  - Store ID of function with registered info
- Attacker must try all n functions to see if his guess matches any password
- When does this help? When does it not?

Examples

- Vanilla UNIX method
  - Use DES to encipher 0 message with password as key; iterate 25 times
  - Perturb E table in DES in one of 4096 ways
    - 12 bit salt flips entries 0–11 with entries 24–35
    - E Table is per round expansion table
- Alternate methods
  - Use salt as first part of input to hash function

Take-home message --- use n extra bits independent of password to increase work needed by brute-force attack by 2^n

Calculating Password System Strength using Time

Anderson’s formula:

- *P* probability of guessing a password in specified period of time
- *G* number of guesses tested in 1 time unit
- *T* number of time units
- *N* number of possible passwords
- Then P = (TG/N)

Example

- Goal
  - Passwords drawn from a 96-char alphabet
  - Can test 10^4 guesses per second
  - Probability of a success to be 0.5 over a 365 day period
  - What is minimum password length?
- Solution
  - $N \geq TG/P = (365 \times 24 \times 60 \times 60) \times 10^4 / 0.5 = 6.31 \times 10^{11}$
  - Choose s such that $\sum_{j=0}^{s} 96^j \geq N$
  - So $s \geq 6$, meaning passwords must be at least 6 chars long
  - What exactly does that equation mean?
    - Total # passwords using 96 chars, of length s or less

User Selection

- Problem: people pick easy-to-guess passwords
  - Based on account names, user names, computer names, place names
  - Dictionary words (also reversed, odd capitalizations, control characters, “l33t-speak”, conjugations or declensions, Torah/Bible/Koran/… words)
  - Too short, digits only, letters only
  - License plates, acronyms, social security numbers
  - Personal characteristics or foibles (pet names, nicknames, *etc*.)
  - Using the same password in multiple accounts

User Password Education

- Use the first letter of each word in a phrase
  - “My dog’s first name is Rex.” becomes “MdfniR”

- Video – What is your password?
  - https://www.youtube.com/watch?v=opRMrEfAIiI

Reactive Password Checking

- Have a password cracking program running in the background
  - Shut down account of passwords it can crack
  - CPU intensive
  - Shutting down active accounts is likely to annoy someone important eventually.

Proactive password checking

- Don’t let them pick a “bad” password in the first place
- Need to have a fairly fast test of the “goodness” of a password

Bloom Filter

Space efficient probabilistic data structure to tell whether a given element is a member of a set

- No false negatives
  - If an element is not a member, the BF will not report that it is a member
- False positives are possible

Application – determine whether a password given at creation is one of a large list of easily cracked passwords

Bloom Filter

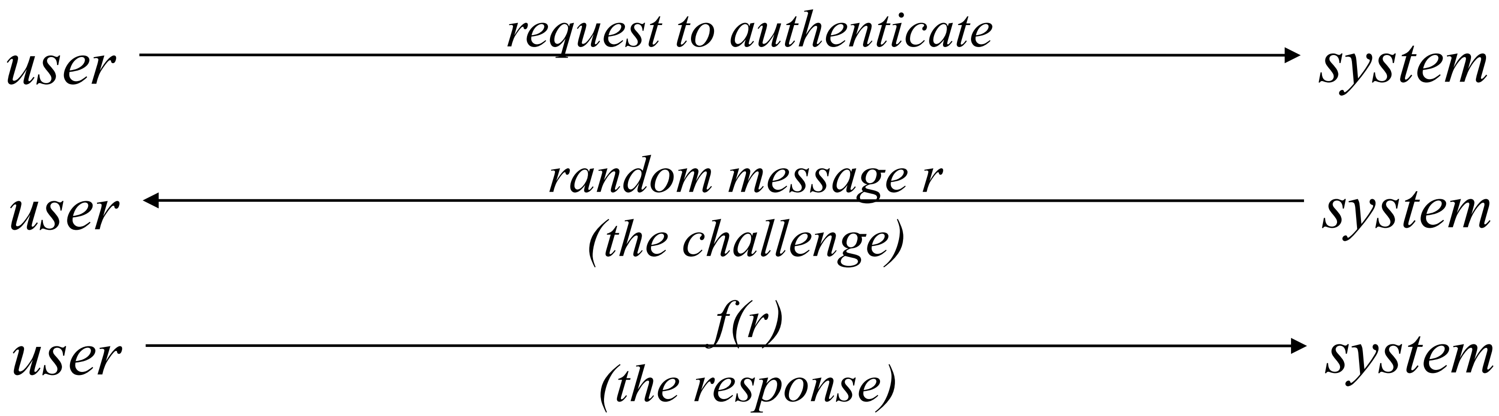
- Create N bit array
- Use k independent hash functions which hash into a space of 0 to N-1
- For each bad password bp,
  - For every hash function h compute h(bp) in [0,N-1] and set the corresponding bit in the hash table
    - Each word marks up to k bits

