I&A

CSH6 Chapter 28
"Identification & Authentication"

Ravi Sandhu, Jennifer Hadley,
Steven Lovaas, & Nicholas Takacs

1

Topics

- ➢Introduction
- ➢Four Principles of Authentication
- ➢Password-Based Authentication
- ➢Token-Based Authentication
- ➢Biometric Authentication
- ➢Cross-Domain Authentication
- ➢Relative Costs of Authentication Technologies
- ➢Concluding Remarks

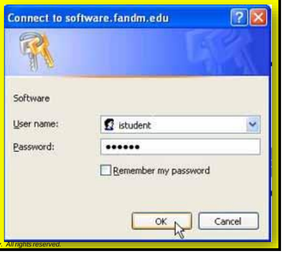
Introduction (1)

- ➢ Identification
  - ❑Assigning specific code to user or device
  - ❑*User identifier,* aka *user ID,* aka *userID*
- ➢ Authentication
  - ❑Binding or linking specific human being (or device such as computer) to specific ID
- ➢ Authorization
  - ❑Granting specific permissions for particular actions on particular data; e.g.,
    - ✓Read, Write, Append, Lock, Execute
    - ✓Create new file, save old file, rename file
    - ✓Define check amount, payee, OK payment

3

Introduction (2)

- ➢Focus of chapter is person-to-computer authentication
- ➢Also need computer-to-person authentication
  - ❑Prevent spoofing of services on network
  - ❑Phishing e-mails send victims to fake Web sites that look legitimate
- ➢Computer-to-computer authentication
  - ❑Essential to safeguard critical transactions
  - ❑E.g., interbank transfers, B2B e-commerce



Connect to software.fandm.edu

Software

User name: istudent

Password: ••••••

□ Remember my password

4

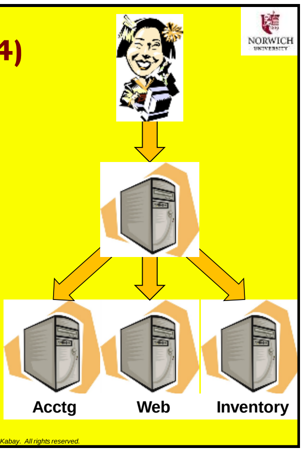
Introduction (3)

- ➢ Historically, mainframes authenticated users
  - ❑Within single enterprise
  - ❑Allowed centralized, controlled assignment of user IDs
- ➢ Identifiers have never necessarily been unique
  - ❑Not usual to have 1:1 relation between userID and person: usually at least N:1
  - ❑But may have several people who use one userID
    - ✓May have controls to prevent simultaneous multiple uses of same userID
  - ❑And one person may have several userIDs
    - ✓May become difficult to maintain authentication methods for multitude of IDs

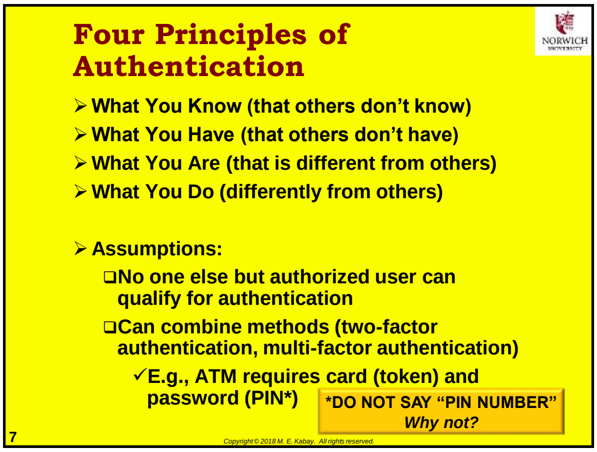
5

Introduction (4)

- ➢Single sign-on
  - ❑Goal of today's I&A research
  - ❑Arrange to identify and authenticate once for entire network
- ➢I&A also used in physical security
  - ❑See *CSH6* Chapter 23



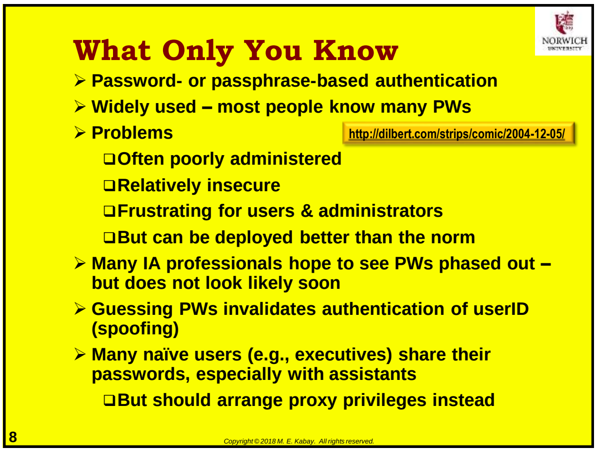
6



Four Principles of Authentication

- ➢ What You Know (that others don't know)
- ➢ What You Have (that others don't have)
- ➢ What You Are (that is different from others)
- ➢ What You Do (differently from others)
- ➢ Assumptions:
  - ❑ No one else but authorized user can qualify for authentication
  - ❑ Can combine methods (two-factor authentication, multi-factor authentication)
    - ✓ E.g., ATM requires card (token) and password (PIN*)

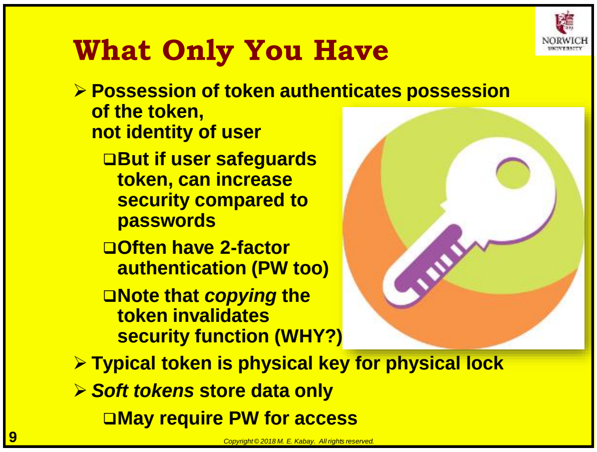
*DO NOT SAY "PIN NUMBER"
Why not?



