Estimating Password Strength

(fools rush in where angels fear to tread
- this approach is preliminary and may change)

Bill Burr
NIST
william.burr@nist.gov
301-975-2914

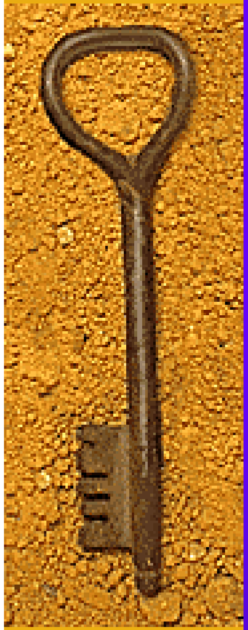
Draft for comment
Subject to change

Disclaimer - Preliminary

- This is a proposal for review and comment.
- It is subject to change, large and small
  - Can easily adjust threshold
  - May also significantly change approach
  - There probably is no right solution

Review the Bidding - Assurance Levels

- Draft GSA/OMB guidance defines 4 assurance levels
  - http://a257.g.akamaitech.net/7/257/2422/14mar20010800/edocket.access.gpo.gov/2003/pdf/03-17634.pdf
- Assurance level needed determined by consequences of authentication error
  - Inconvenience
  - Financial loss
  - Distress
  - Standing or reputation
  - Harm to agency programs or reputation
  - Civil or criminal violations
  - Personal safety



Assurance Levels

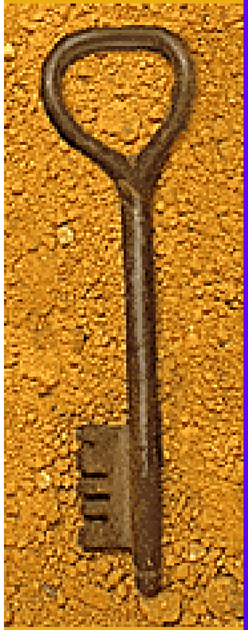
- Level 1 – Minimal Assurance
  - Little or no assurance on the asserted identity
  - Authentication Error might at worst result in
    - minimal inconvenience, financial loss, distress, damage to reputation
    - no risk of harm to agency programs or public interests, release of sensitive information, civil or criminal violations or to personal safety

Assurance Levels

- Level 2 – Low Assurance
  - “On the balance of probabilities” there is confidence in the asserted identity
  - Authentication Error might at worst result in
    - minor inconvenience, financial loss, distress, damage to reputation
    - no risk of harm to agency programs, public interests, release of sensitive information or personal safety
    - civil or criminal violations not normally subject to agency enforcement efforts

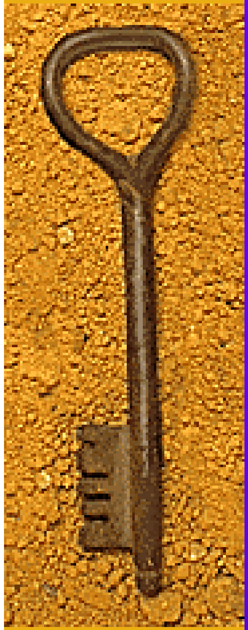
Assurance Levels

- Level 3 – Substantial Assurance
  - Transactions that are “official in nature”
  - High confidence in the asserted identity
  - Authentication error might at worst result in
    - significant inconvenience, financial loss, distress, damage to reputation, harm to agency programs & public interests
    - a significant release of sensitive information
    - civil or criminal violations normally subject to agency enforcement efforts
    - no risk to personal safety



Assurance Levels

- Level 4 – High Assurance
  - Very high confidence in the asserted identity
  - Authentication error might result in
    - considerable inconvenience, financial loss, distress, damage to reputation, harm to agency programs & public interests
    - extensive release of sensitive information
    - considerable risk of an egregious criminal act
    - civil or criminal violations of special importance to agency enforcement efforts
    - risk to personal safety



Passwords and Assurance levels

- Level 1 – PINs
- Level 2 – "Strong" passwords done tolerably well
  - What's a strong password?
- Level 3 – very strong passwords done really well
  - What's very strong and done really well?
- Level 4 – you gotta be kidding

