ECE750: Usable Security and Privacy

Communicating Securely
Why Johnny Can't Encrypt

Dr. Kami Vaniea,
Electrical and Computer Engineering
kami.vaniea@uwaterloo.ca

UNIVERSITY OF WATERLOO | FACULTY OF ENGINEERING

First, something random...

- First 5 minutes we talk about something interesting, often from recent events
- You will not be tested on the 5 minutes part of lecture
- This part of lecture will sometimes not be recorded
- Why do this?
    1. Some students show up late
    2. Reward students who show up on time
    3. Important to see real world examples

Today…

1. Overview of public/private key encryption
2. Cognitive Walkthrough
3. Deep discussion of the paper: Why Johnny Can't Encrypt

Why Johnny Can’t Encrypt

In Proceedings of the 8th USENIX Security Symposium, August 1999, pp. 169-183

Why Johnny Can’t Encrypt: A Usability Evaluation of PGP 5.0

Alma Whitten
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
alma@cs.cmu.edu

J. D. Tygar[1]
EECS and SIMS
University of California
Berkeley, CA 94720
tygar@cs.berkeley.edu

Abstract

User errors cause or contribute to most computer security failures, yet user interfaces for security still tend to be clumsy, confusing, or near-nonexistent. Is this simply due to a failure to apply standard user interface design techniques to security? We argue that, on the contrary, effective security requires a different usability standard, and that it will not be achieved through the user interface design techniques appropriate to other types of consumer software.

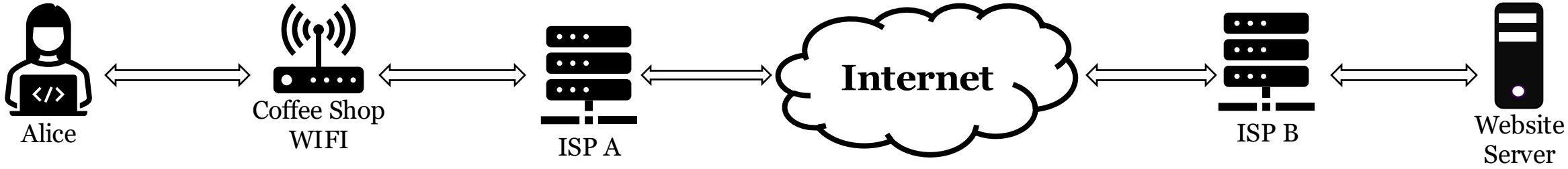
To test this hypothesis, we performed a case study of a security program which does have a good user interface by general standards: PGP 5.0. Our case study used a cognitive walkthrough analysis together with a laboratory user test to evaluate whether PGP 5.0 can be successfully used by cryptography novices to achieve effective electronic mail security. The analysis found a number of user interface design flaws that may

1 Introduction

Security mechanisms are only effective when used correctly. Strong cryptography, provably correct protocols, and bug-free code will not provide security if the people who use the software forget to click on the encrypt button when they need privacy, give up on a communication protocol because they are too confused about which cryptographic keys they need to use, or accidentally configure their access control mechanisms to make their private data world-readable. Problems such as these are already quite serious: at least one researcher [2] has claimed that configuration errors are the probable cause of more than 90% of all computer security failures. Since average citizens are now increasingly encouraged to make use of networked computers for private transactions, the need to make security manageable for even untrained users has become critical [4, 9].

Sample connection: Alice loads a website

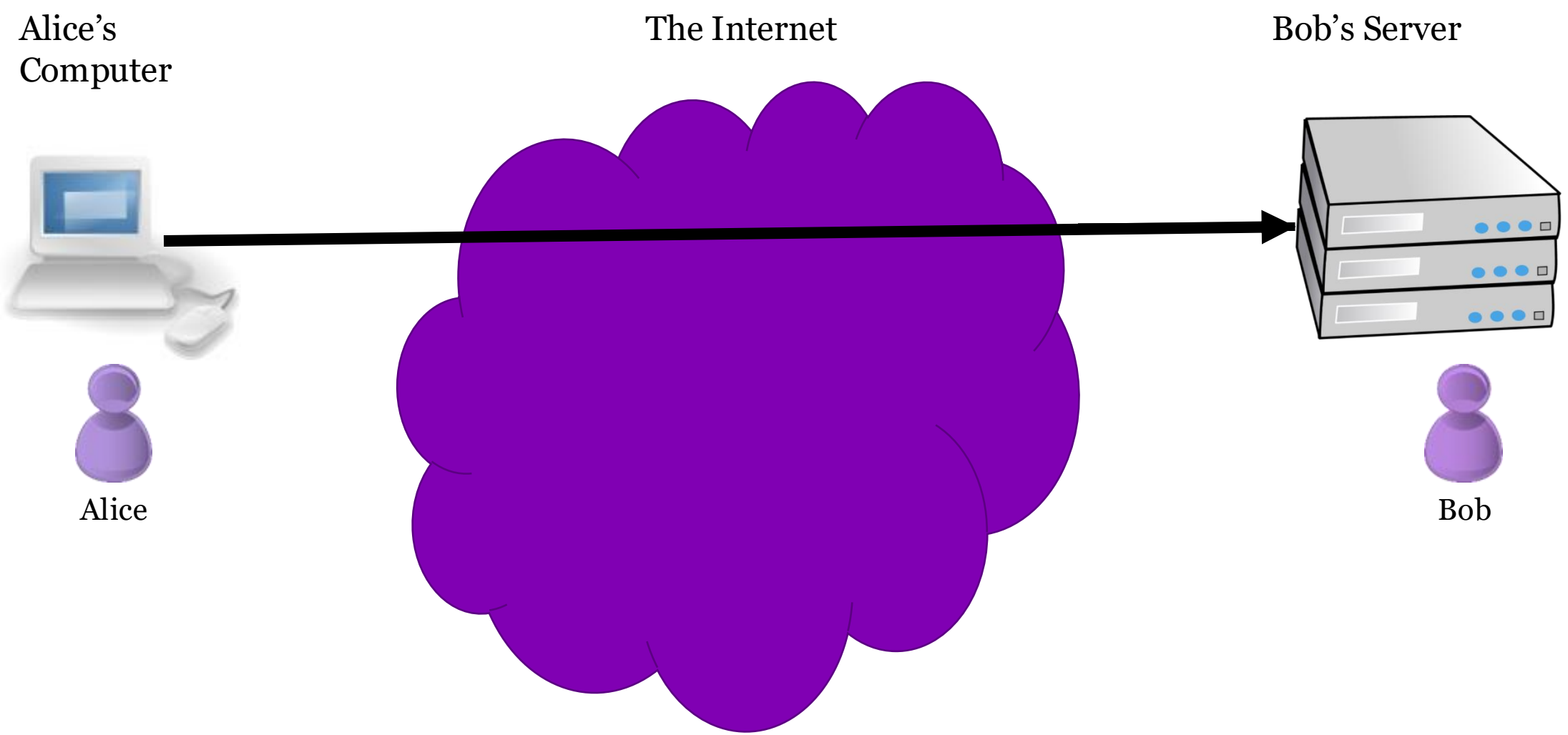
Alice visits: http://example.com

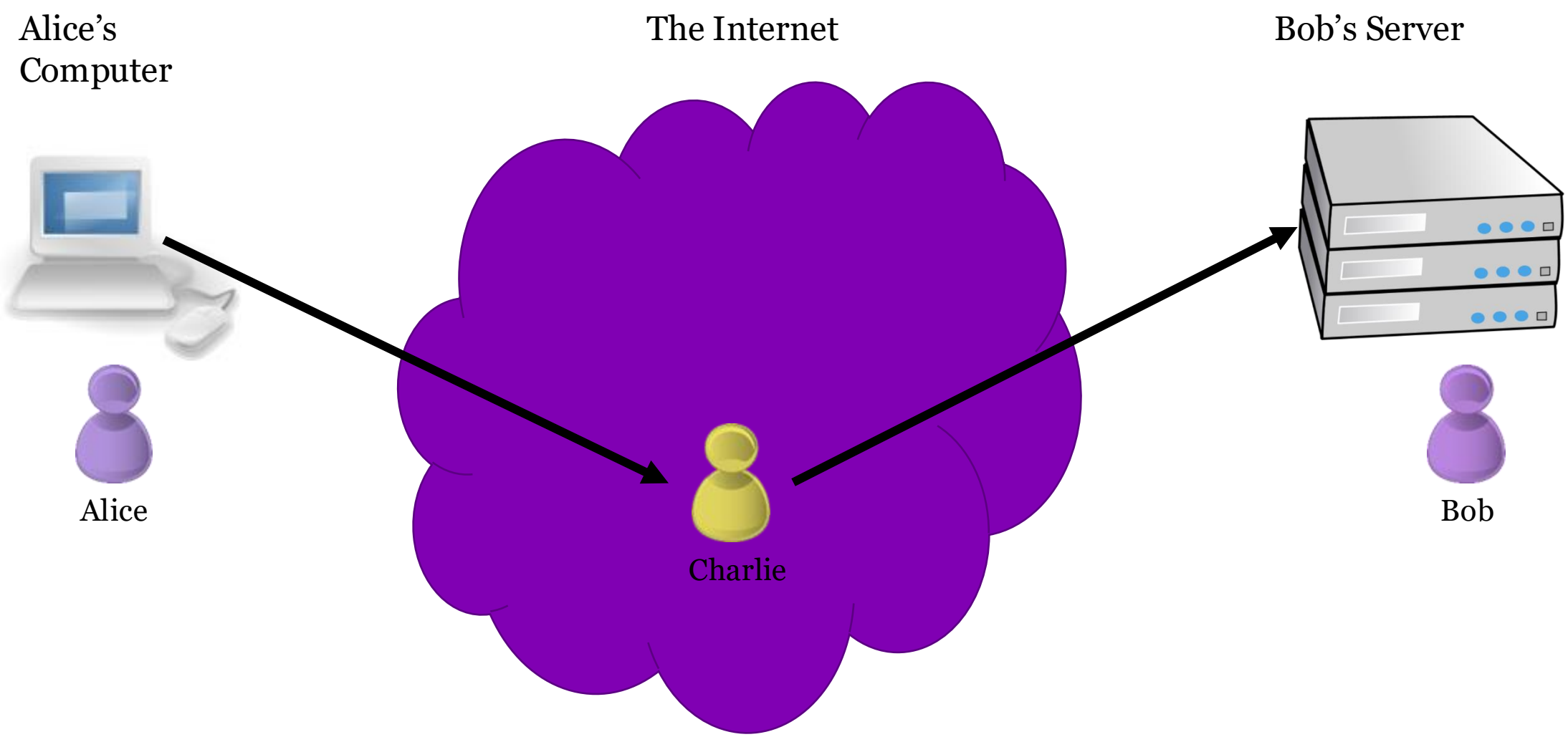


For each of the above connection points, can they learn:

1. The name and/or IP address of the website Alice is visiting
2. The content of the webpage Alice is viewing
3. Alice's Operating System (Linux, Windows, MacOS)

MAN IN THE MIDDLE ATTACK

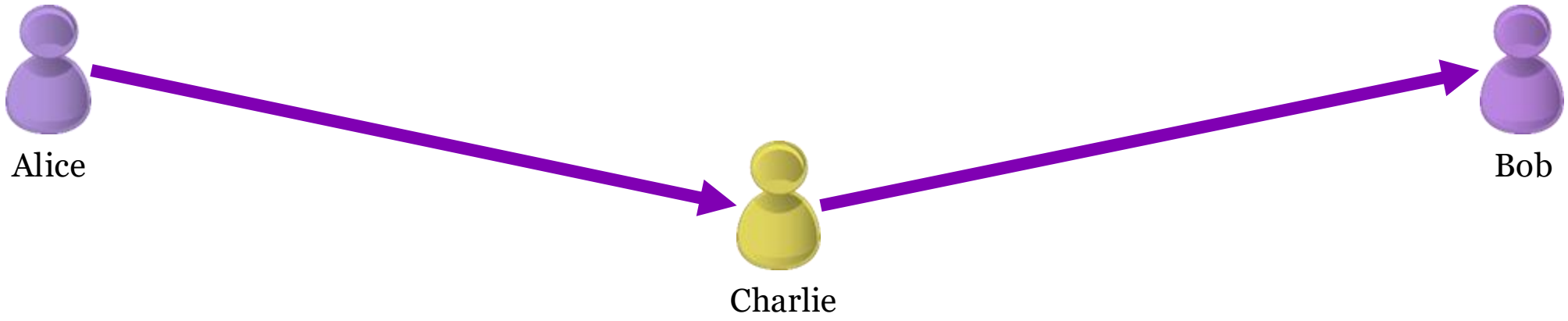

Alice's Computer
The Internet
Bob's Server
Alice
Bob


Alice's Computer
The Internet
Bob's Server
Alice
Charlie
Bob

Charlie is in the middle between Alice and Bob.

- Charlie can:
  - View (confidentiality)
  - Change (integrity)
  - Add (integrity)
  - Delete (integrity, availability)

- Charlie could be:
  - Internet service provider
  - Virtual Private Network (VPN) provider
  - WIFI provider such as a coffee shop
  - An attacker re-routing your connection
  - An incompetent admin (it happens)



Man in the middle attacks happen all the time and they are not always bad.

Alice goes to her favorite coffee shop and tries to visit BBC News

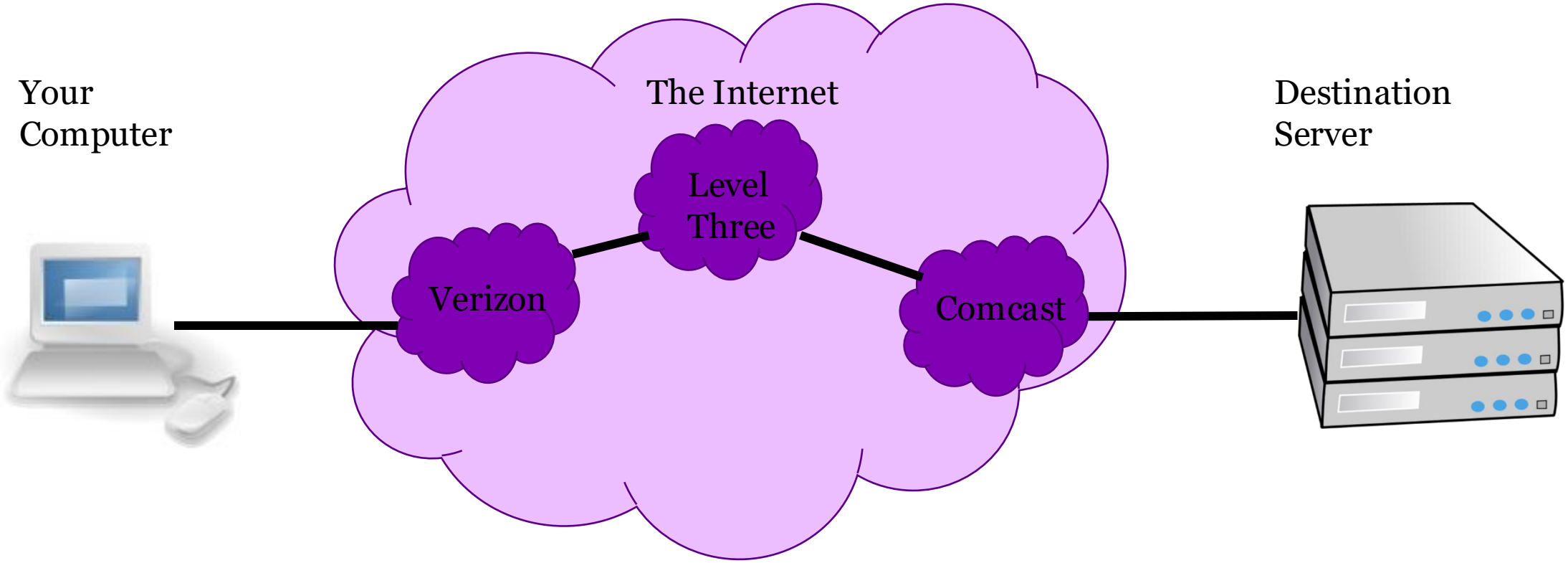
Alice

Alice

BBC NEWS

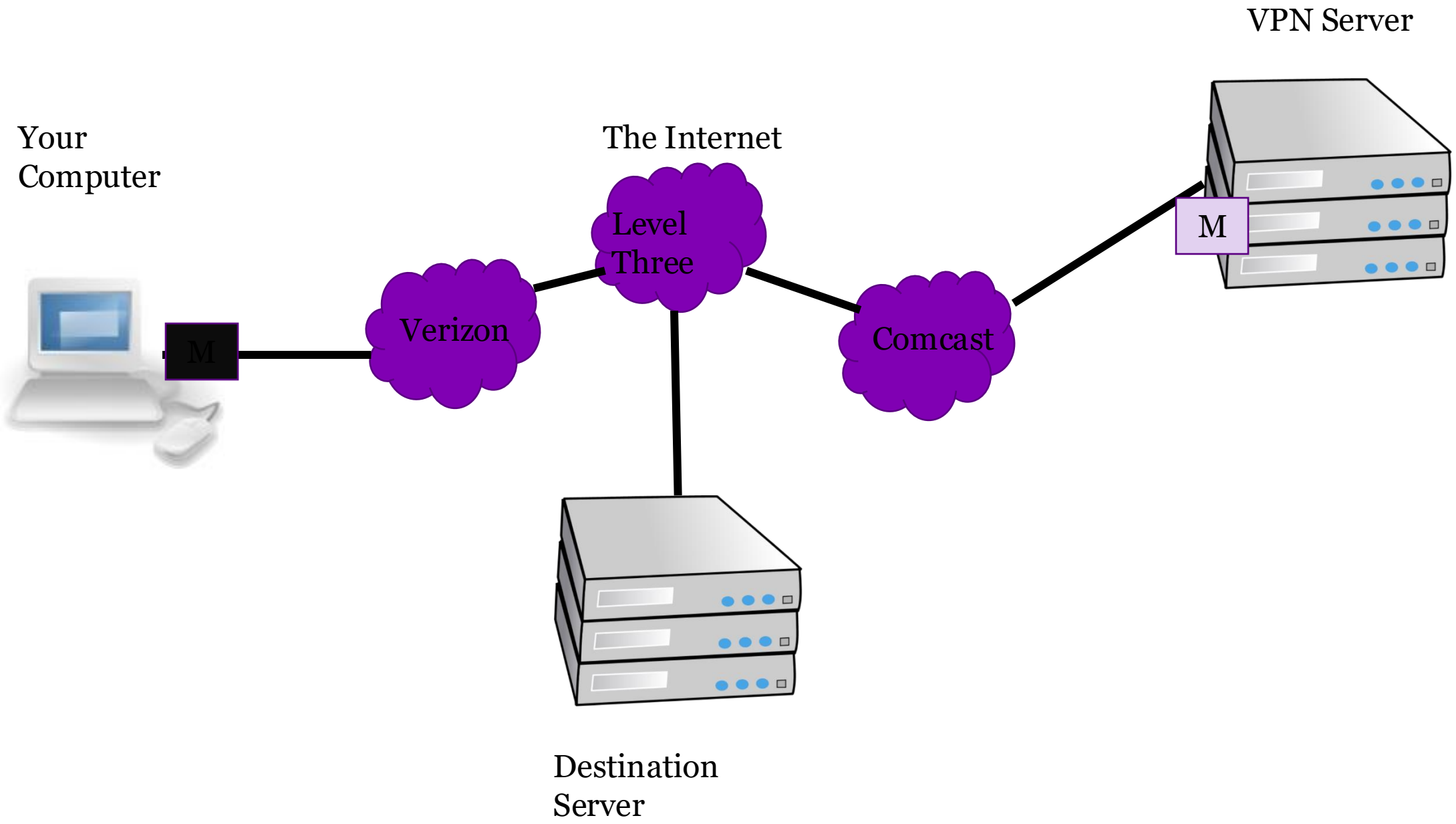

I'm Feelin
STARBUCKS
Google
WIFI
Free Wi-Fi
From our friends at Google
Accept & Connect
I agree to the Terms of Service and have reviewed the Google Privacy Policy
Need help? 855-446-2374
¡HOLA
GOOD MORNING
Google