What Only You Know

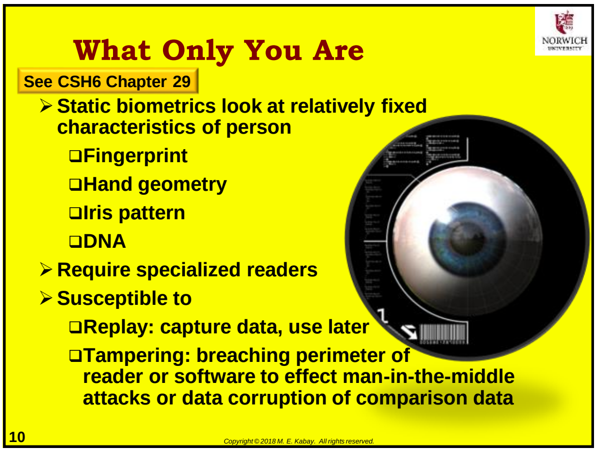
- ➢ Password- or passphrase-based authentication
- ➢ Widely used – most people know many PWs
- ➢ Problems
  - ❑ Often poorly administered
  - ❑ Relatively insecure
  - ❑ Frustrating for users & administrators
  - ❑ But can be deployed better than the norm
- ➢ Many IA professionals hope to see PWs phased out – but does not look likely soon
- ➢ Guessing PWs invalidates authentication of userID (spoofing)
- ➢ Many naïve users (e.g., executives) share their passwords, especially with assistants
  - ❑ But should arrange proxy privileges instead

http://dilbert.com/strips/comic/2004-12-05/



What Only You Have

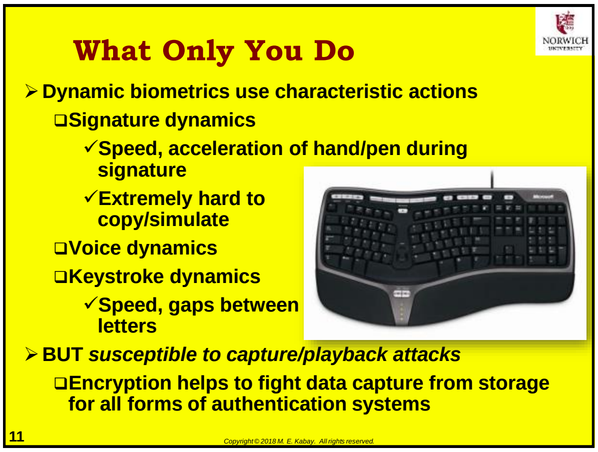
- ➢ Possession of token authenticates possession of the token, not identity of user
  - ❑ But if user safeguards token, can increase security compared to passwords
  - ❑ Often have 2-factor authentication (PW too)
  - ❑ Note that *copying* the token invalidates security function (WHY?)
- ➢ Typical token is physical key for physical lock
- ➢ *Soft tokens* store data only
  - ❑ May require PW for access



What Only You Are

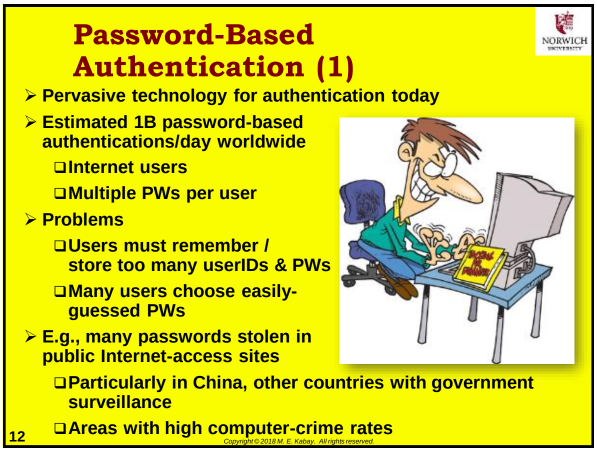
See CSH6 Chapter 29

- ➢ Static biometrics look at relatively fixed characteristics of person
  - ❑ Fingerprint
  - ❑ Hand geometry
  - ❑ Iris pattern
  - ❑ DNA
- ➢ Require specialized readers
- ➢ Susceptible to
  - ❑ Replay: capture data, use later
  - ❑ Tampering: breaching perimeter of reader or software to effect man-in-the-middle attacks or data corruption of comparison data



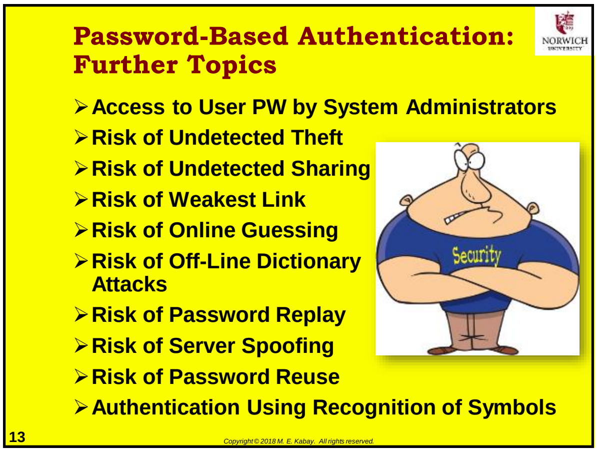
What Only You Do

- ➢ Dynamic biometrics use characteristic actions
  - ❑ Signature dynamics
    - ✓ Speed, acceleration of hand/pen during signature
    - ✓ Extremely hard to copy/simulate
  - ❑ Voice dynamics
  - ❑ Keystroke dynamics
    - ✓ Speed, gaps between letters
- ➢ BUT *susceptible to capture/playback attacks*
  - ❑ Encryption helps to fight data capture from storage for all forms of authentication systems