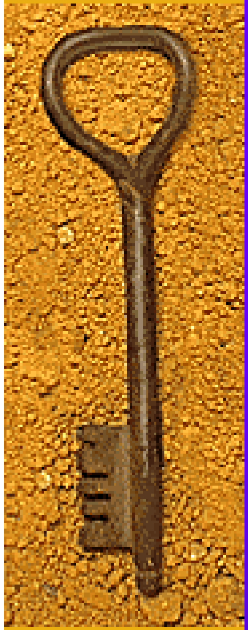
What is a password?

- Password is a secret character string you commit to memory.
  - Secret and memory are the key words here
  - As a practical matter we often do write our passwords down, whatever we are supposed to do with them, and when we do write them down we have to protect them
- A password is really a (generally weak) key
  - People can't remember good keys
- Enrolment and verification phases

Draft for comment
Subject to change

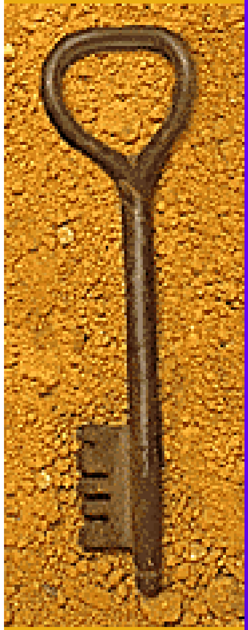
Passwords will ever be with us

- Multifactor authentication
  - Something you are
  - Something you have
  - ***Something you know***
- Problem comes when we depend on passwords as the only factor in remote authentication



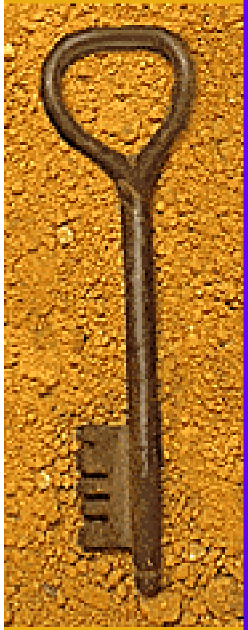
Password Hell

- We all are asked to remember far too many passwords
  - Forced to change them frequently
    - often peremptorily forced to change a password without warning when we try to log on
  - Every system has different rules for passwords
  - Often use them only very infrequently
  - May be given arbitrary, randomly generated passwords
    - who can remember these?



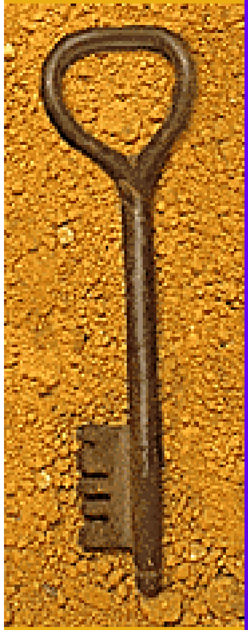
Simplification

- We’re only concerned with on-line authentication to a server, not passwords used, for example to encrypt or lock local files
- Assume that the authentication server is secure and can impose rules to detect or limit attacks



Attacks on Passwords

- In-band
  - Attacker repeatedly tries passwords until he authenticates/gets access
    - guessing, dictionary, or brute force exhaustion
  - Can’t entirely prevent these attacks
    - can ensure they don’t succeed very often
- Out of band – everything else
  - Eavesdropper
  - Man-in-the-middle
  - Shoulder surfing
  - Social engineering



Password Strength

- ♦ Define password strength in terms of probability of a determined attacker discovering a selected user’s password by an in-band attack
  - – Strength is then a function of both the “entropy” of the password and the way unsuccessful trials are limited
  - – Many strategies for limiting unsuccessful trials
    - 3 strikes and you’re out
    - hang up after an unsuccessful trial
    - some total number of unsuccessful trials and lock account
    - change passwords periodically
    - notify user of successful and unsuccessful login attempts
  - – Trade-offs with help desk costs