Bloom Filter

- To check a password
  - Computer every version of the hash, and check the corresponding bits in the array
  - If all bits are 1, then the password is bad
- What about false positive

Challenge-Response

- User and system share a secret function
- User proves knowledge of secret function by answering challenge



user — *request to authenticate* → *system*

user ← *random message r* *(the challenge)* — *system*

user — *f(r)* *(the response)* → *system*

One-Time Passwords

- Password that can be used exactly *once*
  - After use, it is immediately invalidated
- Challenge-response mechanism
  - Challenge is one of a number of authentications; response is password for that particular number
- Problems
  - Synchronization of user, system
  - Generation of good random passwords
  - Password distribution problem

S/Key

- One-time password scheme based on idea of Lamport
- *h,* one-way hash function (MD5 or SHA-1, for example)
- User chooses initial seed *k*
- System calculates:

$$h(k) = k_1,\ h(k_1) = k_2,\ \ldots,\ h(k_{n-1}) = k_n$$

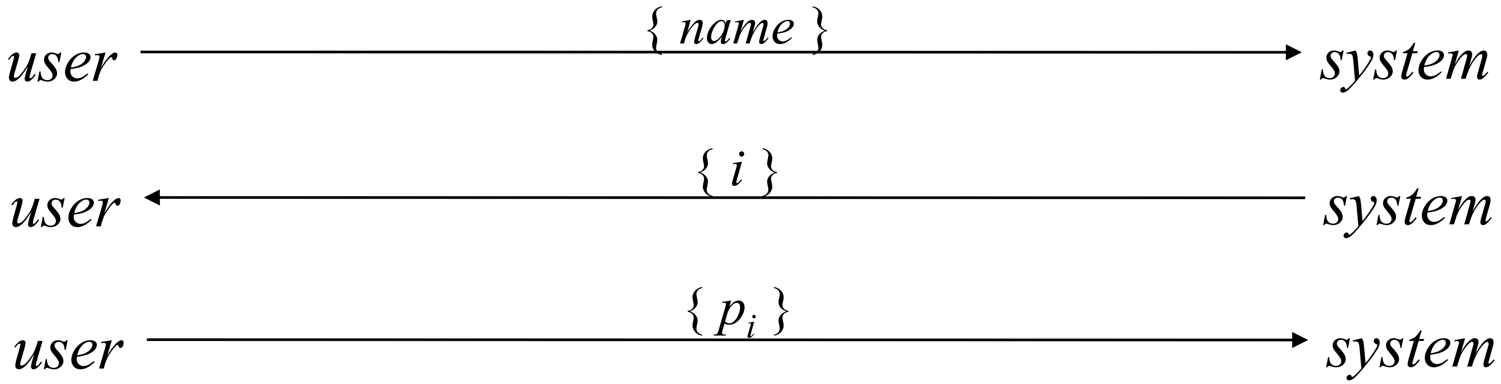
- Passwords are reverse order:

$$p_1 = k_n,\ p_2 = k_{n-1},\ \ldots,\ p_{n-1} = k_2,\ p_n = k_1$$

Central Ideas: Given last pwd p, observer cannot predict p’ s.t. h(p’) = p, i.e., cannot predict next password. Server remembers last pwd p, and when p’ is offered, validates h(p’) = p