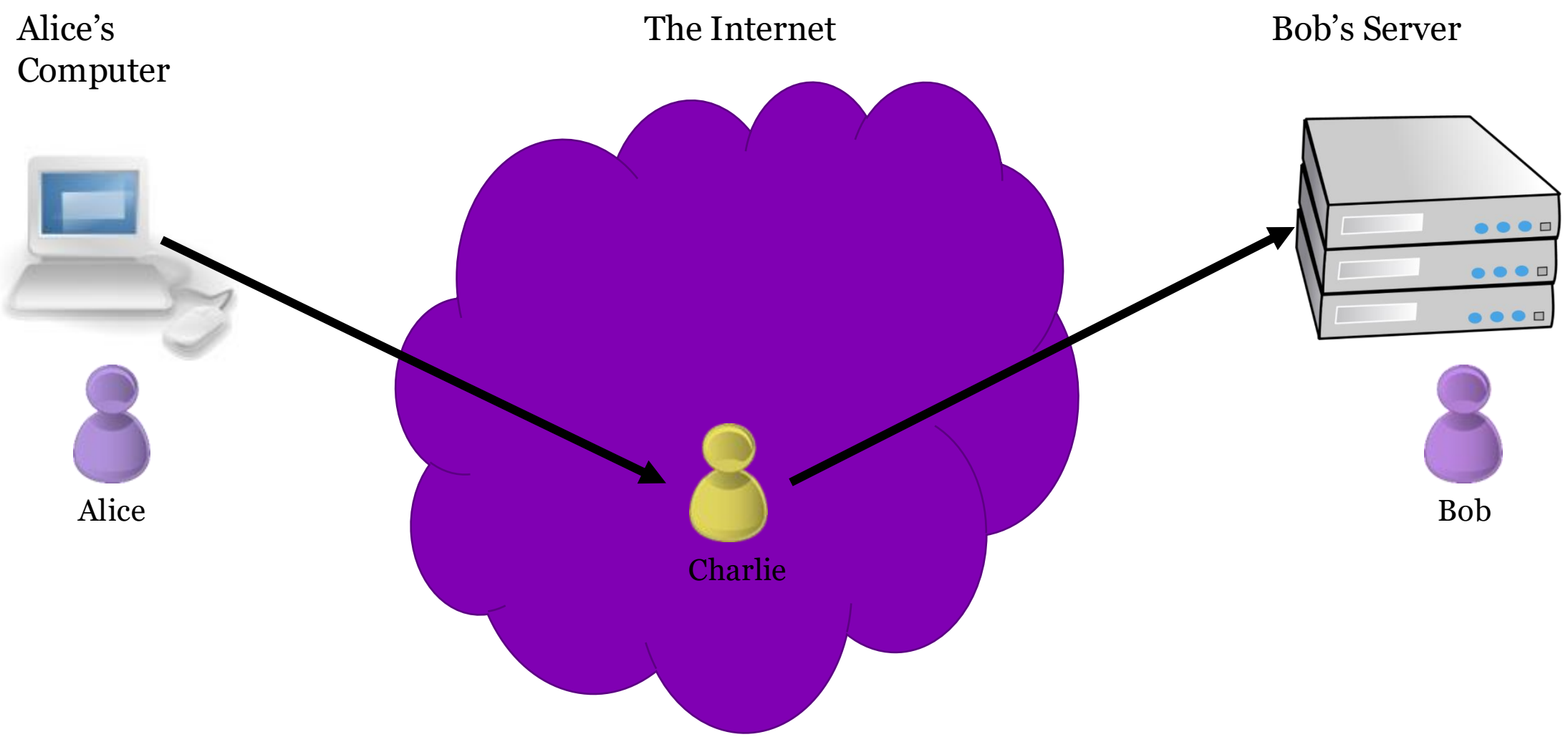

Alice
Free Wi-Fi
Accept & Connect
Google
BBC
NEWS
STARBUCKS
COFFEE

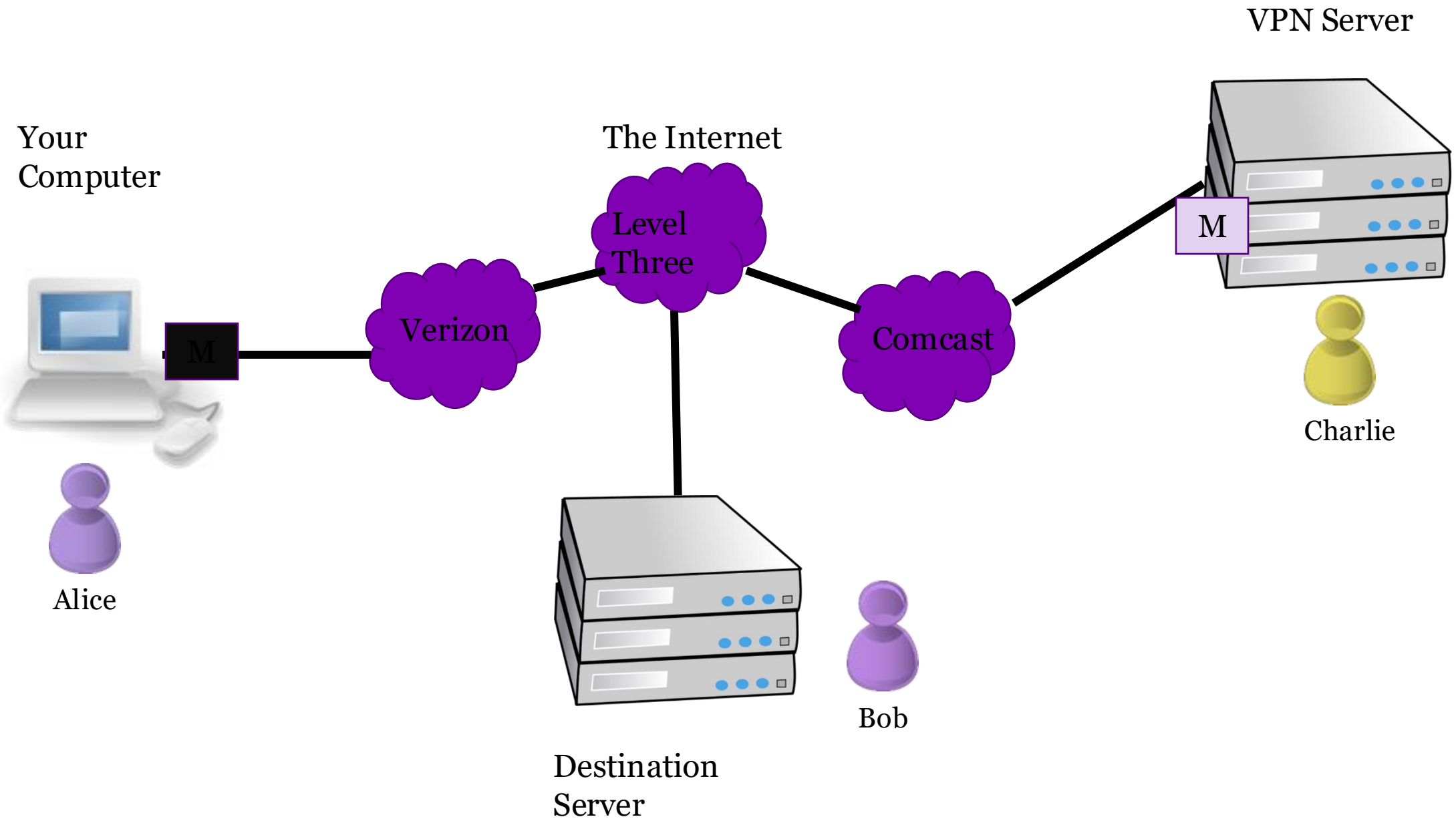

Your
Computer
The Internet
Level
Three
Verizon
Comcast
Destination
Server

Virtual Private Networks (VPNs)

Self-inflicted Man in the Middle


VPN Server
Your Computer
The Internet
Level Three
M
Verizon
Comcast
M
Destination Server


Alice's Computer
The Internet
Bob's Server
Alice
Charlie
Bob


VPN Server
Your Computer
The Internet
Level Three
M
Verizon
Comcast
M
Charlie
Alice
Bob
Destination Server

The following is an attack that actually happened to a student of mine when they were trying to upload their “set a cookie” homework using a free VPN.

```
<html>
<head>
        <title>Basic web page</title>
        <link href="http://vaniea.com/teaching/privacyToday/basic.css" rel="stylesheet" type="text/css"/>
        <script>
                document.cookie="username=John Doe;";
        </script>
</head>
<body>  THIS TEXT HAS BEEN CHANGED.
</body>
</html>
```