Strong Password Definition

- The probability of an attacker with no *a priori* knowledge of the password finding a given user’s password by an in-band attack shall not exceed one in 2^{16} (1/65,563) over the life of the password
  - The more entropy required in the password, the more trials the system can allow
  - Note that there is not necessarily any particular time limit

Estimating Password Entropy

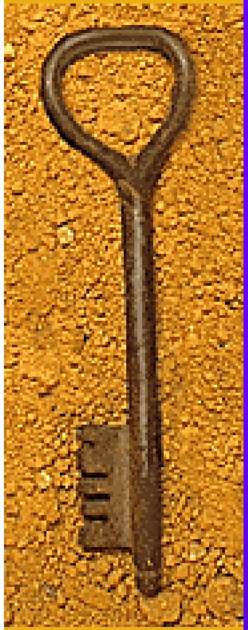
- Entropy of a password is the uncertainty an attacker has in his knowledge of the password, that is how hard it is to guess it.

$$H(X) := -\sum_{x} P(X = x)\log_2 P(X = x)$$

- Easy to compute entropy of random passwords
- We typically state entropy in bits. A random 32-bit number has 2^{32} values and 32-bits of entropy
- A password of length l selected at random from the keyboard set of 94 printable (nonblank) characters has 94^{l} values and about $6.55 \times l$ bits of entropy.

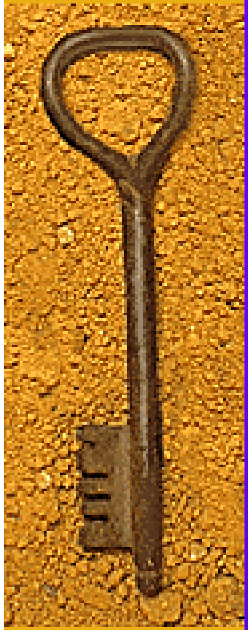
User Selected Passwords

- ♦ People have a hard time remembering random passwords
  - – So we may let them pick their own
- ♦ People pick bad passwords
  - – Passwords that are easy to remember are often easy to guess
    - use common words
    - frequency distributions of characters
    - phone number, street address, SSN, dog’s name, birthday…
  - – Sophisticated attacker takes advantage of this with (possibly large) dictionaries of common passwords



Entropy of User Chosen Pswd

- No really rigorous way to estimate
- Propose starting from Shannon's estimate of entropy in English text
  - C. E. Shannon, "Prediction and Entropy of Printed English" *Bell System Technical Journal*, v.30, n.1, 1951, pp. 50-64
    - One of the most widely referenced papers in computing
    - Seems to be relatively little progress beyond Shannon.

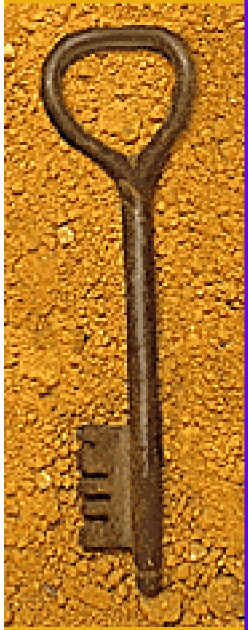


Shannon's estimate of entropy

- Shannon used 26 English letters plus space
  - Left to their own devices user will choose only lower case letters.
- Shannon's method involves knowing the *i*-1 first letters of a string of English text; how well can we guess the *i*th letter?
- Entropy per character decreases for longer strings
  - 1 character 4.7 bits/character
  - ≤ 8 characters 2.3 bits per character
  - order of 1 bit/char for very long strings

Use Shannon as Lower Bound

- Users are supposed to pick passwords that don't look like ordinary English
  - But, of course, they want to remember them
- Attacker won't have a perfect dictionary or learn much by each unsuccessful trial



