One Time Password Access to Any Server without Changing the Server

Dinei Florêncio and Cormac Herley
Microsoft Research, Redmond

Acknowledgements: Eric Lawrence, Ziqing Mao, Nikita Pandey, Erin Renshaw, and Dany Rouhana

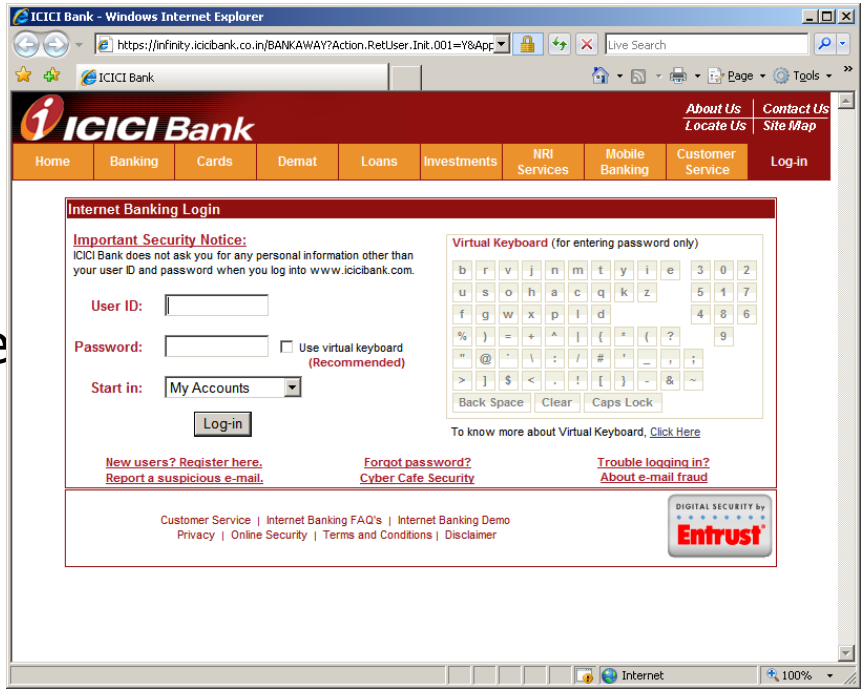
The Problem: Passwords do not resist replay

- Login from malware infected PC
  - Malware now has your password
- Untrusted machines
  - Internet Cafes
  - Any machine not properly patched/maintained.
- Unlike phishing, spam or 419 scams
  - Even sophisticated users susceptible

Our current best advice: “Good luck out there”

Other stuff to address this

- Coping strategies:
  - Type password in reverse order,
  - Cut and paste the password
  - Intersperse random chars (outside of the pwd field focus) between chars
- On-screen keyboard



- These can help against keyloggers
  - But not browser plugins
  - Or screen and mouse logging

Sustained safety: password must not be entered or downloaded to the untrusted machine

Setup: Public PC (or untrusted PC)

Publi
c
PC

Login Server
E.g. OWA,
BoA, PayPal

- Assume at an untrusted PC ***everything*** is logged
  - Every key, every click, every screen, all network traffic
- Cannot change ***anything*** at legacy login server
- Cannot instal or change ***anything*** at public PC
  - No boot from CD, run app from USB etc
  - No browser plug-ins, BHO
  - No smartcard reader
  - No choice of browser.
- Protecting password, ***not rest of session***

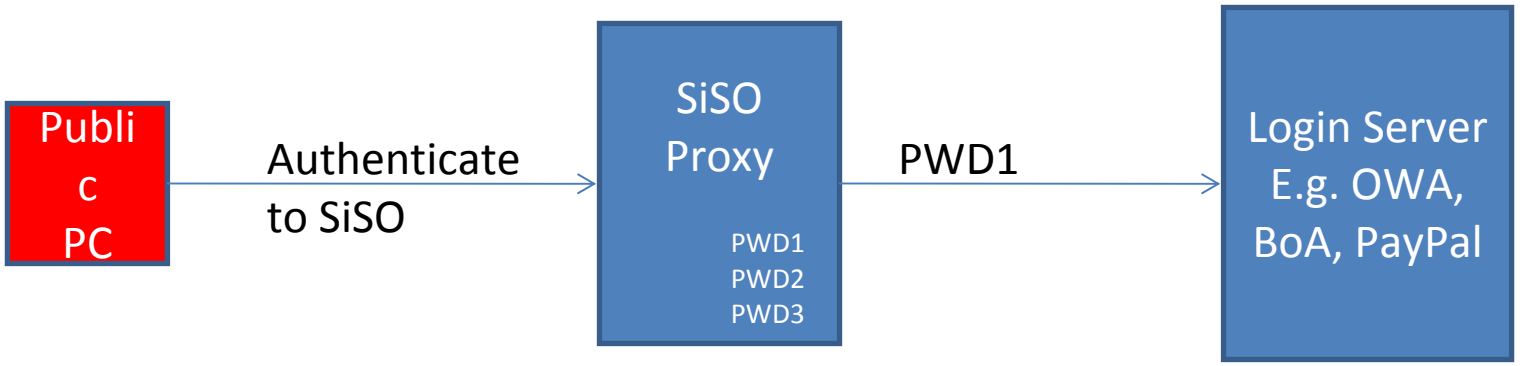
Our Solution: Universal Replay-Resistant Secure Authentication (URRSA)

- All you need is an address bar
- It works from anywhere
- With (almost) all login servers
- You can try it today*

*Caveats: trial for this conf only. We have not done PEN testing. Please no banking credentials. The password flows through the proxy.