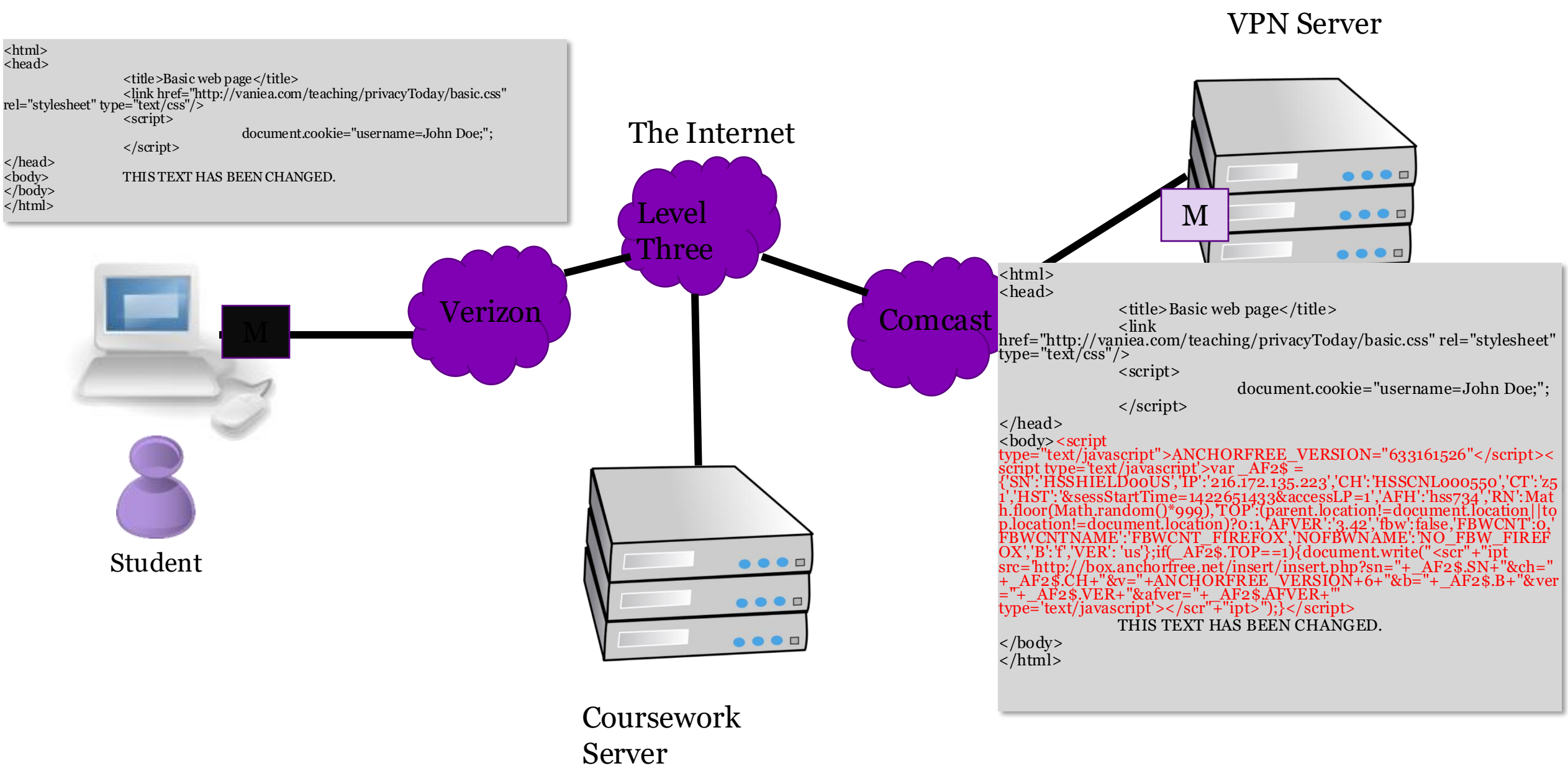
Correct Answer

Correct Answer

```
<html>
<head>
        <title>Basic web page</title>
        <link href="http://vaniea.com/teaching/privacyToday/basic.css" rel="stylesheet" type="text/css"/>
        <script>
                document.cookie="username=John Doe;";
        </script>
</head>
<body>  THIS TEXT HAS BEEN CHANGED.
</body>
</html>
```

Attacked Answer

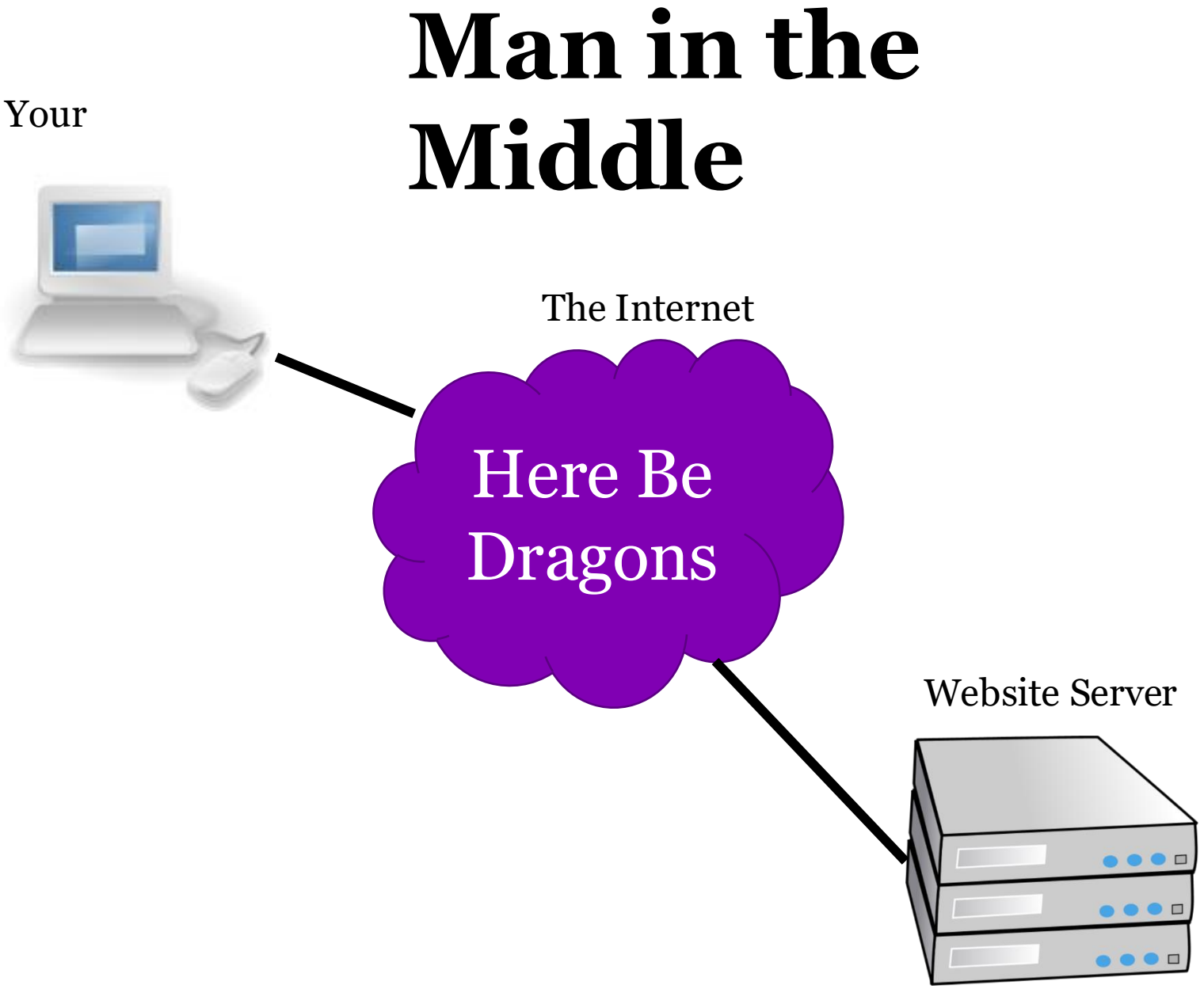
```
<html>
<head>
        <title>Basic web page</title>
        <link href="http://vaniea.com/teaching/privacyToday/basic.css" rel="stylesheet" type="text/css"/>
        <script>
                document.cookie="username=John Doe;";
        </script>
</head>
<body><script type="text/javascript">ANCHORFREE_VERSION="633161526"</script><script type='text/javascript'>var _AF2$ =
{'SN':'HSSHIELDooUS','IP':'216.172.135.223','CH':'HSSCNL00055O','CT':'z51','HST':'&sessStartTime=1422651433&accessLP=1','AFH':'hss734','RN':Math.floor(Math.random()*999),'TOP':(parent.location!=docum
ent.location||top.location!=document.location)?0:1,'AFVER':'3.42','fbw':false,'FBWCNT':0,'FBWCNTNAME':'FBWCNT_FIREFOX','NOFBWNAME':'NO_FBW_FIREFOX','B':'f','VER':
'us'};if(_AF2$.TOP==1){document.write("<scr"+"ipt
src='http://box.anchorfree.net/insert/insert.php?sn="+_AF2$.SN+"&ch="+_AF2$.CH+"&v="+ANCHORFREE_VERSION+6+"&b="+_AF2$.B+"&ver="+_AF2$.VER+"&afver="+_AF2$.AFVER+"'
type='text/javascript'></scr"+"ipt>");}</script>
        THIS TEXT HAS BEEN CHANGED.
</body>
</html>
```


VPN Server
The Internet
Level Three
Verizon
Comcast
M
M
Student
Coursework Server
<html>
<head>
<title>Basic web page</title>
<link href="http://vaniea.com/teaching/privacyToday/basic.css" rel="stylesheet" type="text/css"/>
<script>
document.cookie="username=John Doe;";
</script>
</head>
<body>
THIS TEXT HAS BEEN CHANGED.
</body>
</html>
<html>
<head>
<title>Basic web page</title>
<link href="http://vaniea.com/teaching/privacyToday/basic.css" rel="stylesheet" type="text/css"/>
<script>
document.cookie="username=John Doe;";
</script>
</head>
<body><script type="text/javascript">ANCHORFREE_VERSION="633161526"</script><script type='text/javascript'>var _AF2$ = {'SN':'HSSHIELD00US','IP':'216.172.135.223','CH':'HSSCNL000550','CT':'z51','HST':'&sessStartTime=1422651433&accessLP=1','AFH':'hss734','RN':Math.floor(Math.random()*999),'TOP':(parent.location!=document.location||top.location!=document.location)?0:1,'AFVER':'3.42','fbw':false,'FBWCNT':0,'FBWCNTNAME':'FBWCNT_FIREFOX','NOFBWNAME':'NO_FBW_FIREFOX','B':'f','VER': 'us'};if(_AF2$.TOP==1){document.write("<scr"+"ipt src='http://box.anchorfree.net/insert/insert.php?sn="+_AF2$.SN+"&ch="+_AF2$.CH+"&v="+ANCHORFREE_VERSION+6+"&b="+_AF2$.B+"&ver="+_AF2$.VER+"&afver="+_AF2$.AFVER+"' type='text/javascript'></scr"+"ipt>");}</script>
THIS TEXT HAS BEEN CHANGED.
</body>
</html>

In short:

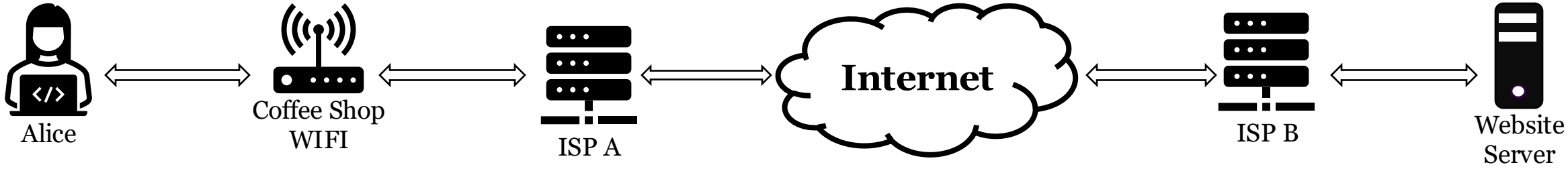
Dangerous stuff happens on the Internet, do not assume data will be safe in transit

Man in the Middle



Sample connection: Alice loads a website

Alice visits: http://example.com



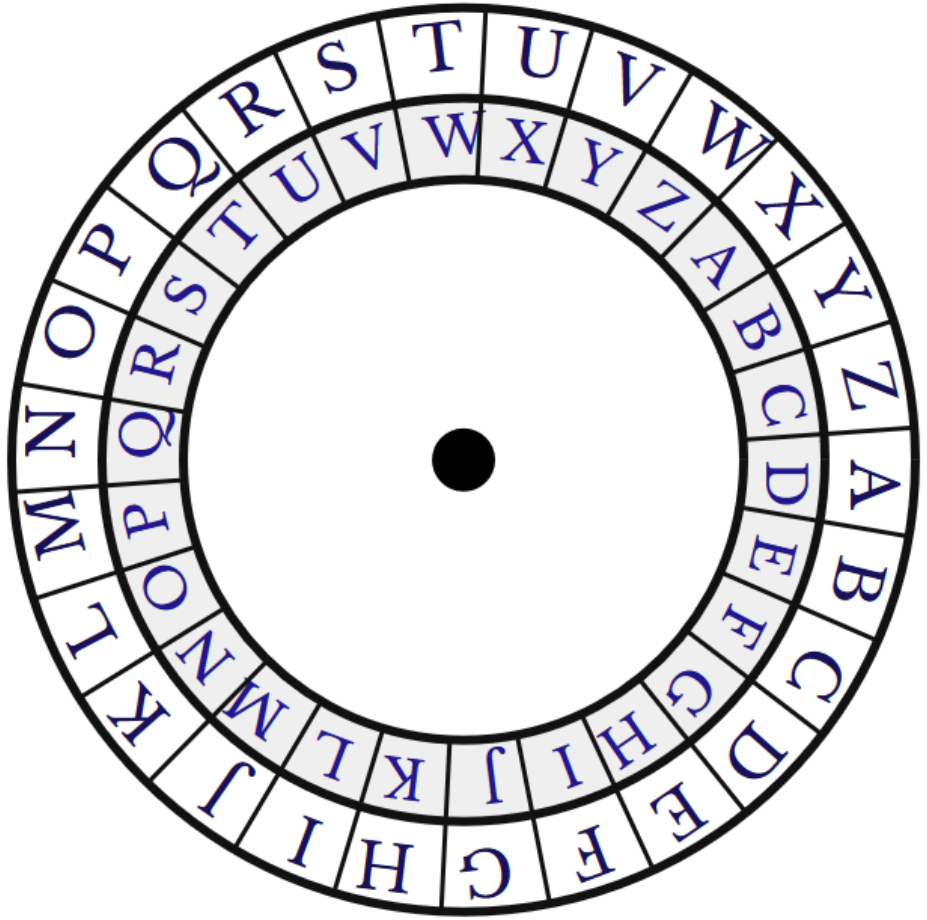
For each of the above connection points, can they learn:

1. The name and/or IP address of the website Alice is visiting
2. The content of the webpage Alice is viewing
3. Alice's Operating System (Linux, Windows, MacOS)

PUBLIC PRIVATE KEY ENCRYPTION

Encryption is based on shared secrets

- In a Caesar Cipher (right) the “secret” is how many places to turn the wheel
- Encryption is done by finding a letter on the outer wheel and then writing down the letter on the inner wheel.
- Easy to break using modern computers by just guessing all 26 possibilities.



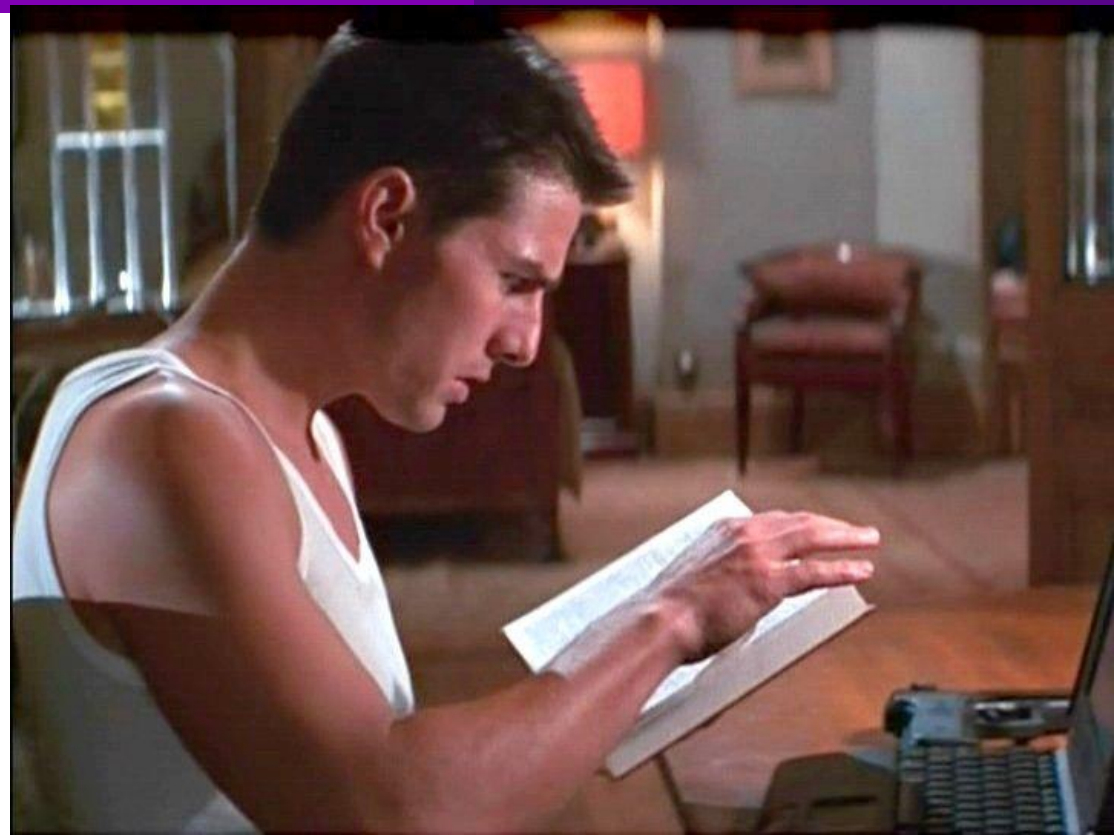
Overly simple example using Caesar Cipher with key 3:

Message	**C**	**R**	**Y**	**P**	**T**	**O**
Encrypted	F	U	V	S	E	R

https://www.xarg.org/tools/caesar-cipher/

One-time-pads

- Share a “key” in advance.
- Use the “key” to encrypt/decrypt by adding the letters.
- One of the strongest forms of encryption.
- But… you have to pre-share long keys.



Mission Impossible movie, use of hotel bible as key

Overly simple example:

Message	**C**	**R**	**Y**	**P**	**T**	**O**
Key	A	Y	S	Y	I	F
Encrypted	C	P	Q	N	B	T

Encryption properties we want:

1. The communication between you and the other party is **confidential** and has **not been changed**
   - No one can read what you sent
   - No one can change what you sent
2. **Knowing who** you are communicating with
   - You are talking to who you think you are talking to and not someone else

Public/private key cryptography

- Generate two “keys” that are paired
- **Whatever one key locks only the other key can unlock**