Estimate Entropy vs PWD length

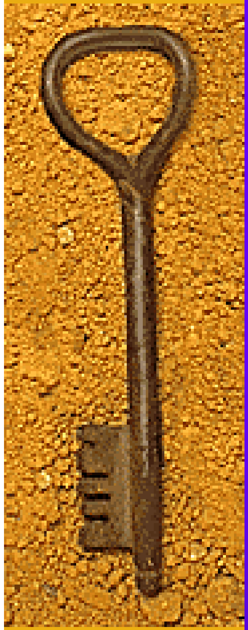
Password Length	Entropy Bits	Password Length	Entropy Bits
1	4	10	21
2	6	12	24
3	8	14	27
4	10	16	30
5	12	18	33
6	14	20	36
7	16	30	46
8	18		

Estimate Entropy vs PWD length

- 1- 10 character passwords consistent with curves in Fig. 4 of paper
- 10 – 20 character passwords assume that entropy grows at 1.5 bits of entropy per character
- Over 20 character passwords assume that entropy grows at 1 bit per character

Password Rules

- We can increase the “effective” entropy of user chosen passwords by imposing rules on them that make the passwords less like ordinary English (or French or German or..) words. For example:
  - Passwords must contain at least one upper case letter, one number and one special character
  - Passwords must not contain any strings from a dictionary of common strings



Password Rules

- Rules reduce the total number of possible passwords, which is bad
  - But they can eliminate a lot of commonly used (easily guessed) passwords and make users select passwords they just wouldn't otherwise choose, stretching the effective space
- If we go overboard rules make it hard to remember the passwords
  - We let users pick their passwords in the first place so they can remember them

Proposal

- Award an entropy bonus of up to 6 bits for password composition rules
- Award an entropy bonus of up to 6 bits for a dictionary test
  - Bonus declines for long “pass-phrases”
    - Have to contain common words or you can’t remember them
    - No bonus for over 20 char.
- Rules don’t work as well in combination for very short passwords

How do rules affect entropy?

- Assign entropy “bonus” for composition rules
- Consider
  - Passwords must contain at least one upper case letter, one lower case letter, one number and one special character
    - we’ll often get just one of each, however long the password, at the the beginning or the end of the password
      - Redskins1!
      - Algernon8*
      - A!1lgernon
    - some combinations will be common
      - 1! 2@ 3#
  - Probably some benefit even for very long passwords

Estimate Entropy vs PWD length with Composition Rule

Password Length	Entropy Bits	Password Length	Entropy Bits
1	-	10	27
2	-	12	30
3	-	14	33
4	15	16	36
5	18	18	39
6	20	20	42
7	22	30	52
8	24		

Dictionary Test

- Attacker will use a dictionary first
- Can be quite extensive
- Test passwords against a dictionary
  - Even a big dictionary doesn’t occupy much of the total password space and half the passwords is one bit of entropy
- Dictionary less effective for long passwords
  - Need to allow phrases of words if long passwords are to be practical
  - Assume dictionary test doesn’t help for 20 char or longer passwords