OLD: Single Sign-on Server (SiSO)

- Store PWD in the cloud:

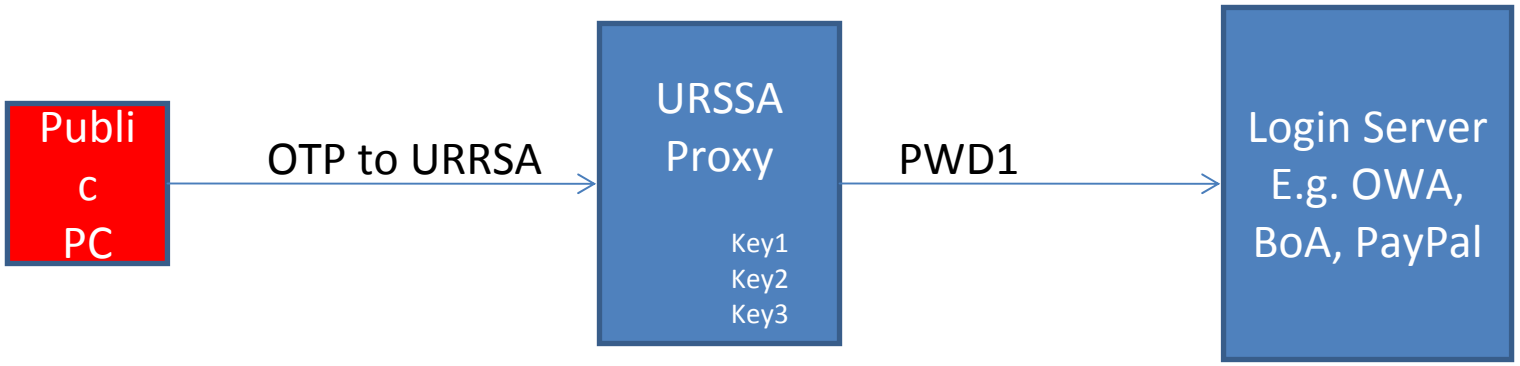


Why this is not good enough:

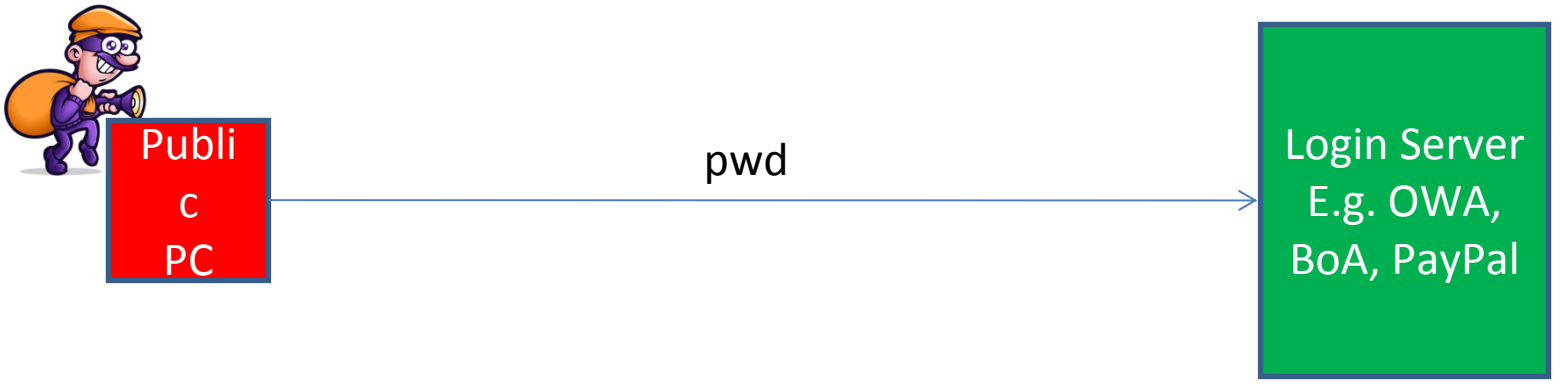
- Liability: Server storing millions of PWDs is now a target.
- Need to authenticate to SiSO proxy. Mishandling authentication may compromise all your accounts.