S/Key Protocol

System stores maximum number of authentications n, number of next authentication i, last correctly supplied password p_{i-1}.



user { *name* } → *system*

user ← { *i* } *system*

user { p_i } → *system*

System computes $h(p_i) = h(k_{n-i+1}) = k_{n-i+2} = p_{i-1}$. If match with what is stored, system replaces p_{i-1} with p_i and increments i.

Token-based Authentication

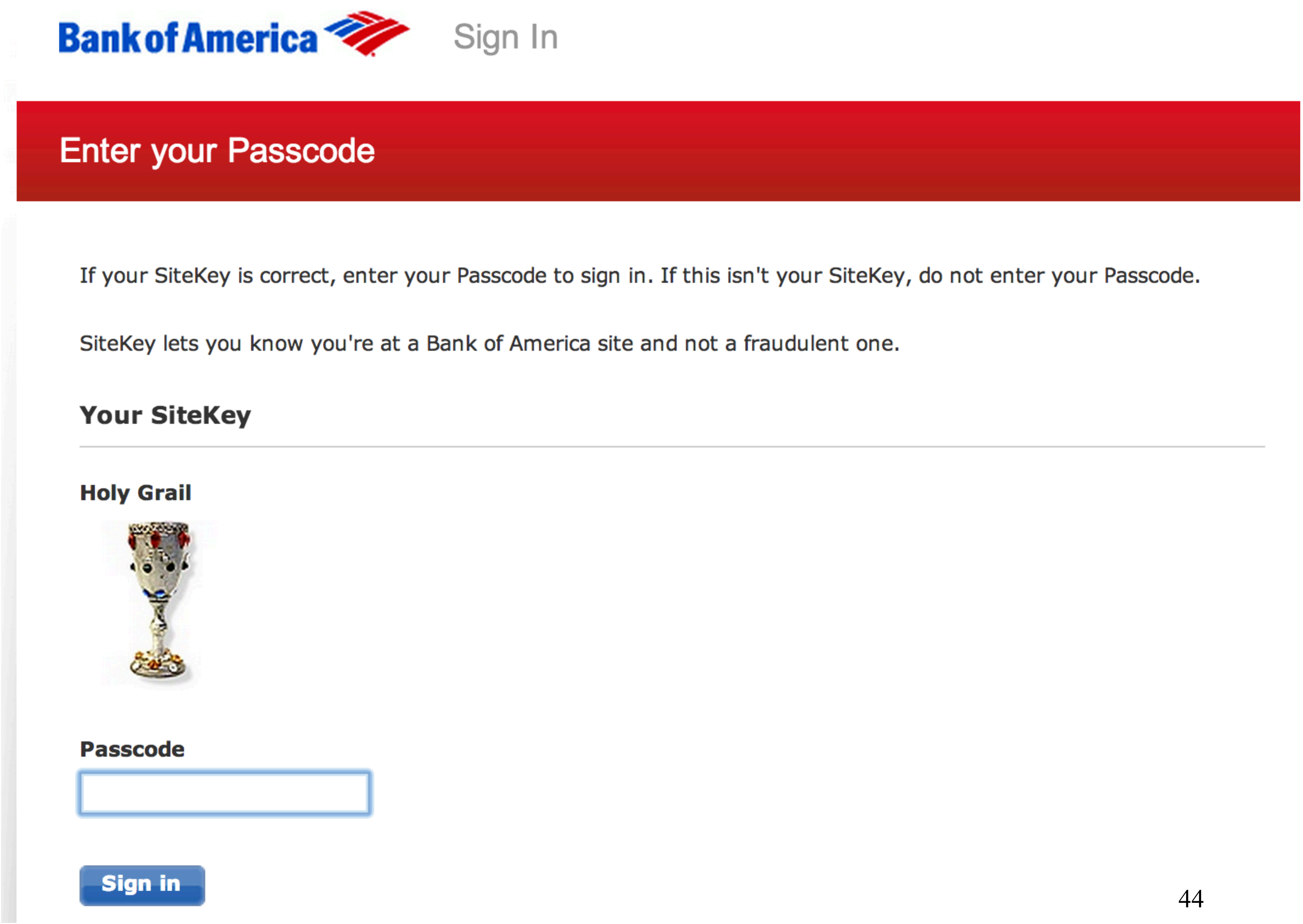
- Something you have
- Memory Cards
  - – No computation on the card
  - – Need special reader to pull data off the card
  - – Need pin to decrypt data off of card
  - – E.g., ATM card or debit card
- By adding PIN (something you know) you get multi-factor authentication