Password-Based Authentication (1)

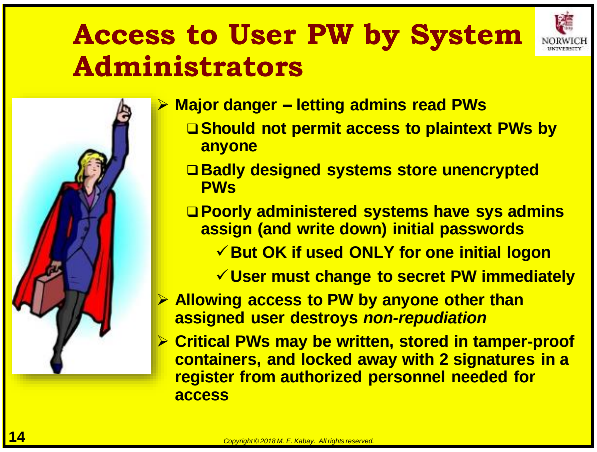
- ➢ Pervasive technology for authentication today
- ➢ Estimated 1B password-based authentications/day worldwide
  - ❑ Internet users
  - ❑ Multiple PWs per user
- ➢ Problems
  - ❑ Users must remember / store too many userIDs & PWs
  - ❑ Many users choose easily-guessed PWs
- ➢ E.g., many passwords stolen in public Internet-access sites
  - ❑ Particularly in China, other countries with government surveillance
  - ❑ Areas with high computer-crime rates



Password-Based Authentication: Further Topics

- Access to User PW by System Administrators
- Risk of Undetected Theft
- Risk of Undetected Sharing
- Risk of Weakest Link
- Risk of Online Guessing
- Risk of Off-Line Dictionary Attacks
- Risk of Password Replay
- Risk of Server Spoofing
- Risk of Password Reuse
- Authentication Using Recognition of Symbols

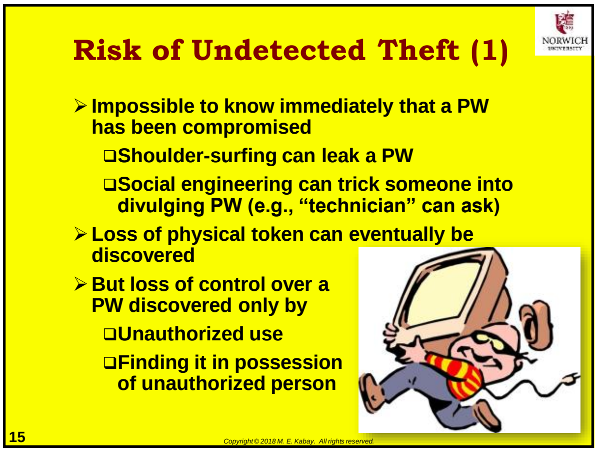
13



Access to User PW by System Administrators

- Major danger – letting admins read PWs
  - ❑ Should not permit access to plaintext PWs by anyone
  - ❑ Badly designed systems store unencrypted PWs
  - ❑ Poorly administered systems have sys admins assign (and write down) initial passwords
    - ✓ But OK if used ONLY for one initial logon
    - ✓ User must change to secret PW immediately
- Allowing access to PW by anyone other than assigned user destroys *non-repudiation*
- Critical PWs may be written, stored in tamper-proof containers, and locked away with 2 signatures in a register from authorized personnel needed for access

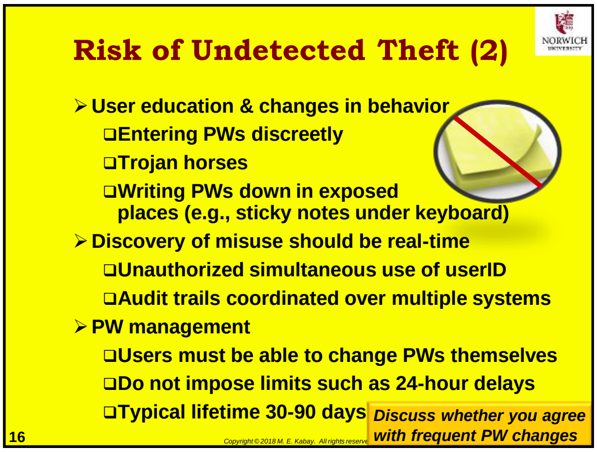
14



Risk of Undetected Theft (1)

- Impossible to know immediately that a PW has been compromised
  - ❑ Shoulder-surfing can leak a PW
  - ❑ Social engineering can trick someone into divulging PW (e.g., "technician" can ask)
- Loss of physical token can eventually be discovered
- But loss of control over a PW discovered only by
  - ❑ Unauthorized use
  - ❑ Finding it in possession of unauthorized person