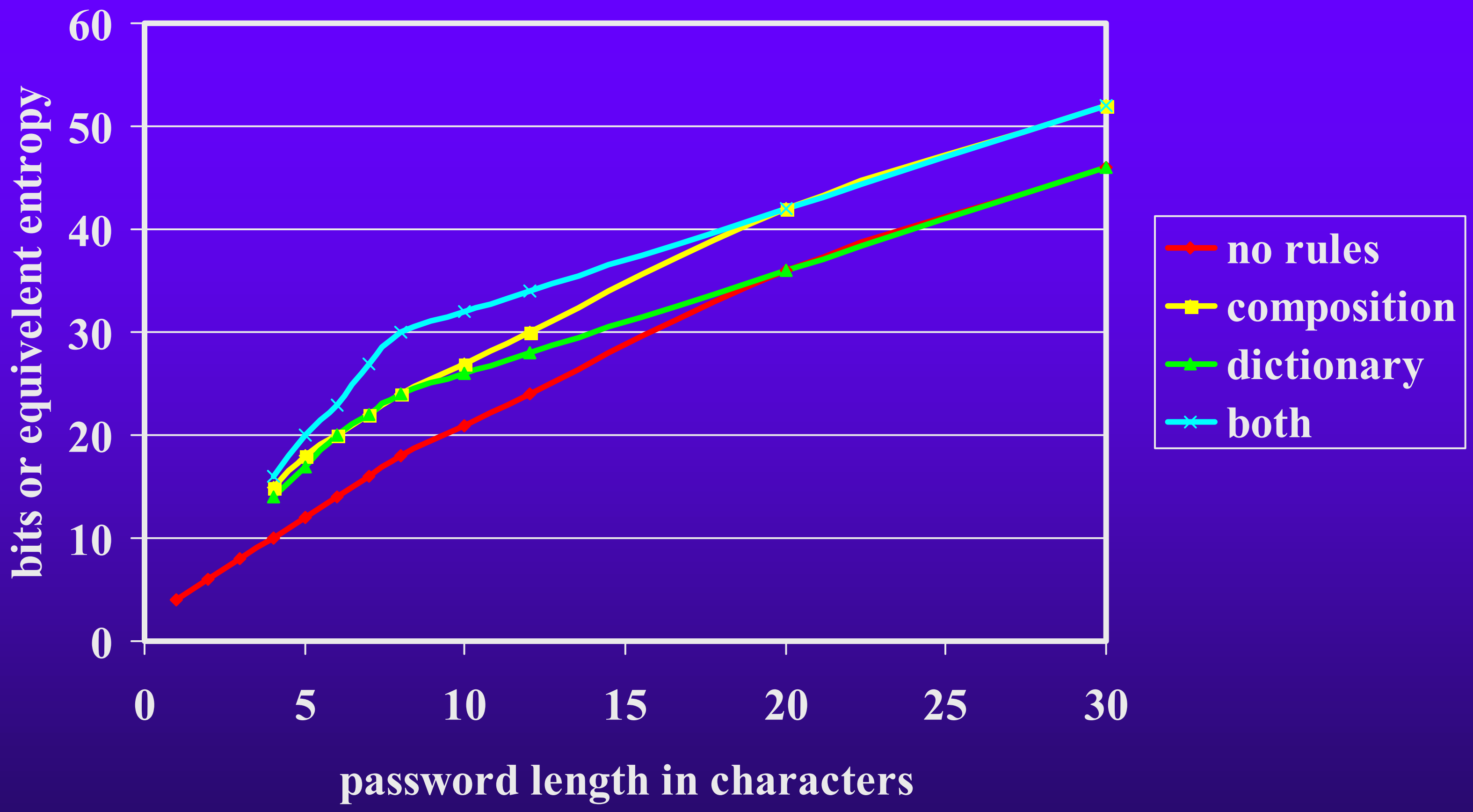
Estimate Entropy vs PWD length with Dictionary Test

Password Length	Entropy Bits	Password Length	Entropy Bits
1	-	10	26
2	-	12	28
3	-	14	30
4	14	16	32
5	17	18	34
6	20	20	36
7	22	30	50
8	24		

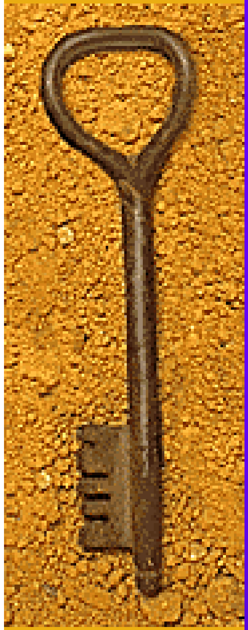
Estimate Entropy vs PWD length with Rule & Dictionary

Password Length	Entropy Bits	Password Length	Entropy Bits
1	-	10	32
2	-	12	34
3	-	14	36
4	16	16	38
5	20	18	40
6	23	20	42
7	27	30	52
8	30		

Entropy estimate versus length



Draft for comment
Subject to change

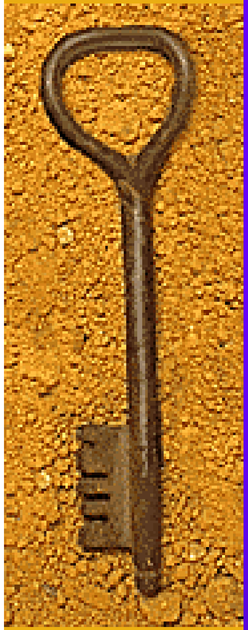


A Measurement Experiment?