Token-based Authentication

- Smart Card
  - Computation on the card
  - Plug in with USB or wireless communication (credit card)
- Authentication options
  - Static – equivalent to memory card
  - Dynamic password generator – generates a unique password every minute.
  - Challenge response

Two Factor Authentication

- Use two factors, e.g., password + ?



Bank of America Sign In

Enter your Passcode

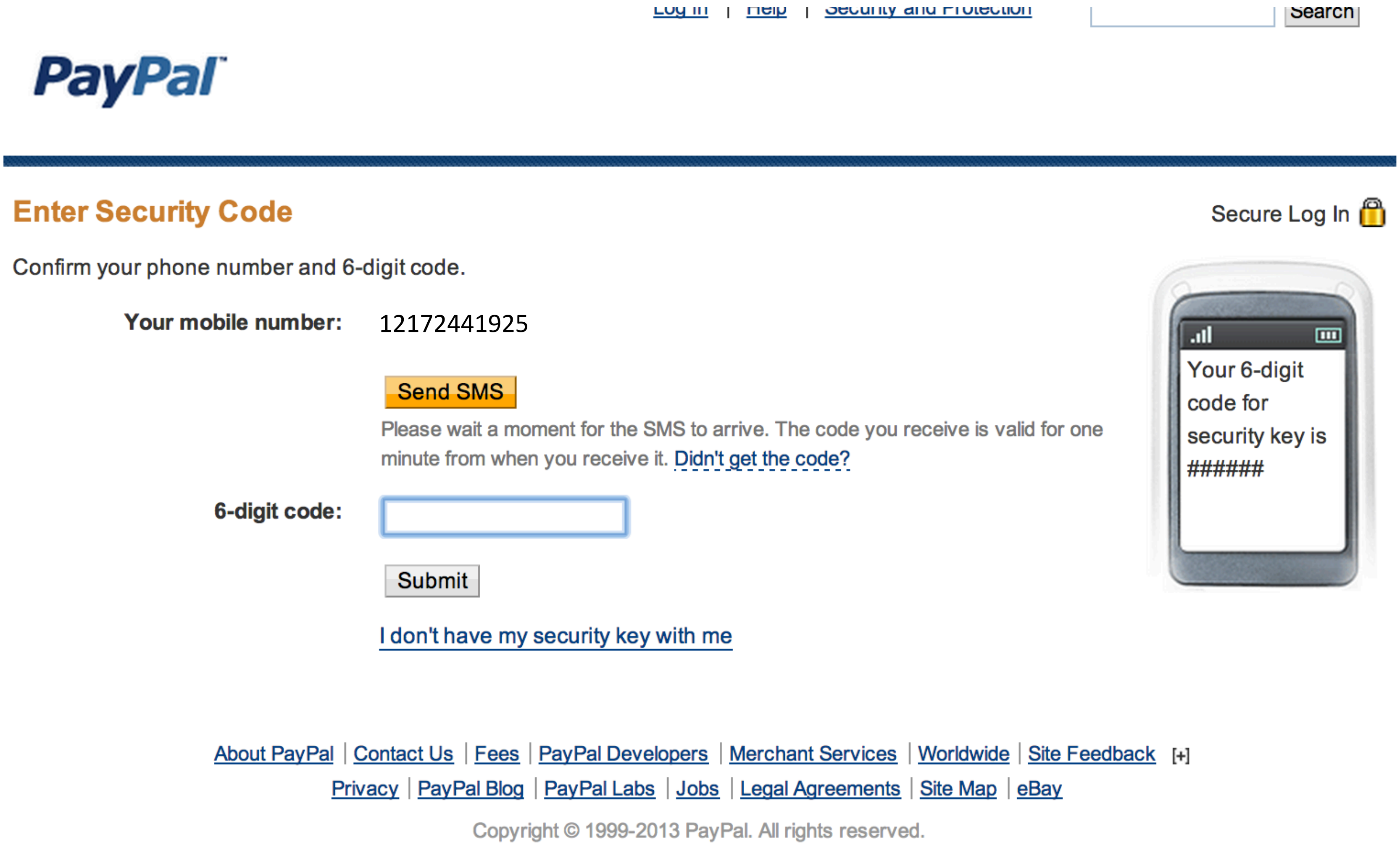
If your SiteKey is correct, enter your Passcode to sign in. If this isn't your SiteKey, do not enter your Passcode.