URRSA:

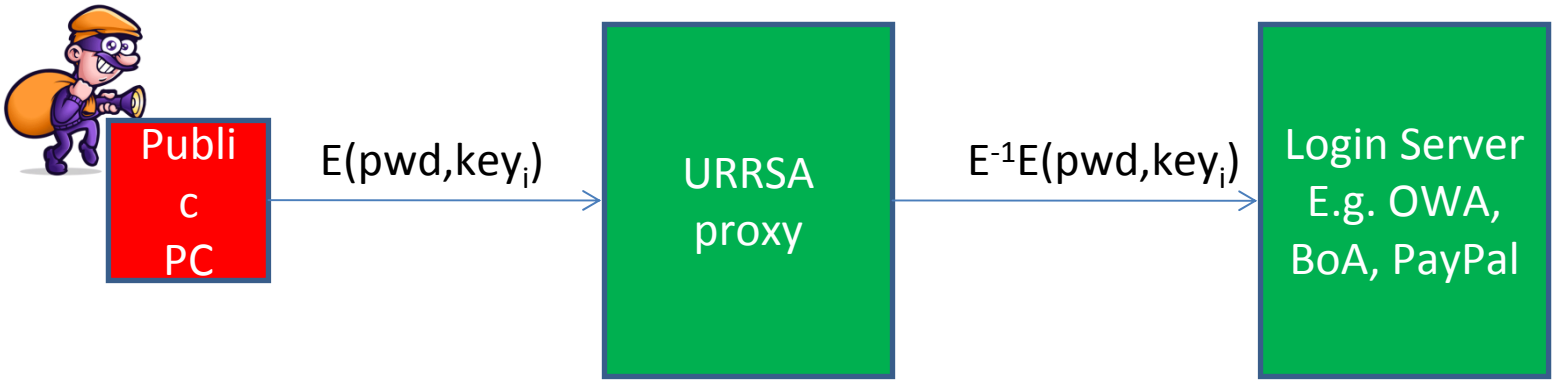
- OTP = encrypted pwd



- No need to authenticate to Proxy;
- Proxy does not store any PWDs, only (worthless) encryption keys.


Publi
c
PC
pwd
Login Server
E.g. OWA,
BoA, PayPal

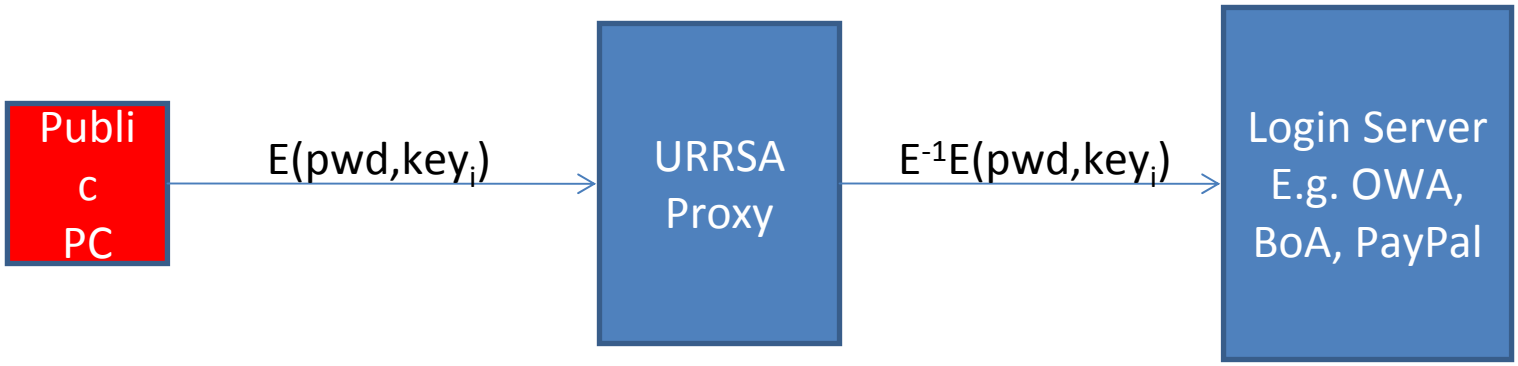
URRSA



- Place Proxy as Man-In-The-Middle;
- Type *Encrypted* PWD at PC
- Proxy decrypts once and only once
- Proxy needs to know key$_i$ in advance. HOW?