15

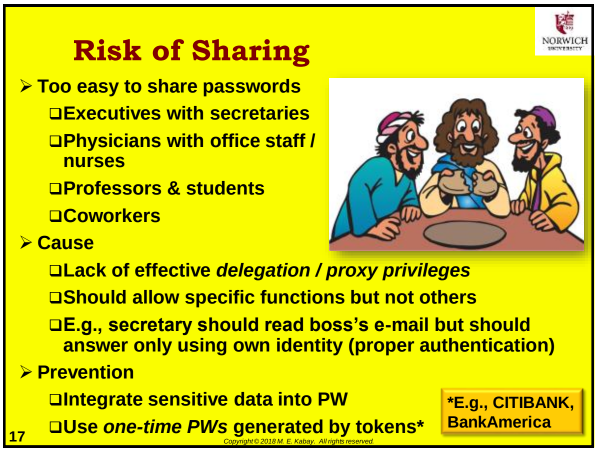


Risk of Undetected Theft (2)

- User education & changes in behavior
  - ❑ Entering PWs discreetly
  - ❑ Trojan horses
  - ❑ Writing PWs down in exposed places (e.g., sticky notes under keyboard)
- Discovery of misuse should be real-time
  - ❑ Unauthorized simultaneous use of userID
  - ❑ Audit trails coordinated over multiple systems
- PW management
  - ❑ Users must be able to change PWs themselves
  - ❑ Do not impose limits such as 24-hour delays
  - ❑ Typical lifetime 30-90 days

Discuss whether you agree with frequent PW changes

16

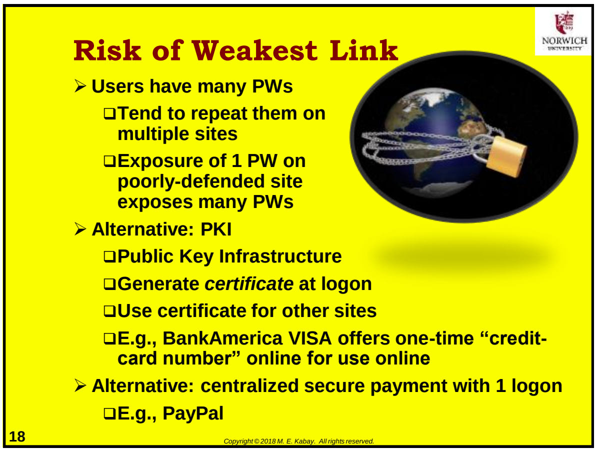


Risk of Sharing

- Too easy to share passwords
  - ❑ Executives with secretaries
  - ❑ Physicians with office staff / nurses
  - ❑ Professors & students
  - ❑ Coworkers
- Cause
  - ❑ Lack of effective *delegation / proxy privileges*
  - ❑ Should allow specific functions but not others
  - ❑ E.g., secretary should read boss's e-mail but should answer only using own identity (proper authentication)
- Prevention
  - ❑ Integrate sensitive data into PW
  - ❑ Use *one-time PWs* generated by tokens*

*E.g., CITIBANK, BankAmerica

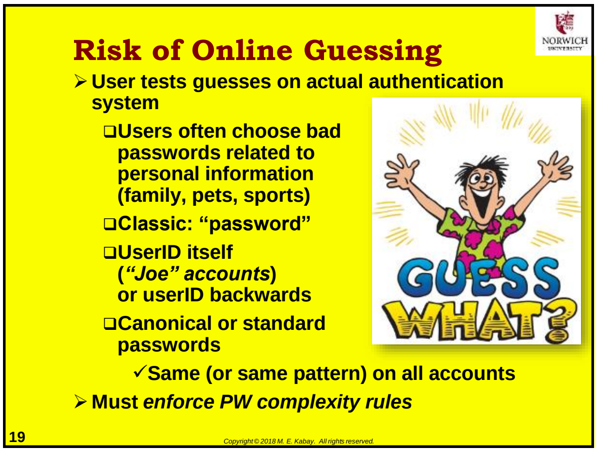
17



Risk of Weakest Link

- Users have many PWs
  - ❑ Tend to repeat them on multiple sites
  - ❑ Exposure of 1 PW on poorly-defended site exposes many PWs
- Alternative: PKI
  - ❑ Public Key Infrastructure
  - ❑ Generate *certificate* at logon
  - ❑ Use certificate for other sites
  - ❑ E.g., BankAmerica VISA offers one-time "credit-card number" online for use online
- Alternative: centralized secure payment with 1 logon
  - ❑ E.g., PayPal

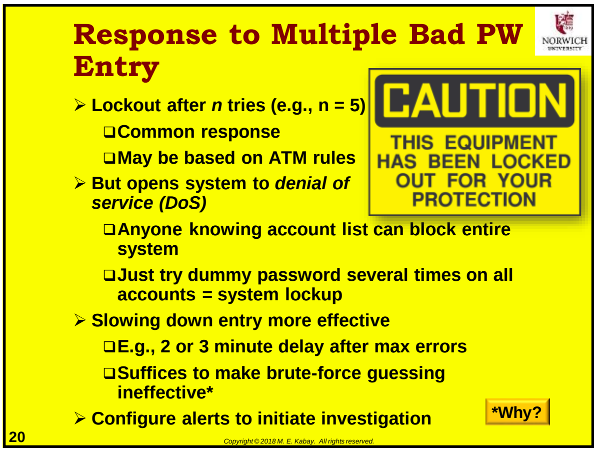
18



Risk of Online Guessing

- User tests guesses on actual authentication system
  - ❑Users often choose bad passwords related to personal information (family, pets, sports)
  - ❑Classic: "password"
  - ❑UserID itself (*"Joe" accounts*) or userID backwards
  - ❑Canonical or standard passwords
    - ✓Same (or same pattern) on all accounts
- Must *enforce PW complexity rules*