SiteKey lets you know you're at a Bank of America site and not a fraudulent one.

Your SiteKey

Holy Grail

Passcode



Enter Security Code

Secure Log In

Confirm your phone number and 6-digit code.

Your mobile number: 12172441925

Please wait a moment for the SMS to arrive. The code you receive is valid for one minute from when you receive it. Didn't get the code?

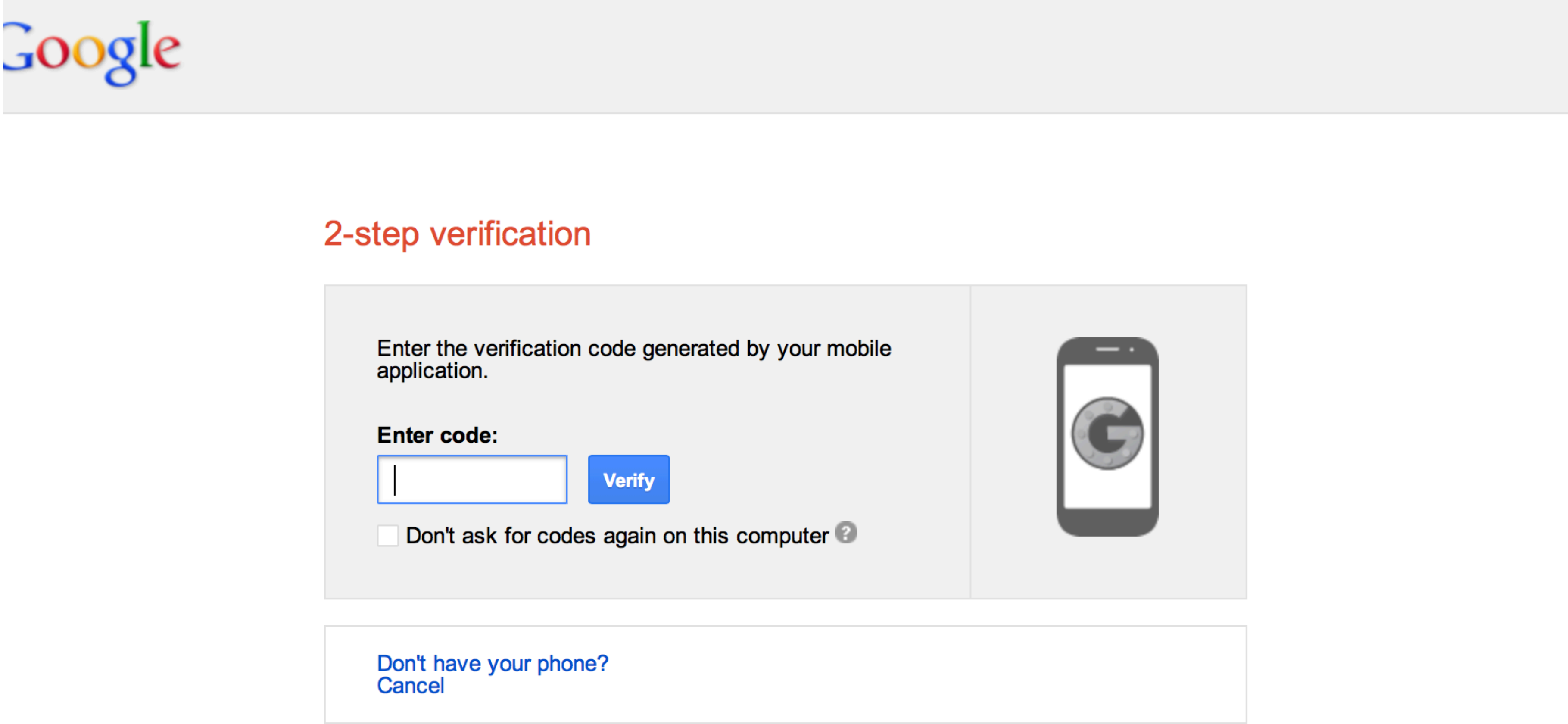
6-digit code:

I don't have my security key with me

Your 6-digit code for security key is ######

About PayPal | Contact Us | Fees | PayPal Developers | Merchant Services | Worldwide | Site Feedback [+]
Privacy | PayPal Blog | PayPal Labs | Jobs | Legal Agreements | Site Map | eBay

Copyright © 1999-2013 PayPal. All rights reserved.



2-step verification

Enter the verification code generated by your mobile application.

Enter code:

□ Don't ask for codes again on this computer

Don't have your phone?
Cancel

Biometrics

- Automated measurement of biological, behavioural features that identify a person
  - Fingerprints: optical or electrical techniques
    - Maps fingerprint into a graph, then compares with database
    - Measurements imprecise, so approximate matching algorithms used
  - Voices: speaker verification or recognition
    - Verification: uses statistical techniques to test hypothesis that speaker is who is claimed (speaker dependent)
    - Recognition: checks content of answers (speaker independent)

Other Characteristics

- Can use several other characteristics
  - Eyes: patterns in irises unique
    - Measure patterns, determine if differences are random; or correlate images using statistical tests
  - Faces: image, or specific characteristics like distance from nose to chin
    - Lighting, view of face, other noise can hinder this
  - Keystroke dynamics: believed to be unique
    - Keystroke intervals, pressure, duration of stroke, where key is struck
    - Statistical tests used

Biometric

- Physical characteristics encoded in a template
  - The C or complement information
- User registers physical information (S)
  - Generally with multiple measurements
- The verification function takes a measurement and tries to line up with template

Biometric Cautions

- These can be fooled!
  - Assumes biometric device accurate *in the environment it is being used in!*
  - Transmission of data to validator is tamperproof, correct (remember *pax vobiscum*)
- Physical characteristics change over time
- Some people may not be able to identify via specific characteristics
  - Albinos and iris scans

Biometric Cautions

- Where are the biometric templates stored?
- What if your biometric template data is stolen?

Key Points

- Passwords are the reality for now
- Multi-factor authentication is must stronger
- Biometrics can help, but not a silver bullet yet