On the Road: From Public PC

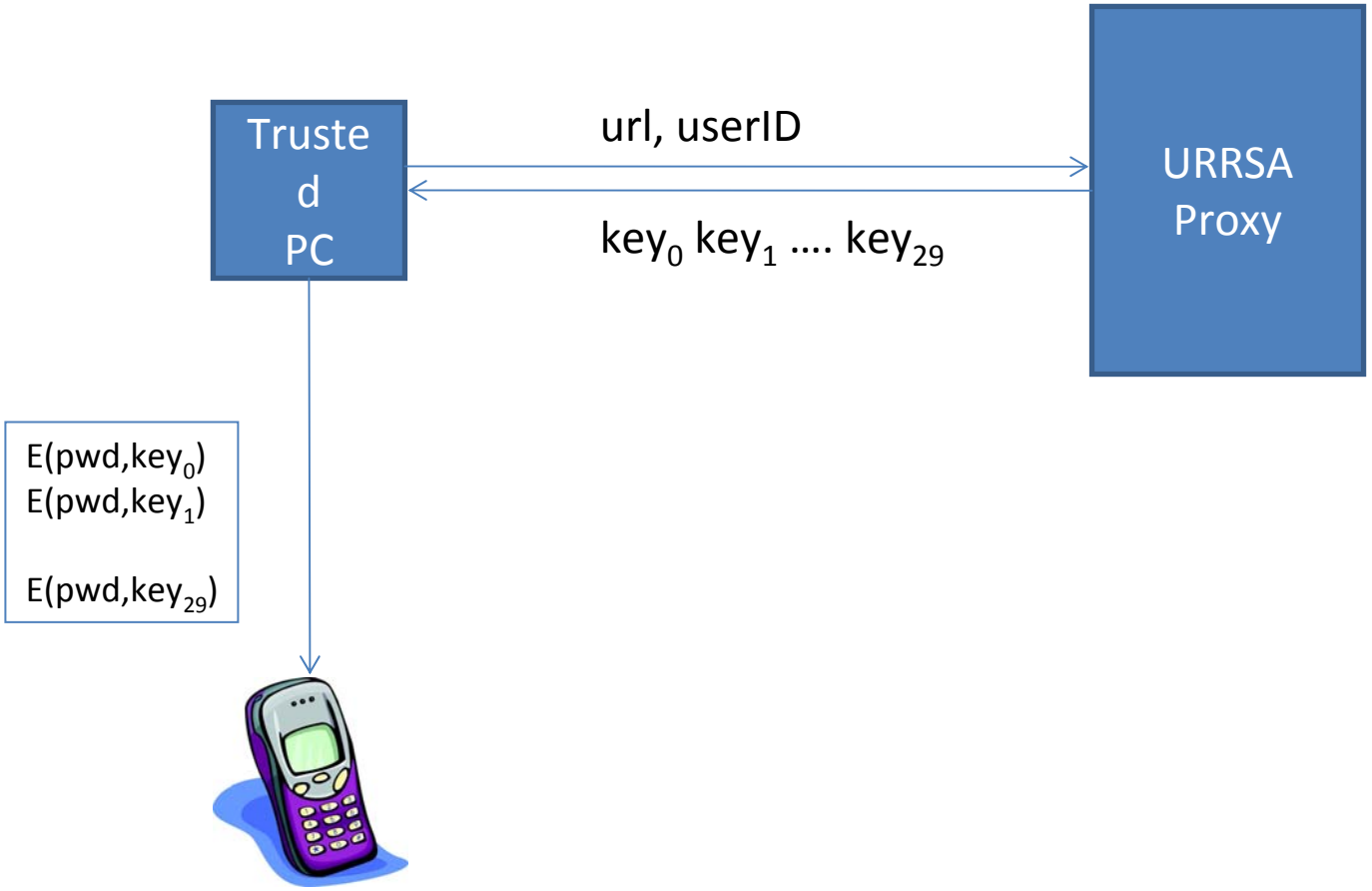
- Enter E(pwd,key)



- Proxy serves as a Man-In-The-Middle

Convert Any Server into One-Time Password Server

- Before you travel: From a trusted PC



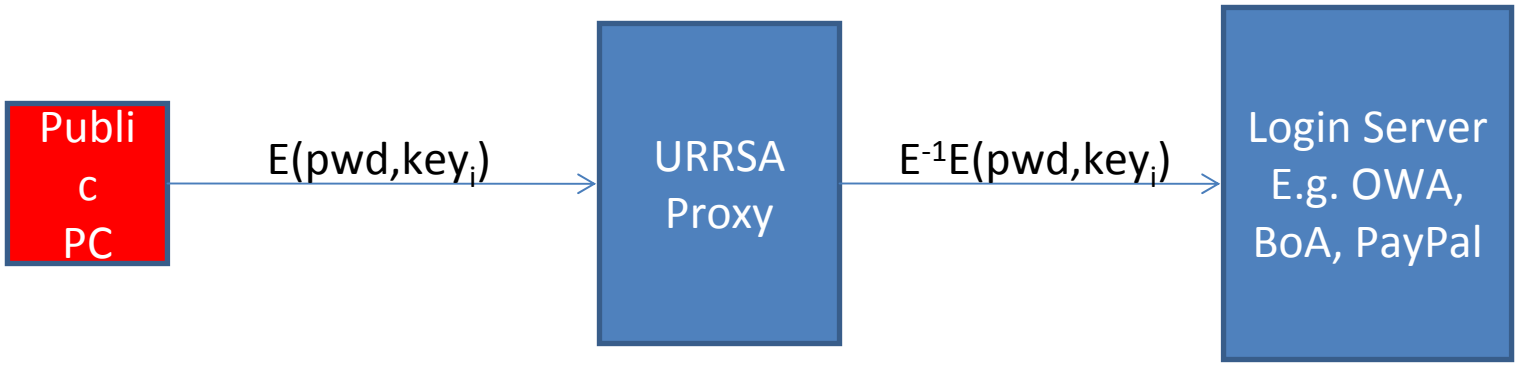
Registering (safe location)