19

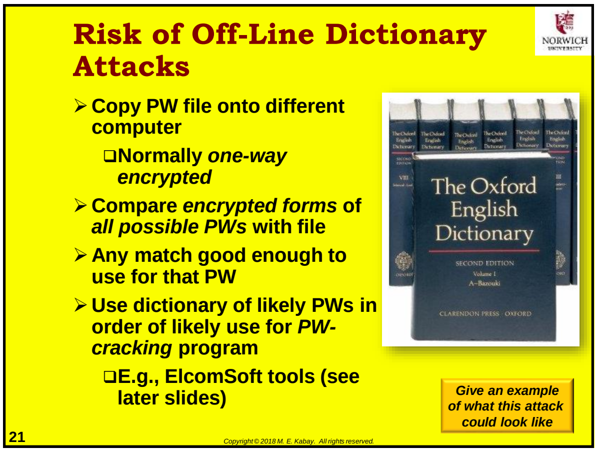


Response to Multiple Bad PW Entry

- Lockout after *n* tries (e.g., n = 5)
  - ❑Common response
  - ❑May be based on ATM rules
- But opens system to *denial of service (DoS)*
  - ❑Anyone knowing account list can block entire system
  - ❑Just try dummy password several times on all accounts = system lockup
- Slowing down entry more effective
  - ❑E.g., 2 or 3 minute delay after max errors
  - ❑Suffices to make brute-force guessing ineffective*
- Configure alerts to initiate investigation

*Why?

20

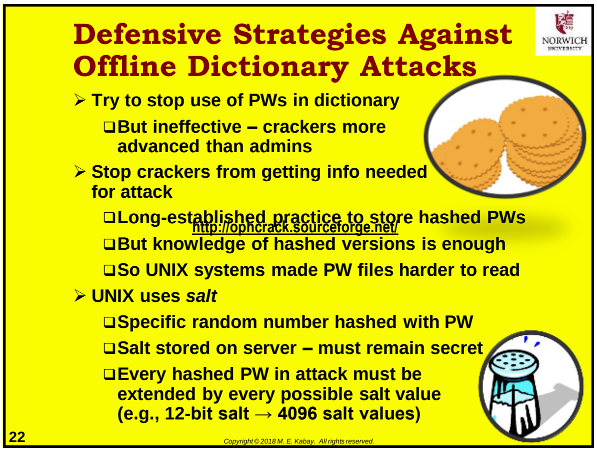


Risk of Off-Line Dictionary Attacks

- Copy PW file onto different computer
  - ❑Normally *one-way encrypted*
- Compare *encrypted forms* of *all possible PWs* with file
- Any match good enough to use for that PW
- Use dictionary of likely PWs in order of likely use for *PW-cracking* program
  - ❑E.g., ElcomSoft tools (see later slides)

Give an example of what this attack could look like

21

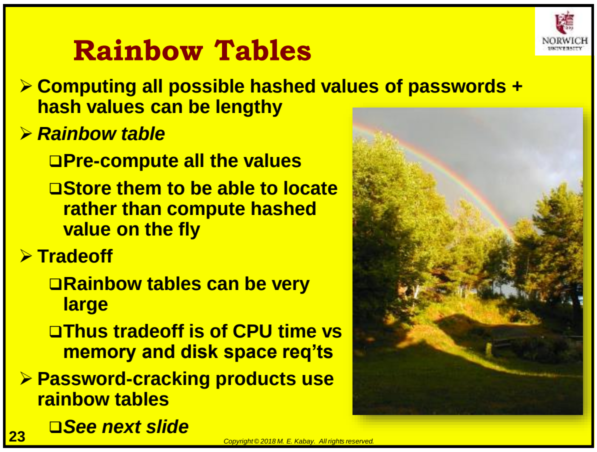


Defensive Strategies Against Offline Dictionary Attacks

- Try to stop use of PWs in dictionary
  - ❑But ineffective – crackers more advanced than admins
- Stop crackers from getting info needed for attack
  - ❑Long-established practice to store hashed PWs
  - ❑But knowledge of hashed versions is enough
  - ❑So UNIX systems made PW files harder to read
- UNIX uses *salt*
  - ❑Specific random number hashed with PW
  - ❑Salt stored on server – must remain secret
  - ❑Every hashed PW in attack must be extended by every possible salt value (e.g., 12-bit salt → 4096 salt values)

http://ophcrack.sourceforge.net/

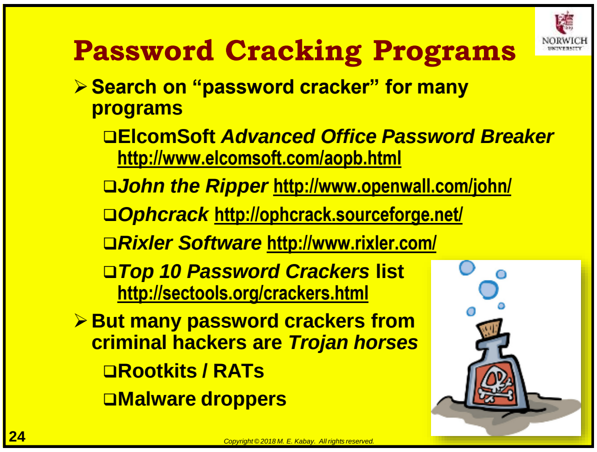
22



Rainbow Tables

- Computing all possible hashed values of passwords + hash values can be lengthy
- *Rainbow table*
  - ❑Pre-compute all the values
  - ❑Store them to be able to locate rather than compute hashed value on the fly
- Tradeoff
  - ❑Rainbow tables can be very large
  - ❑Thus tradeoff is of CPU time vs memory and disk space req'ts
- Password-cracking products use rainbow tables
  - ❑*See next slide*