- No time to affect the first round of guidance; but
- Can we find a source of lots of actual user selected passwords?
  - On the order of at least hundreds of thousands
  - With different rules
  - Probably could live with password hashes
    - Use collision frequencies
    - Couldn’t use hash(password||username||salt)

Proposed Thresholds

- Level 1, minimal assurance
  - Probability of a successful in-band password attack less than .0005 (one in 2^{11})
- Level 2, low assurance
  - Probability of a successful in-band password attack less than .000015 (one in 2^{16}).
- Level 3 , substantial assurance
  - Probability of a successful in-band password attack less than .000001 (one in 2^{20}).



Level 1 – Minimal Assurance

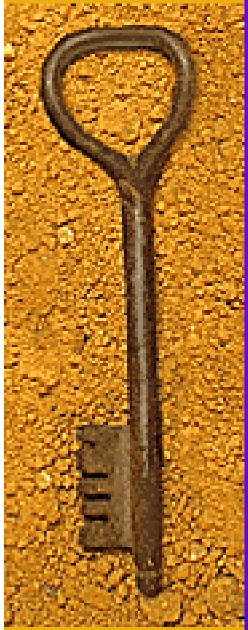
- ♦ Basically for PINs, or passwords sent without encryption
  - – Not expected to resist eavesdroppers
- ♦ No more than 1 in 2^{11} (2048) chance of in-band attack succeeding over life of password

Level 2 – Low Assurance

- Useful for routine e-commerce and e-gov transactions
- Must resist eavesdroppers
  - resist off-line analysis of authentication protocol run
- Resist replays
- No more than 1 in 2^{16} (65,536) chance of in-band attack succeeding over life of password
- Not required to defeat man-in-the-middle or verifier impersonation attacks

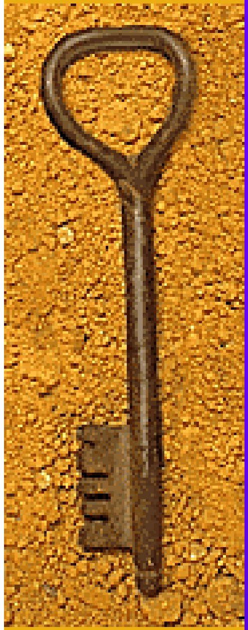
Level 3 – Substantial Assurance

- Useful for e-commerce and e-gov transactions of substantial value
- Must resist eavesdroppers
  - resist off-line analysis of authentication protocol run
- Resist replays
- Resist man-in-the-middle or verifier impersonation attacks
- No more than 1 in 2^{20} (1,000,000) chance of in-band attack succeeding over life of password



Example – Level 2

- 6 characters, randomly selected
  - 94^6 possible values (about 6.9×10^{11})
  - That’s about 39 bits of entropy
- Authentication system must limit the total number of unsuccessful authentication attempts to $94^6/2^{16} \approx 10{,}000{,}000$



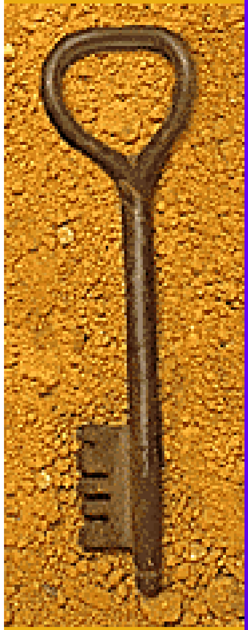
Example – Level 2

- 8 characters, user selected, no composition rule or dictionary check
  - estimate 18-bits of entropy which is about 250,000
- Authentication system must limit the total number of unsuccessful authentication attempts to $2^{18}/2^{16} = 4$ trials