- Register (url, userID)
- Enter password
- Server generates:

- User carries
  - $E(p,Key_0)$ ….. $E(p,Key_{29})$

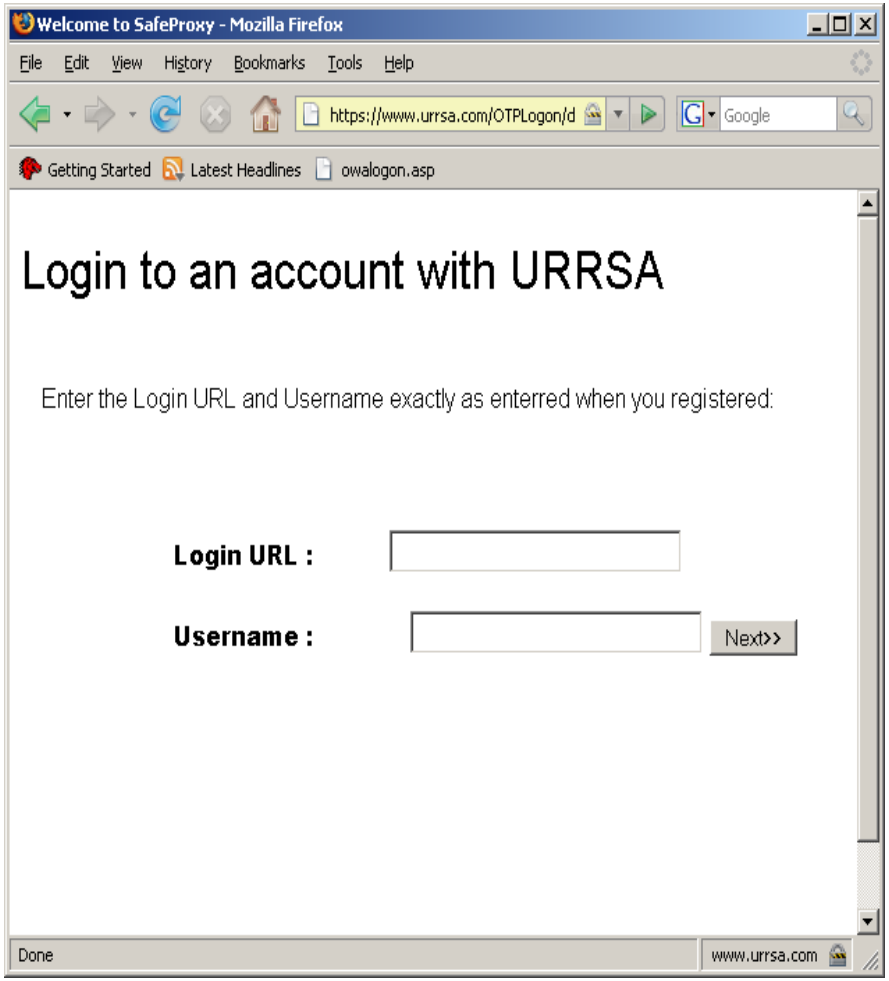
On the Road: From Public PC

- Enter E(pwd,key)