- Public keys are given out to everybody

- Private keys are kept private

My public key

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2

mQENBFHMcgABCAC9WrYDO6K2L3VHyi4eHN6suHLqMpJ+SO+IUTuLEVnUzIoXAUXH
KozHejfV/9XoG8j933ZtszXKCog3aMESe0E0z6fNGfolvaCe5B4jwq0Jt8NHwb5L
B2dnq0CplgXcN2GJxfEHHUaf27COSobCJxPMeshUh4ZHke+g6DatmiEtBpVp41Ot
1zgxdMQkgb2H2xw28RYfYkdDoueteIkOrFLrCy9ZF9KdMhA1eBH94KnwIQshdiZR
QYEX25+M8cKCb++Rc9H6an7EG9WHOFRW40UsY52OfveOyfQPzkkRto7u2339hvHo
B/h+7xLM6FQbOUZQ9BD5w7IQHgYtXJVsUjodABEBAAG0IkthbWkgVmFuaWVhIDxr
dmFuaWVhQGluZi5lZC5hYy51az6JAT8EEwEIACkFAlYKYvECGyMFCQlmAYAHCwkI
BwMCAQYVCAIJCgsEFgIDAQIeAQIXgAAKCRCTdsxl9/HZffG+CACShuKxje3QAqew
GWh8K4gCdiY0xDqJwq3PHxmyhZmQeN/1a1KcOrIjI2b+Q75/5t+EgXOHpR0PIxfG
lZ6zOEpf6A18iFXx3JgQZdwPD0jtBiWNpOyMeBGTgIvEYG3so2VueQoeXcq3dbYp
5vstVxtD+TKHQ5CioIT75P2bzYq/XLT5aIbNQhQDPcTo0DgbRH+FvqsRXr7yeaef
JaPnxX0+1L33t2QY9zctiGyebwrvHMrIPBJ2VYCDzQkJ7uQ5eFh4ZhsMgOmzLQD4
YiGr5weIMFwAvxZOaRxEa9Vf48jiWvrxuJ8YfHWS0hEScNOcYC2P8q20lJwwE26T
lpdtrwCqtB1LYW1pIFZhbmllYSA8a2FtaUB2YW5pZWEuY29tPokBQgQTAQIALAIb
IwUJCWYBgAcLCQgHAwIBBhUIAgkKCwQWAgMBAh4BAheABQJWCmMeAhkBAAoJEJN2
zGX38dl9JJAIAIW0rxrlYsrmKS6CbW8MgTxxTDOXaCt1b7F0W0QZHskIUQhEcE+a
XBYib1A5uHaatLfyjeXaD3qMEoZnQHoYMGE0GKu00wWsbhfoQzHPgwzRLkD1i75M
BIbaww0KW0VB9e4AkMakXJCnF5BXe06AHRL2v15V205DikVnlCRX0cKtu8b7LnkM
cLn70L0br1de1uyKoNzbSnO/vpKDJp0/EY5yUeV90IypZy/6wFQBehg1sXye6zn0
9wb9uUsu9+/P8pz4JILMDSevjfT7zSRSl/YP3fOfZ6N4bc+KOdwPM7u5Iyoeu9zh
pzibv3ge7VhH2xIWz8vYZ/2xT1345tWRRMOJAhwEEwECAAYFAlTnSpEACgkQjyxM
p99tBt2B8A/+OpIzOsQbQJB8yxti4I7PpD1weJDf3a81Vhm7JyXE/Xy66ypfdt3w
XmFRUuIrwezY1NebWNCRQHzQvRv/VJwjbTUx+Q3HsjIkKlHbE7iCiQXXtTRk0Eny
2nudcjGI2v03C3B2JCucEw6esF1x79PI/lPv2+6tgUBKmDfOpsB2vbtqrHnmAYKL
4lQBFH1YSJgnzwo2Jkh0hcHdF90Zem1eMeiDEeVkH63893N8Swk5fBKdTj+SKZ/L
rQElBBlpMR9BmeY6bPvWRuycVKonIMR80G9iFABxjTpWBL8aGk6EeVK5EqYDGvkd
ZIarK84r+KU1KD5IfgOCN7nhwgy7VImE68caZHSRiPWZP1fVVMhydiRJv8WsoUs6
INfVU3nxH+ZYthPbY0T86leGSchBT5K/fBQvbjhrRTbTFwvjzSifb9efWylDi994
nzP6cNorir3GIpsT8gPgBB2/NjxaWiM6y3X1az1vRnsunQHuyKkFWPZwnEvDJYaC
NN/3jWcbhLFwKBDsaHps2+1meFP0oJFvNetzp2bjT9a9pXaQ6KhOmo5DnhLcaV97
bFBpsUuBGaYZTSS05x1RdXHqpEbgap8dtuHhVvJw9QYDQBJro K4aKyG9qqMD8cta
Pl/FAdyAqwH8Nw9efqAK+RQxSVUaue9BYEnbIRpsDK6MkP3YMFmu5ki5AQ0EUcxy
AAEIALyXYy8G2ZaTDJpdGcRhmIqOOSUlzPV7/5E5BbYKBNu4KU3nX+JLVcF5jxPQ
42c7i/WRVxE1BJTiarKGsEvCi94TTXSIUKAt3T10GBtXmGvqbGBq8ljSGl1UTwdF
5yu50JyRSf2fqRND6P/2eHNXejDUtdvhUXIUt8h9MuUO/ipD0DnwIvMnAATJHA+R
Zqw60NpyjRGzvr3iuWUwe4PtyJDI3ELAFkbp/NAc5TIuVHRHNOWNplcIJhM5zHuB
QQb3G/EsCn2PQZ5w5SDzavF2SpvQfDqxYpDaTLAXtF+wsJL5iaUjxwRgJPOdbCZf
2Tozd7h9MXtGJDlPKJ8eLG80gcMAEQEAAYkBJQQYAQIADwUCUcxyAAIbDAUJCWYB
gAAKCRCTdsxl9/HZfS+hB/9BJqSmIgcoHFXnb1PVIKxekzL8+WVm5Pk/EgMQSLZ2
HX4p3ial5PEPcYgUw9YnaG4i00dwJGw5/daTWRrTzcnKd8Yq0P+DU0t96HZDSu3m
mCzE9NVAQYboFbVmGOx0eo627UBSvFqaXvAxBDYkoR8B0TnKhrQFwXkZVb30hKwD
TgAFjOGlZiE6uAdST231tFaq0bizYfe5AVXRqro20xBqNbaJNqs3SW0D831Syvdv
llOBx83/R0gg7hUkI6F2vzXicWmUwFSXRrggCSbLosHsP6isBWwvlHeRmna/aQab
YKG3gbV9iyczAS31gbogVLAZqNSWhp8vVIEE28Fyf/Ed
=x5FK
-----END PGP PUBLIC KEY BLOCK-----

I want to send Bob a message that no one else can read

- I encrypt (lock) the message with Bob's public key.
- Only Bob has his private key, so only Bob can decrypt (unlock) the message.

Bob's public key

Bob's private key

My public key

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2

mQENBFHMcgABCAC9WrYDO6K2L3VHyi4eHN6suHLqMpJ+SO+IUTuLEVnUzIoXAUXH
KozHejfV/9XoG8j933ZtszXKCog3aMESe0E0z6fNGfolvaCe5B4jwq0Jt8NHwb5L
B2dnq0CplgXcN2GJxfEHHUaf27COSobCJxPMeshUh4ZHke+g6DatmiEtBpVp41Ot
1zgxdMQkgb2H2xw28RYfYkdDoueteIkOrFLrCy9ZF9KdMhA1eBH94KnwIQshdiZR
QYEX25+M8cKCb++Rc9H6an7EG9WHOFRW40UsY52OfveOyfQPzkkRto7u2339hvHo
B/h+7xLM6FQbOUZQ9BD5w7IQHgYtXJVsUjodABEBAAG0IkthbWkgVmFuaWVhIDxr
dmFuaWVhQGluZi5lZC5hYy51az6JAT8EEwEIACkFAlYKYvECGyMFCQlmAYAHCwkI
BwMCAQYVCAIJCgsEFgIDAQIeAQIXgAAKCRCTdsxl9/HZffG+CACShuKxje3QAqew
GWh8K4gCdiYoxDqJwq3PHxmyhZmQeN/1a1KcOrIjI2b+Q75/5t+EgXOHpR0PIxfG
lZ6zOEpf6A18iFXx3JgQZdwPD0jtBiWNpOyMeBGTgIvEYG3so2VueQoeXcq3dbYp
5vstVxtD+TKHQ5CioIT75P2bzYq/XLT5aIbNQhQDPcTo0DgbRH+FvqsRXr7yeaef
JaPnxX0+1L33t2QY9zctiGyebwrvHMrIPBJ2VYCDzQkJ7uQ5eFh4ZhsMgOmzLQD4
YiGr5weIMFwAvxZOaRxEa9Vf48jiWvrxuJ8YfHWS0hEScNOcYC2P8q20lJwwE26T
lpdtrwCqtB1LYW1pIFZhbmllYSA8a2FtaUB2YW5pZWEuY29tPokBQgQTAQIALAIb
IwUJCWYBgAcLCQgHAwIBBhUIAgkKCwQWAgMBAh4BAheABQJWCmMeAhkBAAoJEJN2
zGX38dl9JJAIAIW0rxrlYsrmKS6CbW8MgTxxTDOXaCt1b7F0W0QZHskIUQhEcE+a
XBYib1A5uHaatLfyjeXaD3qMEoZnQHoYMGE0GKu00wWsbhfoQzHPgwzRLkD1i75M
BIbaww0KWoVB9e4AkMakXJCnF5BXeo6AHRL2v15V205DikVnlCRXocKtu8b7LnkM
cLn70L0br1de1uyKoNzbSnO/vpKDJp0/EY5yUeV90IypZy/6wFQBehg1sXye6zn0
9wb9uUsu9+/P8pz4JILMDSevjfT7zSRSl/YP3fOfZ6N4bc+KOdwPM7u5Iyoeu9zh
pzibv3ge7VhH2xIWz8vYZ/2xT1345tWRRMOJAhwEEwECAAYFAlTnSpEACgkQjyxM
p99tBt2B8A/+OpIzOsQbQJB8yxti4I7PpD1weJDf3a81Vhm7JyXE/Xy66ypfdt3w
XmFRUuIrwezY1NebWNCRQHzQvRv/VJwjbTUx+Q3HsjIkKlHbE7iCiQXXtTRkoEny
2nudcjGI2v03C3B2JCucEw6esF1x79PI/lPv2+6tgUBKmDfOpsB2vbtqrHnmAYKL
4lQBFH1YSJgnzwo2Jkh0hcHdF90Zem1eMeiDEeVkH63893N8Swk5fBKdTj+SKZ/L
rQElBBlpMR9BmeY6bPvWRuycVKonIMR80G9iFABxjTpWBL8aGk6EeVK5EqYDGvkd
ZIarK84r+KU1KD5IfgOCN7nhwgy7VImE68caZHSRiPWZP1fVVMhydiRJv8WsoUs6
INfVU3nxH+ZYthPbY0T86leGSchBT5K/fBQvbjhrRTbTFwvjzSifb9efWylDi994
nzP6cNorir3GIpsT8gPgBB2/NjxaWiM6y3X1az1vRnsunQHuyKkFWPZwnEvDJYaC
NN/3jWcbhLFwKBDsaHps2+1meFP0oJFvNetzp2bjT9a9pXaQ6KhOmo5DnhLcaV97
bFBpsUuBGaYZTSS05x1RdXHqpEbgap8dtuHhVvJw9QYDQBJr0K4aKyG9qqMD8cta
Pl/FAdyAqwH8Nw9efqAK+RQxSVUaue9BYEnbIRpsDK6MkP3YMFmu5ki5AQ0EUcxy
AAEIALyXYy8G2ZaTDJpdGcRhmIqOOSUlzPV7/5E5BbYKBNu4KU3nX+JLVcF5jxPQ
42c7i/WRVxE1BJTiarKGsEvCi94TTXSIUKAt3T1oGBtXmGvqbGBq8ljSGl1UTwdF
5yu50JyRSf2fqRND6P/2eHNXejDUtdvhUXIUt8h9MuUO/ipD0DnwIvMnAATJHA+R
Zqw6oNpyjRGzvr3iuWUwe4PtyJDI3ELAFkbp/NAc5TIuVHRHNOWNplcIJhM5zHuB
QQb3G/EsCn2PQZ5w5SDzavF2SpvQfDqxYpDaTLAXtF+wsJL5iaUjxwRgJPOdbCZf
2Tozd7h9MXtGJDlPKJ8eLG8ogcMAEQEAAYkBJQQYAQIADwUCUcxyAAIbDAUJCWYB
gAAKCRCTdsxl9/HZfS+hB/9BJqSmIgcoHFXnb1PVIKxekzL8+WVm5Pk/EgMQSLZ2
HX4p3ial5PEPcYgUw9YnaG4i00dwJGw5/daTWRrTzcnKd8Yq0P+DU0t96HZDSu3m
mCzE9NVAQYboFbVmGOx0eo627UBSvFqaXvAxBDYkoR8BoTnKhrQFwXkZVb30hKwD
TgAFjOGlZiE6uAdST231tFaq0bizYfe5AVXRqro20xBqNbaJNqs3SW0D831Syvdv
llOBx83/R0gg7hUkI6F2vzXicWmUwFSXRrggCSbLosHsP6isBWwvlHeRmna/aQab
YKG3gbV9iyczAS31gbogVLAZqNSWhp8vVIEE28Fyf/Ed
=x5FK
-----END PGP PUBLIC KEY BLOCK-----