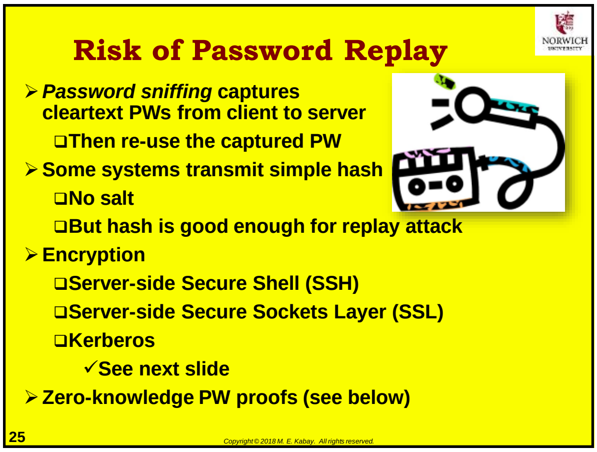
23



Password Cracking Programs

- Search on "password cracker" for many programs
  - ❑ElcomSoft *Advanced Office Password Breaker* http://www.elcomsoft.com/aopb.html
  - ❑*John the Ripper* http://www.openwall.com/john/
  - ❑*Ophcrack* http://ophcrack.sourceforge.net/
  - ❑*Rixler Software* http://www.rixler.com/
  - ❑*Top 10 Password Crackers* list http://sectools.org/crackers.html
- But many password crackers from criminal hackers are *Trojan horses*
  - ❑Rootkits / RATs
  - ❑Malware droppers

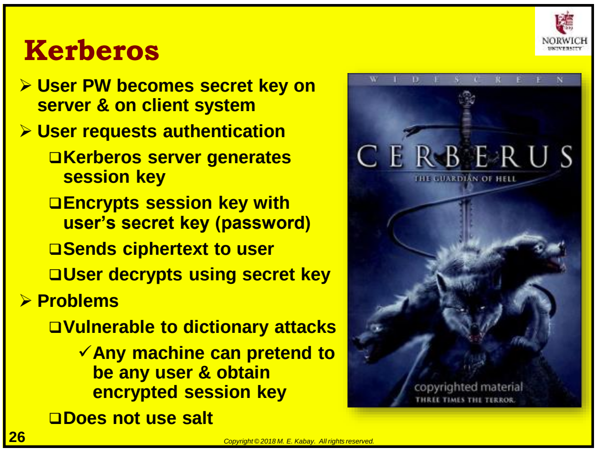
24



Risk of Password Replay

- ➢ ***Password sniffing*** captures cleartext PWs from client to server
  - ❑Then re-use the captured PW
- ➢ Some systems transmit simple hash
  - ❑No salt
  - ❑But hash is good enough for replay attack
- ➢ Encryption
  - ❑Server-side Secure Shell (SSH)
  - ❑Server-side Secure Sockets Layer (SSL)
  - ❑Kerberos
    - ✓See next slide
- ➢ Zero-knowledge PW proofs (see below)

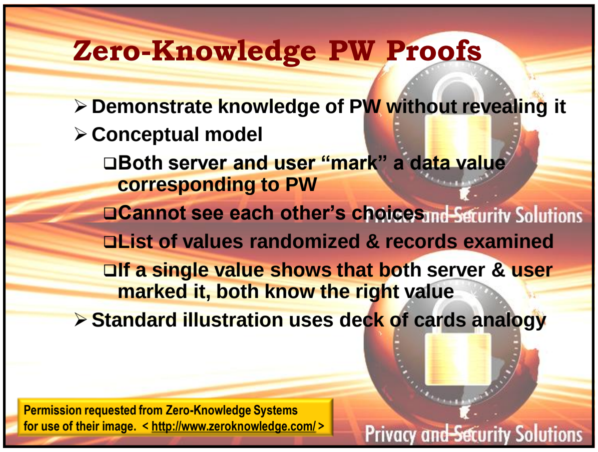
25



Kerberos

- ➢ User PW becomes secret key on server & on client system
- ➢ User requests authentication
  - ❑Kerberos server generates session key
  - ❑Encrypts session key with user's secret key (password)
  - ❑Sends ciphertext to user
  - ❑User decrypts using secret key
- ➢ Problems
  - ❑Vulnerable to dictionary attacks
    - ✓Any machine can pretend to be any user & obtain encrypted session key
  - ❑Does not use salt

26

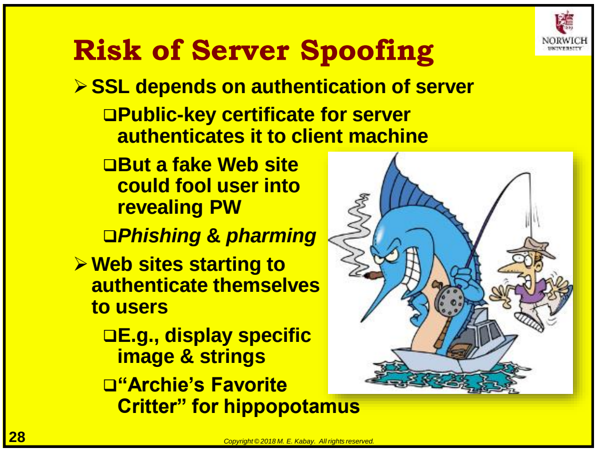


Zero-Knowledge PW Proofs

- ➢ Demonstrate knowledge of PW without revealing it
- ➢ Conceptual model
  - ❑Both server and user "mark" a data value corresponding to PW
  - ❑Cannot see each other's choices
  - ❑List of values randomized & records examined
  - ❑If a single value shows that both server & user marked it, both know the right value
- ➢ Standard illustration uses deck of cards analogy