Draft for comment
Subject to change

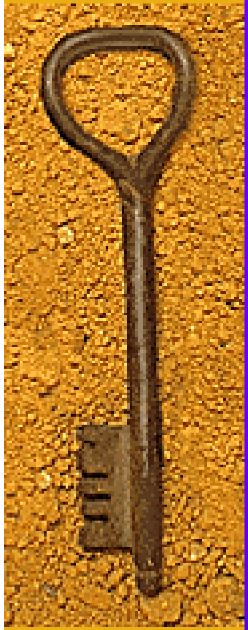
Example - Level 2

- 8 characters, user selected, with composition rule and dictionary check
  - estimate 30-bits of entropy which is about 10^9
- Authentication system must limit the total number of unsuccessful authentication attempts to $2^{30}/2^{16} = 2^{15} \approx 16{,}000$ trials



Example – Level 2

- 6 characters, user selected, with composition rule and dictionary check
  - estimate 26-bits of entropy
- Authentication system must limit the total number of unsuccessful authentication attempts to $2^{26}/2^{16} = 1024$ trials

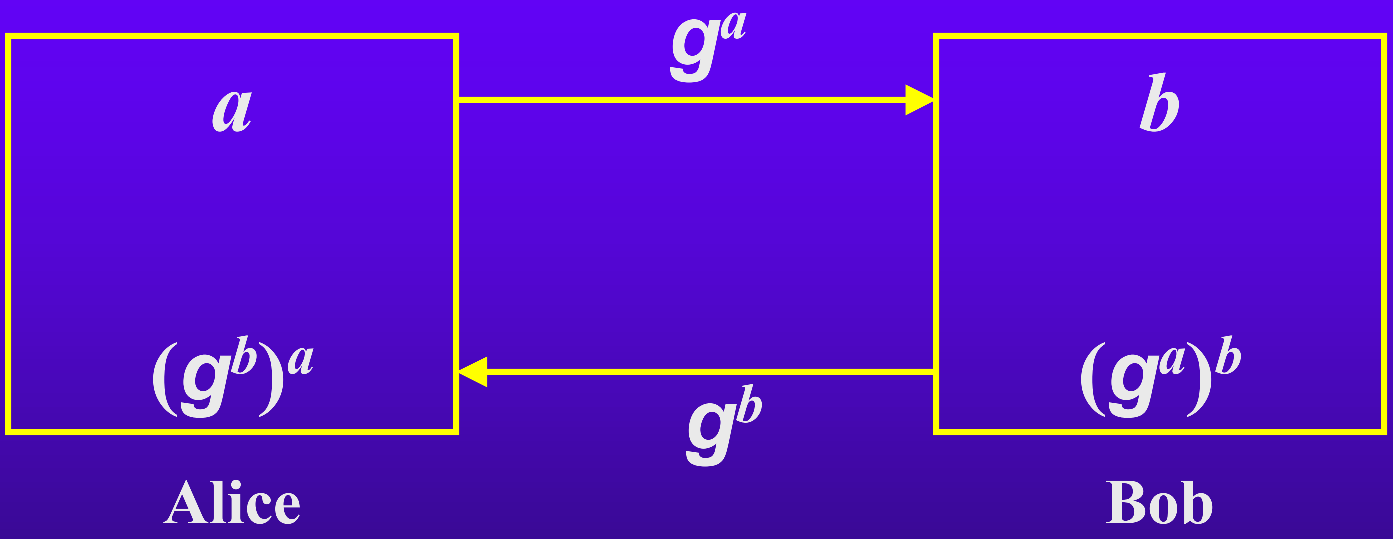


Zero Knowledge Password Auth.