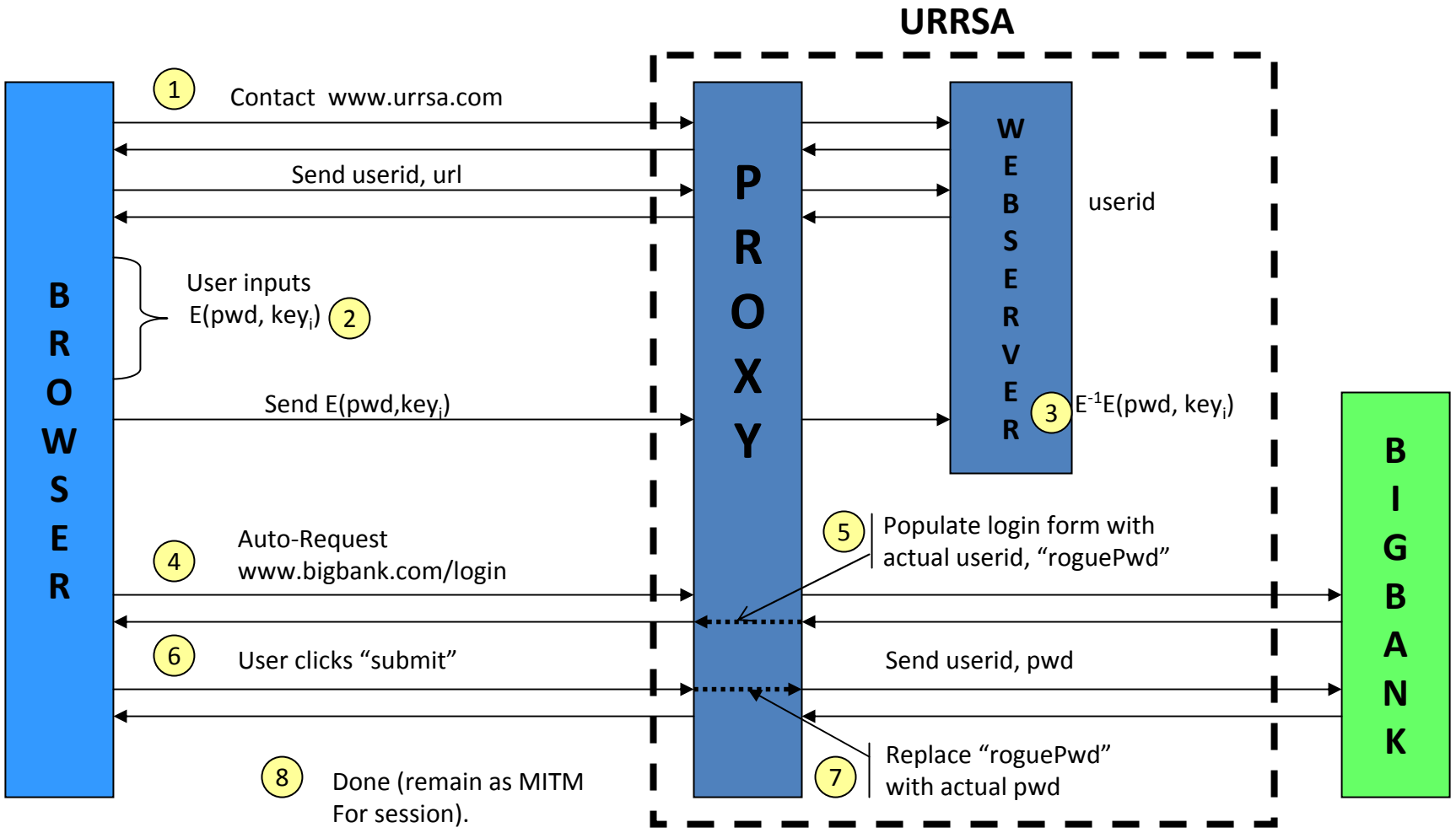
- Proxy serves as a Man-In-The-Middle

Logging in (unsafe location)



- Enter (url, userID)
- This *uniquely identifies* user
- For i-th login user enters
  - $E(p, Key_i)$
- URRSA fwds
  - $E^{-1}E(p, Key_i)$

How the Password Swap gets Done



A few NOT's

- ***This is not a SiSo scheme***
  - Password is not stored at the proxy
    - (though of course must trust proxy)
- ***The proxy does not authenticate the user***
  - URRSA decrypts and fwds with agreed-upon keys
  - If you're not who you claim then login fails

Two Important Details

- How does Mapping and Encryption work?
- How does URRSA get to be MITM?

Mapping strategies

- PASSWORDS: We need to allow special characters: MyP^3&
- We'll allow for 128 characters:
  - 26 lower case + 26 UPPER CASE;
  - 10 digits;
  - 66 special symbols (rarely used, but…);
- But most people use simple passwords… Snoopy
  Standard Mapping would map Snoopy to u\6#¿K

- We'll Huffman encode the characters
- We'll use only 32 easy to find characters:
  - ABCDEFGHJKLMNPQRSTUVWXYZ23456789
  - No confusion sets (e.g., 0-O, l-I-1);
  - All easy to find, even on foreign keyboards;
  - All allowed on SMS.


Esc
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
8
Alt Gr
Ctrl
Alt
Ctrl

Mapping strategies

- PASSWORDS: We need to allow special characters: MyP^3&
- We'll allow for 128 characters:
  - 26 lower case + 26 UPPER CASE;
  - 10 digits;
  - 66 special symbols (rarely used, but…);
- But most people use simple passwords… Snoopy
  Standard Mapping would map Snoopy to uT6#¿K

- We'll huffman encode the characters
- We'll use only 32 easy to find characters:
  - ABCDEFGHJKLMNPQRSTUVWXYZ23456789
  - No confusion sets (e.g., 0-O, l-I-1);
  - All easy to find, even on foreign keyboards;
  - All allowed on SMS.