Permission requested from Zero-Knowledge Systems for use of their image. < http://www.zeroknowledge.com/ >

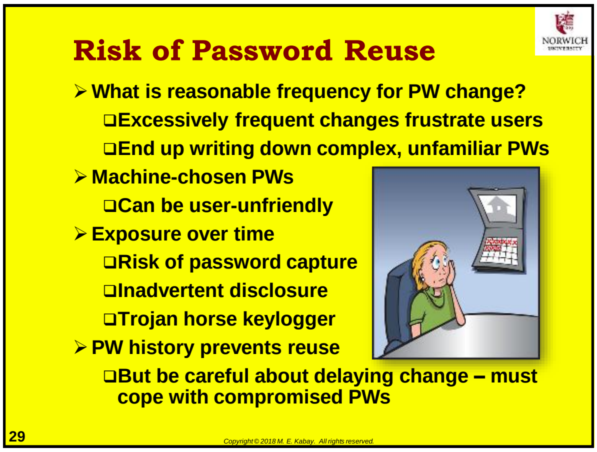
Privacy and Security Solutions
Privacy and Security Solutions



Risk of Server Spoofing

- ➢ SSL depends on authentication of server
  - ❑Public-key certificate for server authenticates it to client machine
  - ❑But a fake Web site could fool user into revealing PW
  - ❑*Phishing* & *pharming*
- ➢ Web sites starting to authenticate themselves to users
  - ❑E.g., display specific image & strings
  - ❑"Archie's Favorite Critter" for hippopotamus

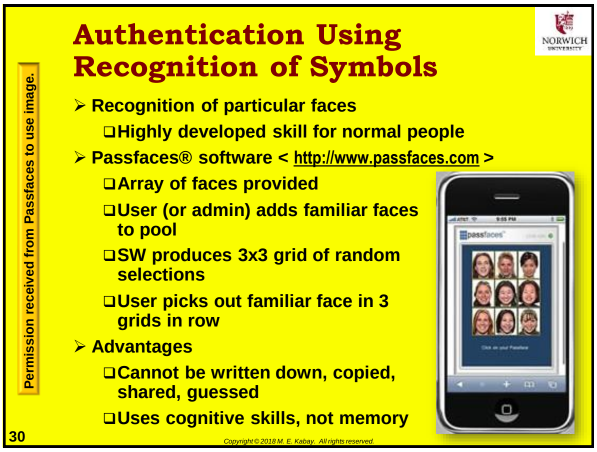
28



Risk of Password Reuse

- ➢ What is reasonable frequency for PW change?
  - ❑Excessively frequent changes frustrate users
  - ❑End up writing down complex, unfamiliar PWs
- ➢ Machine-chosen PWs
  - ❑Can be user-unfriendly
- ➢ Exposure over time
  - ❑Risk of password capture
  - ❑Inadvertent disclosure
  - ❑Trojan horse keylogger
- ➢ PW history prevents reuse
  - ❑But be careful about delaying change – must cope with compromised PWs

29



Authentication Using Recognition of Symbols

- ➢ Recognition of particular faces
  - ❑Highly developed skill for normal people
- ➢ Passfaces® software < http://www.passfaces.com >
  - ❑Array of faces provided
  - ❑User (or admin) adds familiar faces to pool
  - ❑SW produces 3x3 grid of random selections
  - ❑User picks out familiar face in 3 grids in row
- ➢ Advantages
  - ❑Cannot be written down, copied, shared, guessed
  - ❑Uses cognitive skills, not memory

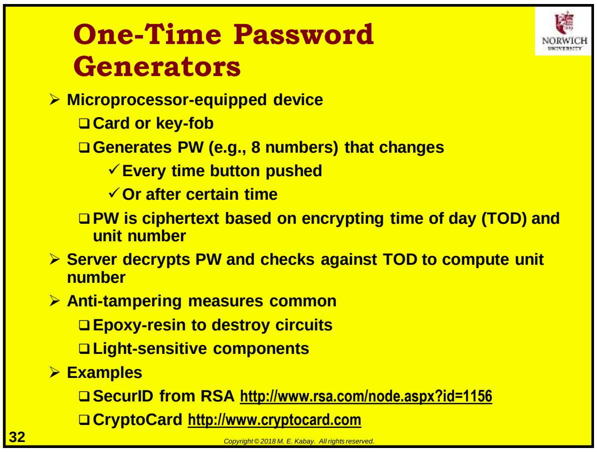
Permission received from Passfaces to use image.

30

Token-Based Authentication

- ➢ One-Time Password Generators
- ➢ Smart Cards and Dongles
- ➢ Soft Tokens

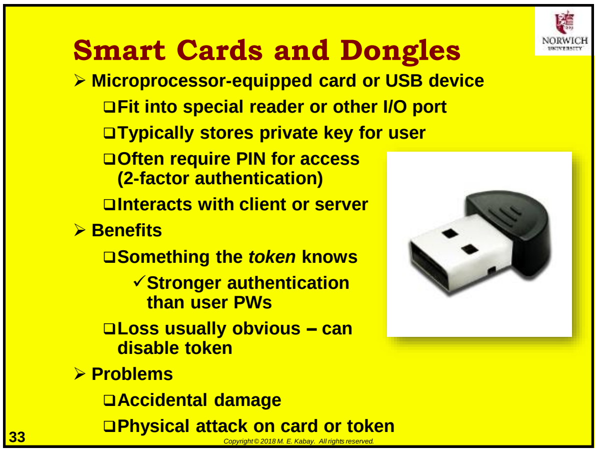
31



One-Time Password Generators

- ➢ Microprocessor-equipped device
  - ❑ Card or key-fob
  - ❑ Generates PW (e.g., 8 numbers) that changes
    - ✓ Every time button pushed
    - ✓ Or after certain time
  - ❑ PW is ciphertext based on encrypting time of day (TOD) and unit number
- ➢ Server decrypts PW and checks against TOD to compute unit number
- ➢ Anti-tampering measures common
  - ❑ Epoxy-resin to destroy circuits
  - ❑ Light-sensitive components
- ➢ Examples
  - ❑ SecurID from RSA http://www.rsa.com/node.aspx?id=1156
  - ❑ CryptoCard http://www.cryptocard.com