I want to send Bob a message that no one else can read

- I encrypt (lock) the message with Bob's public key.
- Only Bob has his private key, so only Bob can decrypt (unlock) the message.
- Using the same key twice just creates an error (meaningless output)

My public key

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2

mQENBFHMcgABCAC9WrYDO6K2L3VHyi4eHN6suHLqMpJ+SO+IUTuLEVnUzIoXAUXH
KozHejfV/9XoG8j933ZtszXKCog3aMESe0E0z6fNGfolvaCe5B4jwqoJt8NHwb5L
B2dnq0CplgXcN2GJxfEHHUaf27COSobCJxPMeshUh4ZHke+g6DatmiEtBpVp41Ot
1zgxdMQkgb2H2xw28RYfYkdDoueteIkOrFLrCy9ZF9KdMhA1eBH94KnwIQshdiZR
QYEX25+M8cKCb++Rc9H6an7EG9WHOFRW40UsY52OfveOyfQPzkkRto7u2339hvHo
B/h+7xLM6FQbOUZQ9BD5w7IQHgYtXJVsUjodABEBAAG0IkthbWkgVmFuaWVhIDxr
dmFuaWVhQGluZi5lZC5hYy51az6JAT8EEwEIACkFAlYKYvECGyMFCQlmAYAHCwkI
BwMCAQYVCAIJCgsEFgIDAQIeAQIXgAAKCRCTdsxl9/HZffG+CACShuKxje3QAqew
GWh8K4gCdiYoxDqJwq3PHxmyhZmQeN/1a1KcOrIjI2b+Q75/5t+EgXOHpR0PIxfG
lZ6zOEpf6A18iFXx3JgQZdwPD0jtBiWNpOyMeBGTgIvEYG3so2VueQoeXcq3dbYp
5vstVxtD+TKHQ5CioIT75P2bzYq/XLT5aIbNQhQDPcTo0DgbRH+FvqsRXr7yeaef
JaPnxX0+1L33t2QY9zctiGyebwrvHMrIPBJ2VYCDzQkJ7uQ5eFh4ZhsMgOmzLQD4
YiGr5weIMFwAvxZOaRxEa9Vf48jiWvrxuJ8YfHWS0hEScNOcYC2P8q20lJwwE26T
lpdtrwCqtB1LYW1pIFZhbmllYSA8a2FtaUB2YW5pZWEuY29tPokBQgQTAQIALAIb
IwUJCWYBgAcLCQgHAwIBBhUIAgkKCwQWAgMBAh4BAheABQJWCmMeAhkBAAoJEJN2
zGX38dl9JJAIAIW0rxrlYsrmKS6CbW8MgTxxTDOXaCt1b7F0W0QZHskIUQhEcE+a
XBYib1A5uHaatLfyjeXaD3qMEoZnQHoYMGE0GKu00wWsbhfoQzHPgwzRLkD1i75M
BIbaww0KWoVB9e4AkMakXJCnF5BXeo6AHRL2v15V205DikVnlCRXocKtu8b7LnkM
cLn70L0br1de1uyKoNzbSnO/vpKDJp0/EY5yUeV90IypZy/6wFQBehg1sXye6znO
9wb9uUsu9+/P8pz4JILMDSevjfT7zSRSl/YP3fOfZ6N4bc+KOdwPM7u5Iyoeu9zh
pzibv3ge7VhH2xIWz8vYZ/2xT1345tWRRMOJAhwEEwECAAYFAlTnSpEACgkQjyxM
p99tBt2B8A/+OpIzOsQbQJB8yxti4I7PpD1weJDf3a81Vhm7JyXE/Xy66ypfdt3w
XmFRUuIrwezY1NebWNCRQHzQvRv/VJwjbTUx+Q3HsjIkKlHbE7iCiQXXtTRkoEny
2nudcjGI2v03C3B2JCucEw6esF1x79PI/lPv2+6tgUBKmDfOpsB2vbtqrHnmAYKL
4lQBFH1YSJgnzwo2Jkh0hcHdF90Zem1eMeiDEeVkH63893N8Swk5fBKdTj+SKZ/L
rQElBBlpMR9BmeY6bPvWRuycVKonIMR80G9iFABxjTpWBL8aGk6EeVK5EqYDGvkd
ZIarK84r+KU1KD5IfgOCN7nhwgy7VImE68caZHSRiPWZP1fVVMhydiRJv8WsoUs6
INfVU3nxH+ZYthPbY0T86leGSchBT5K/fBQvbjhrRTbTFwvjzSifb9efWylDi994
nzP6cNorir3GIpsT8gPgBB2/NjxaWiM6y3X1az1vRnsunQHuyKkFWPZwnEvDJYaC
NN/3jWcbhLFwKBDsaHps2+1meFP0oJFvNetzp2bjT9a9pXaQ6KhOmo5DnhLcaV97
bFBpsUuBGaYZTSS05x1RdXHqpEbgap8dtuHhVvJw9QYDQBJro K4aKyG9qqMD8cta
Pl/FAdyAqwH8Nw9efqAK+RQxSVUaue9BYEnbIRpsDK6MkP3YMFmu5ki5AQ0EUcxy
AAEIALyXYy8G2ZaTDJpdGcRhmIqOOSUlzPV7/5E5BbYKBNu4KU3nX+JLVcF5jxPQ
42c7i/WRVxE1BJTiarKGsEvCi94TTXSIUKAt3T10GBtXmGvqbGBq8ljSGl1UTwdF
5yu50JyRSf2fqRND6P/2eHNXejDUtdvhUXIUt8h9MuUO/ipD0DnwIvMnAATJHA+R
Zqw6oNpyjRGzvr3iuWUwe4PtyJDI3ELAFkbp/NAc5TIuVHRHNOWNplcIJhM5zHuB
QQb3G/EsCn2PQZ5w5SDzavF2SpvQfDqxYpDaTLAXtF+wsJL5iaUjxwRgJPOdbCZf
2Tozd7h9MXtGJDlPKJ8eLG8ogcMAEQEAAYkBJQQYAQIADwUCUcxyAAIbDAUJCWYB
gAAKCRCTdsxl9/HZfS+hB/9BJqSmIgcoHFXnb1PVIKxekzL8+WVm5Pk/EgMQSLZ2
HX4p3ial5PEPcYgUw9YnaG4i00dwJGw5/daTWRrTzcnKd8Yq0P+DU0t96HZDSu3m
mCzE9NVAQYboFbVmGOx0eo627UBSvFqaXvAxBDYkoR8BoTnKhrQFwXkZVb30hKwD
TgAFjOGlZiE6uAdST231tFaq0bizYfe5AVXRqro20xBqNbaJNqs3SW0D831Syvdv
llOBx83/R0gg7hUkI6F2vzXicWmUwFSXRrggCSbLosHsP6isBWwvlHeRmna/aQab
YKG3gbV9iyczAS31gbogVLAZqNSWhp8vVIEE28Fyf/Ed
=x5FK
-----END PGP PUBLIC KEY BLOCK-----

I want to send Bob a message that no one else can read

- I encrypt (lock) the message with Bob's public key.
- Only Bob has his private key, so only Bob can decrypt (unlock) the message.