Current implementation:

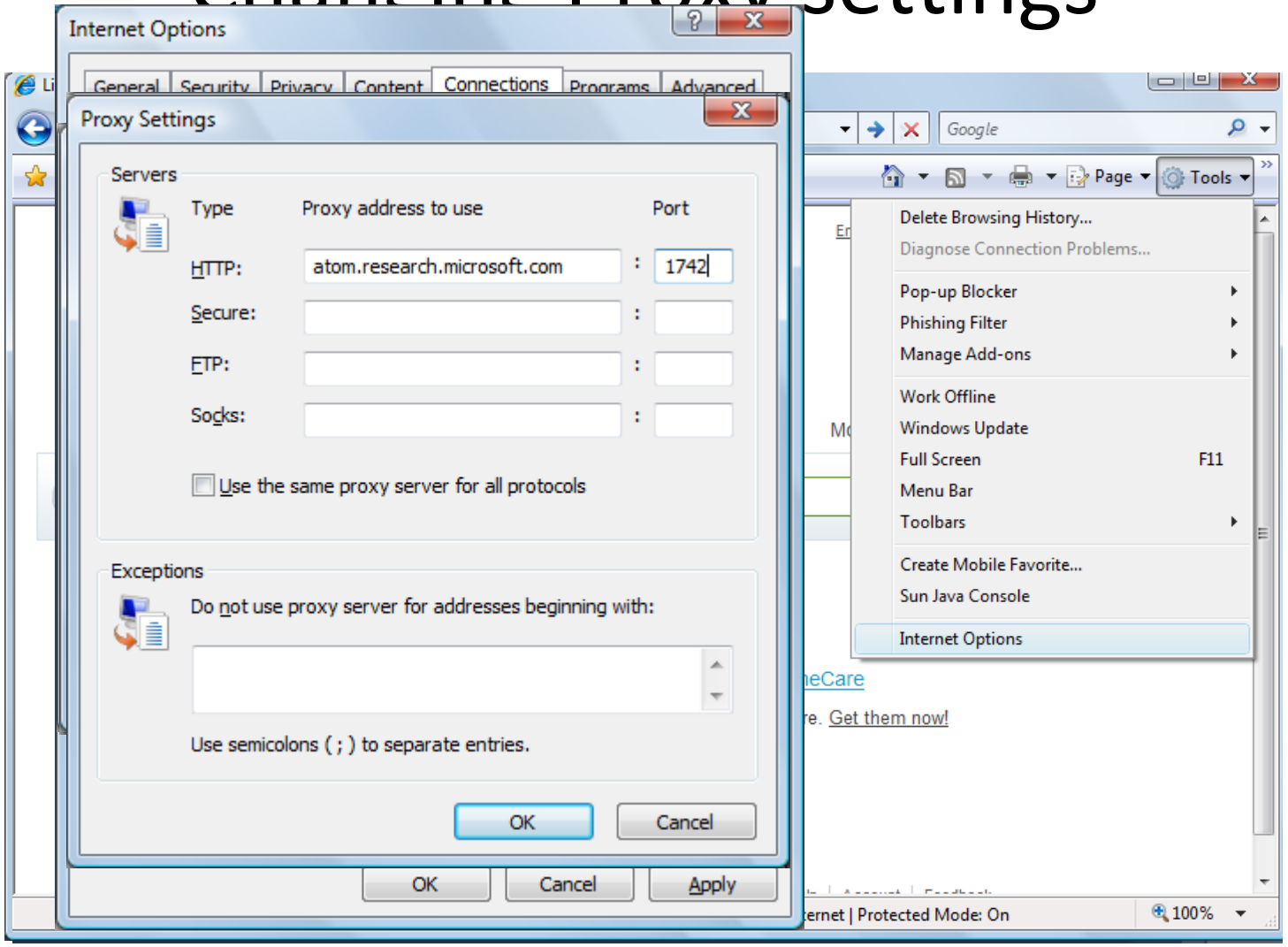
- Input 128 characters => 7 bits/char.
- Huffman encode: a = 3bits, ^=12 bits
- Encrypt all PWD bits
- Maps encrypted bits to OTP => 5 bits/char
- Thus, pwd maps to *ceil*(H(pwd)/5) OTP. E.g.,
  - Seattle => PJ45JWM9ZT

Original PWD	S	e	a	t…
Huffman Codes	1 0 1 0 1 0 1	1 0 1 0 1	0 0 1	1 0 1 …
Encrypted	0 1 1 1 0 0 1	0 0 0 1 1	0 1 0 1 1	0 …
5-bits regroup	0 1 1 1 0	0 1 0 0 0	1 1 0 1 0	1 1 0 …
OTP	P	J	4	5

So how does our Proxy get to be a Man-In-the-Middle?