32



Smart Cards and Dongles

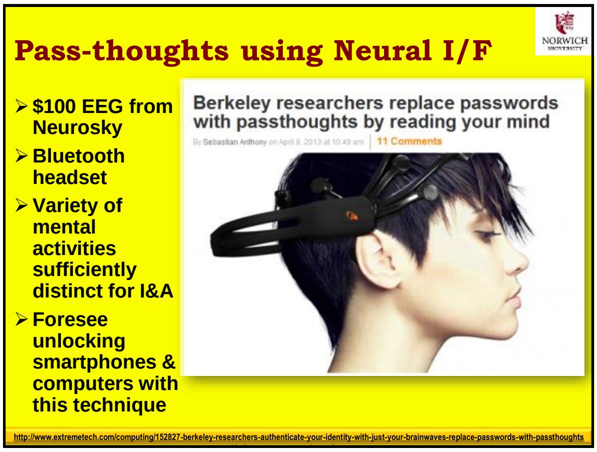
- ➢ Microprocessor-equipped card or USB device
  - ❑ Fit into special reader or other I/O port
  - ❑ Typically stores private key for user
  - ❑ Often require PIN for access (2-factor authentication)
  - ❑ Interacts with client or server
- ➢ Benefits
  - ❑ Something the *token* knows
    - ✓ Stronger authentication than user PWs
  - ❑ Loss usually obvious – can disable token
- ➢ Problems
  - ❑ Accidental damage
  - ❑ Physical attack on card or token

33

Software Tokens

- ➢ AKA *soft tokens*
- ➢ User's private key encrypted & stored on storage device
  - ❑ Originally floppy disks
  - ❑ Now USB flash drives
- ➢ But some people storing soft tokens on network servers
  - ❑ Reduces to a stored password
- ➢ Some systems working on splitting keys
- ➢ Others store user's private key on server
  - ❑ Enables user spoofing

34



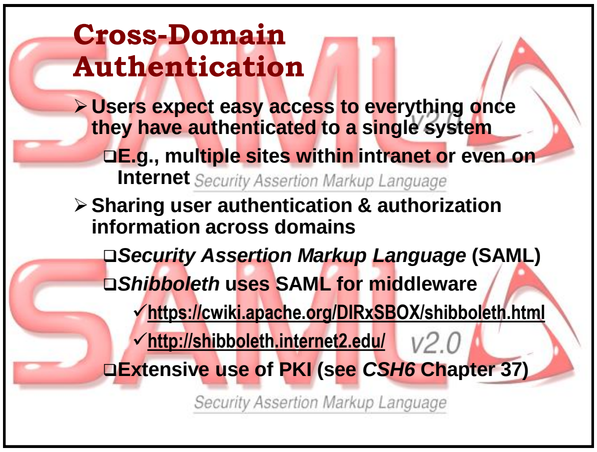
Pass-thoughts using Neural I/F

- ➢ $100 EEG from Neurosky
- ➢ Bluetooth headset
- ➢ Variety of mental activities sufficiently distinct for I&A
- ➢ Foresee unlocking smartphones & computers with this technique

Berkeley researchers replace passwords with passthoughts by reading your mind

By Sebastian Anthony on April 8, 2013 at 10:49 am | 11 Comments

http://www.extremetech.com/computing/152827-berkeley-researchers-authenticate-your-identity-with-just-your-brainwaves-replace-passwords-with-passthoughts



Cross-Domain Authentication

- ➢ Users expect easy access to everything once they have authenticated to a single system
  - ❑ E.g., multiple sites within intranet or even on Internet
- ➢ Sharing user authentication & authorization information across domains
  - ❑ *Security Assertion Markup Language* (SAML)
  - ❑ *Shibboleth* uses SAML for middleware
    - ✓ https://cwiki.apache.org/DIRxSBOX/shibboleth.html
    - ✓ http://shibboleth.internet2.edu/
  - ❑ Extensive use of PKI (see *CSH6* Chapter 37)

SAML
v2.0
Security Assertion Markup Language
SAML
v2.0
Security Assertion Markup Language

Biometric Authentication

- See *CSH6* Chapter 29

37

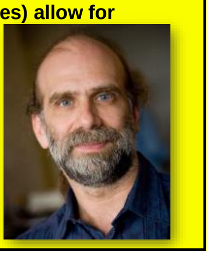
Relative Costs of Authentication Technologies

- Common belief: password are free
  - ❑FALSE
  - ❑Study by RSA Data Security
    - ✓Initializing each userID = $12
    - ✓Maintenance for 3 years = $660/user
  - ❑Fundamentally weak authentication method
- Tokens & biometrics
  - ❑Demonstrably less expensive
  - ❑More effective

38

Concluding Remarks

- Biometrics & tokens likely to replace PWs for high-end security
- IA experts should dispel misconception that I&A are sufficient for improving public safety
  - ❑Identifying someone ≠ trusting someone
  - ❑Closed populations (e.g., employees) allow for background checking
  - ❑But unscreened population (e.g., air passengers) provides no assurance of trustworthiness
- Should criticize *security theater* (Bruce Schneier's term) for security measures as substitute for effective public policy



39

Now go and study

40