- Verifier and claimant share a password
- If attacker fools claimant into an authentication protocol run, he gains no knowledge of password
- Verifier and claimant wind up with a strong shared secret, which can be used in any protocol that requires a symmetric key
- Eavesdropper learns nothing about password or strong shared secret

Diffe-Hellman key exchange

Pick a generator g of a large finite group G.
a and b are large random numbers.



a

g^a

b

$(g^b)^a$

g^b

$(g^a)^b$

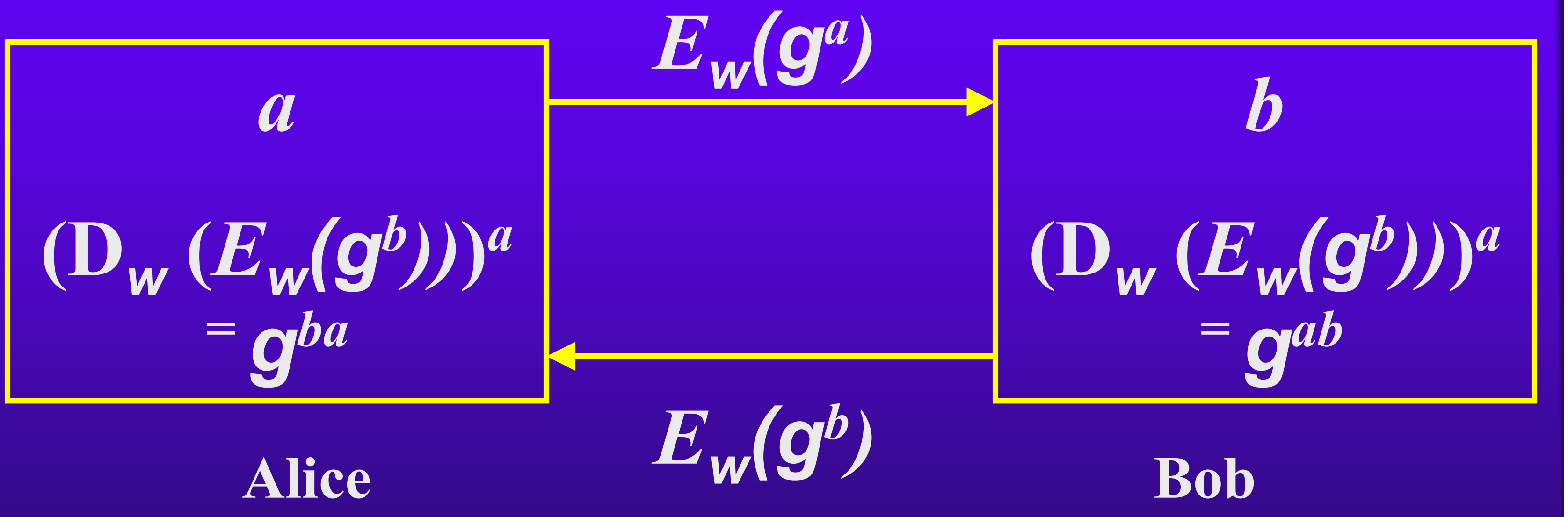
Alice

Bob

Alice and Bob now share a common secret g^{ab}.
An eavesdropper must solve discrete log problem to learn g^{ab}.

EKE exchange

Let p be Alice's password, w=hash(p), Bob knows w, and $E_w(x)$ be x encrypted under key w



$E_w(g^a)$

a

$(D_w(E_w(g^b)))^a = g^{ba}$

Alice

b

$(D_w(E_w(g^b)))^a = g^{ab}$

Bob

$E_w(g^b)$

Alice and Bob now share a common cryptographic strength secret g^{ab}.

Token Type by Level

Allowed Token Types	1	2	3	4
Hard crypto token	√	√	√	√
Soft crypto token	√	√	√	
password with zero knowledge protocol	√	√	√	
Strong password with eavesdropper protection	√	√		
PIN	√			

Required Protections by Level

Protection Against	1	2	3	4
Eavesdropper		√	√	√
Replay	√	√	√	√
On-line guessing	√	√	√	√
Verifier Impersonation			√	√
Man-in-the-middle			√	√
Session Hijacking			√	√

Auth. Protocol Type by Level

Allowed Protocol Types	1	2	3	4
Private key PoP	√	√	√	√
Symmetric key PoP	√	√	√	√
Zero knowledge password	√	√	√	
Tunneled password	√	√		
Challenge-reply password	√			

Required Protocol Properties by Level

Required properties	1	2	3	4
Shared secrets not revealed to 3rd parties		√	√	√
Session Data transfer authenticated			√	√