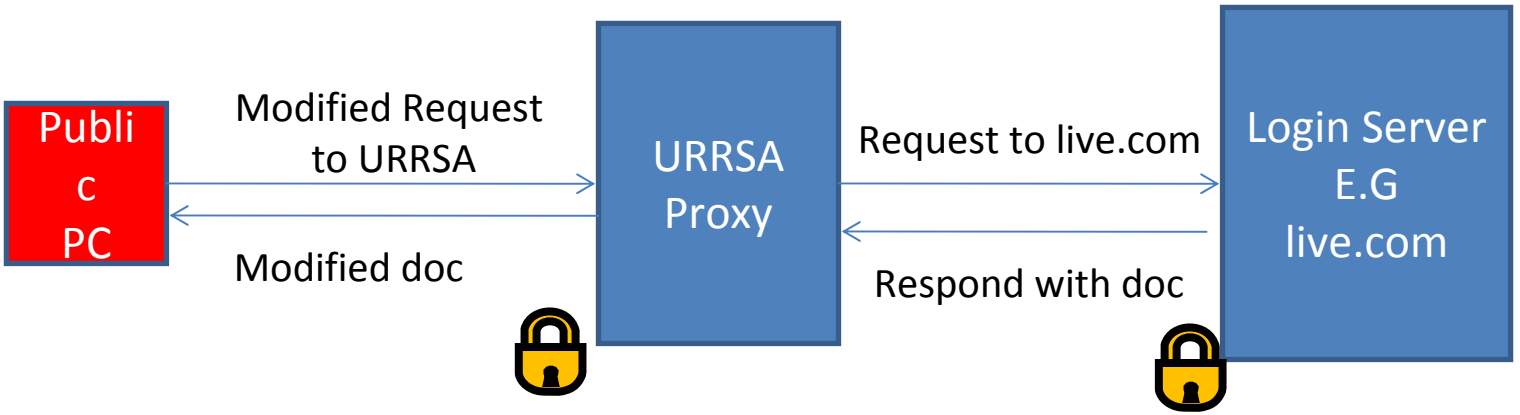
- HTTP Proxy
- Reverse (link translating) Proxy

To use HTTP Proxy: Changing Proxy settings



Don’t forget to undo settings when you leave!!

Instead use Reverse Proxy (MITM)



1. Request doc.htm from urrsa.com instead of live.com
2. URRSA forwards to live.com
3. URRSA receives doc.htm. Modify all links to point to urrsa.com instead of live.com
4. Fwd modified doc.htm to browser

Note: URRSA keeps two separate SSL cxns, one with client, one with server

Demo

Certificates and Browsers

- URRSA treats each domain as a subdomain
  - E.g. live.com becomes live.com.urrsa.com
  - Wildcard cert for *.urrsa.com
- Browsers handle wildcard certs differently
  - IE, Safari: one warning per host loaded
  - FireFox, Opera: no errors
- This fixable, but for now
  - IE: one cert warning for OWA, live.com, facebook, …

Don’t take our word for it…….
Try it yourself

- www.urrsa.com
  - Can register (i.e. get the OTP’s) only from corpnet
- Don’t:
  - Access banking (i.e. send your bank credentials)
  - Access financial sites
- Do:
  - Try free accounts: hotmail, gmail, yahoo, myspace, PayPal, facebook,
  - Tell us about any problems: feedback@urrsa.com

Resources

- URRSA: www.urrsa.com

Attacks

- Session hijacking
  - Vulnerability reduced, since proxy limits attacker actions. (e.g., cannot change pwd). Other limitations/authentication can be added, as proxy knows user is in a hostile environment.
  - OTP stealing : same as above, plus:
  - Stolen OTPs rendered useless by next use;
  - Leaves a clue; user may revoke unused OTPs.
  - Stealing not an attack for Window Mobile version.

Session Hijacking

- Attack:
  - A resident software awaits until a user logins in, then overtakes the session, and/or asks unauthorized transactions;
- Why less critical with URRSA:
  - Since all session flows through proxy:
    - Attacker cannot change PWD.
    - We may add other restrictions/authentication, as per “Delegate” paper.

OTP stealing

- Attack:
  - A resident software awaits until a user types the OTP, holds off that OTP, and tells user login was not successful. Then asks for a second password, but send first one to the proxy. Keeps second OTP for future use.
- Why less critical with URRSA:
  - Since all session flows through proxy:
    - Attacker cannot change PWD.
    - We may add other restrictions/authentication, as per “Delegate” paper.
  - Second PWD useless after User logs-in again at other Café.
  - Leaves a clue; may make user suspicious. User may test/revoke.
- Solved with Windows Mobile version: passwords not requested in sequence.

When does it fail?

- Password switch doesn't work
  - Site does not use HTML forms
    - E.g. www.firsttechCU.com uses Flash
  - Site does not transmit password
    - E.g. www.Livejournal.com sends MD5 of pwd (thanks Faisal)
  - Site restricts format of password sent
    - E.g.
- Reverse Proxying breaks
  - IIS has 260 charURL limit (Apache served URLs can be longer)
  - Problems with certain special chars, e.g. ':', '*'