Bob's public key

Bob's private key

My public key

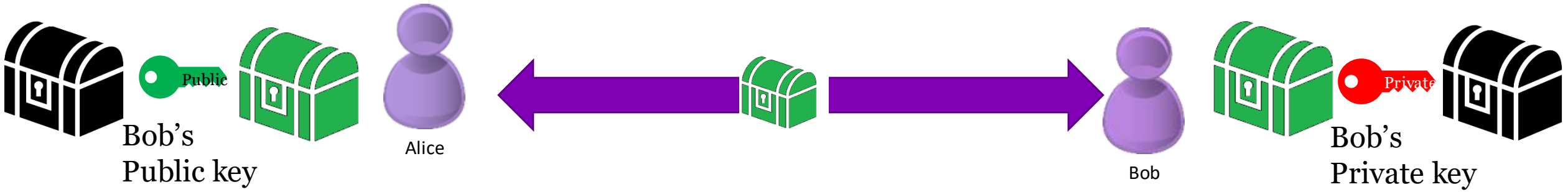
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2

mQENBFHMcgABCAC9WrYDO6K2L3VHyi4eHN6suHLqMpJ+SO+IUTuLEVnUzIoXAUXH
KozHejfV/9XoG8j933ZtszXKCog3aMESe0E0z6fNGfolvaCe5B4jwq0Jt8NHwb5L
B2dnq0CplgXcN2GJxfEHHUaf27COSobCJxPMeshUh4ZHke+g6DatmiEtBpVp41Ot
1zgxdMQkgb2H2xw28RYfYkdDoueteIkOrFLrCy9ZF9KdMhA1eBH94KnwIQshdiZR
QYEX25+M8cKCb++Rc9H6an7EG9WHOFRW40UsY52OfveOyfQPzkkRto7u2339hvHo
B/h+7xLM6FQbOUZQ9BD5w7IQHgYtXJVsUjodABEBAAG0IkthbWkgVmFuaWVhIDxr
dmFuaWVhQGluZi5lZC5hYy51az6JAT8EEwEIACkFAlYKYvECGyMFCQlmAYAHCwkI
BwMCAQYVCAIJCgsEFgIDAQIeAQIXgAAKCRCTdsxl9/HZffG+CACShuKxje3QAqew
GWh8K4gCdiYoxDqJwq3PHxmyhZmQeN/1a1KcOrIjI2b+Q75/5t+EgXOHpR0PIxfG
lZ6zOEpf6A18iFXx3JgQZdwPD0jtBiWNpOyMeBGTgIvEYG3so2VueQoeXcq3dbYp
5vstVxtD+TKHQ5CioIT75P2bzYq/XLT5aIbNQhQDPcTo0DgbRH+FvqsRXr7yeaef
JaPnxX0+1L33t2QY9zctiGyebwrvHMrIPBJ2VYCDzQkJ7uQ5eFh4ZhsMgOmzLQD4
YiGr5weIMFwAvxZOaRxEa9Vf48jiWvrxuJ8YfHWS0hEScNOcYC2P8q20lJwwE26T
lpdtrwCqtB1LYW1pIFZhbmllYSA8a2FtaUB2YW5pZWEuY29tPokBQgQTAQIALAIb
IwUJCWYBgAcLCQgHAwIBBhUIAgkKCwQWAgMBAh4BAheABQJWCmMeAhkBAAoJEJN2
zGX38dl9JJAIAIW0rxrlYsrmKS6CbW8MgTxxTDOXaCt1b7F0W0QZHskIUQhEcE+a
XBYib1A5uHaatLfyjeXaD3qMEoZnQHoYMGE0GKu00wWsbhfoQzHPgwzRLkD1i75M
BIbaww0KWoVB9e4AkMakXJCnF5BXeo6AHRL2v15V205DikVnlCRXocKtu8b7LnkM
cLn70Lobr1de1uyKoNzbSnO/vpKDJp0/EY5yUeV90IypZy/6wFQBehg1sXye6zn0
9wb9uUsu9+/P8pz4JILMDSevjfT7zSRSl/YP3fOfZ6N4bc+KOdwPM7u5Iyoeu9zh
pzibv3ge7VhH2xIWz8vYZ/2xT1345tWRRMOJAhwEEwECAAYFAlTnSpEACgkQjyxM
p99tBt2B8A/+OpIzOsQbQJB8yxti4I7PpD1weJDf3a81Vhm7JyXE/Xy66ypfdt3w
XmFRUuIrwezY1NebWNCRQHzQvRv/VJwjbTUx+Q3HsjIkKlHbE7iCiQXXtTRkoEny
2nudcjGI2v03C3B2JCucEw6esF1x79PI/lPv2+6tgUBKmDfOpsB2vbtqrHnmAYKL
4lQBFH1YSJgnzwo2Jkh0hcHdF90Zem1eMeiDEeVkH63893N8Swk5fBKdTj+SKZ/L
rQElBBlpMR9BmeY6bPvWRuycVKonIMR80G9iFABxjTpWBL8aGk6EeVK5EqYDGvkd
ZIarK84r+KU1KD5IfgOCN7nhwgy7VImE68caZHSRiPWZP1fVVMhydiRJv8WsoUs6
INfVU3nxH+ZYthPbY0T86leGSchBT5K/fBQvbjhrRTbTFwvjzSifb9efWylDi994
nzP6cNorir3GIpsT8gPgBB2/NjxaWiM6y3X1az1vRnsunQHuyKkFWPZwnEvDJYaC
NN/3jWcbhLFwKBDsaHps2+1meFP0oJFvNetzp2bjT9a9pXaQ6KhOmo5DnhLcaV97
bFBpsUuBGaYZTSS05x1RdXHqpEbgap8dtuHhVvJw9QYDQBJr0K4aKyG9qqMD8cta
Pl/FAdyAqwH8Nw9efqAK+RQxSVUaue9BYEnbIRpsDK6MkP3YMFmu5ki5AQ0EUcxy
AAEIALyXYy8G2ZaTDJpdGcRhmIqOOSUlzPV7/5E5BbYKBNu4KU3nX+JLVcF5jxPQ
42c7i/WRVxE1BJTiarKGsEvCi94TTXSIUKAt3T1oGBtXmGvqbGBq8ljSGl1UTwdF
5yu50JyRSf2fqRND6P/2eHNXejDUtdvhUXIUt8h9MuUO/ipD0DnwIvMnAATJHA+R
Zqw6oNpyjRGzvr3iuWUwe4PtyJDI3ELAFkbp/NAc5TIuVHRHNOWNplcIJhM5zHuB
QQb3G/EsCn2PQZ5w5SDzavF2SpvQfDqxYpDaTLAXtF+wsJL5iaUjxwRgJPOdbCZf
2Tozd7h9MXtGJDlPKJ8eLG8ogcMAEQEAAYkBJQQYAQIADwUCUcxyAAIbDAUJCWYB
gAAKCRCTdsxl9/HZfS+hB/9BJqSmIgcoHFXnb1PVIKxekzL8+WVm5Pk/EgMQSLZ2
HX4p3ial5PEPcYgUw9YnaG4i00dwJGw5/daTWRrTzcnKd8Yq0P+DU0t96HZDSu3m
mCzE9NVAQYboFbVmGOx0eo627UBSvFqaXvAxBDYkoR8BoTnKhrQFwXkZVb30hKwD
TgAFjOGlZiE6uAdST231tFaq0bizYfe5AVXRqro20xBqNbaJNqs3SW0D831Syvdv
llOBx83/R0gg7hUkI6F2vzXicWmUwFSXRrggCSbLosHsP6isBWwvlHeRmna/aQab
YKG3gbV9iyczAS31gbogVLAZqNSWhp8vVIEE28Fyf/Ed
=x5FK
-----END PGP PUBLIC KEY BLOCK-----

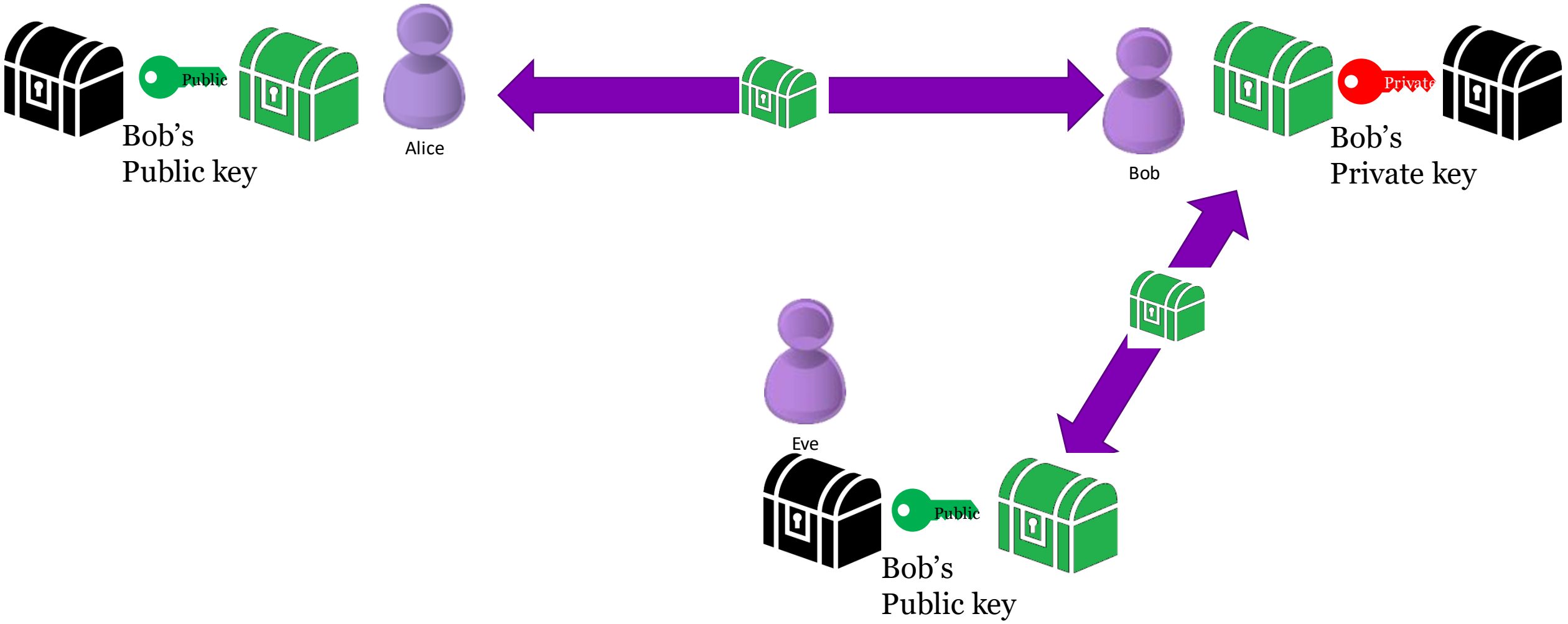
Encryption properties we want:

1. The communication between you and the other party is **confidential** and has **not been changed (integral)**
   - No one can read what you sent
   - No one can change what you sent
2. **Knowing who** you are communicating with
   - You are talking to who you think you are talking to and not someone else

Bob can’t tell who the message is from



Bob can’t tell who the message is from



I want to prove a message is from me

- I encrypt (lock) the message with my private key
- Anyone with the public key can use it to decrypt (unlock) the file. If it decrypts (unlocks), then it must have been encrypted (locked) by my private key and no other.

My private key

My public key

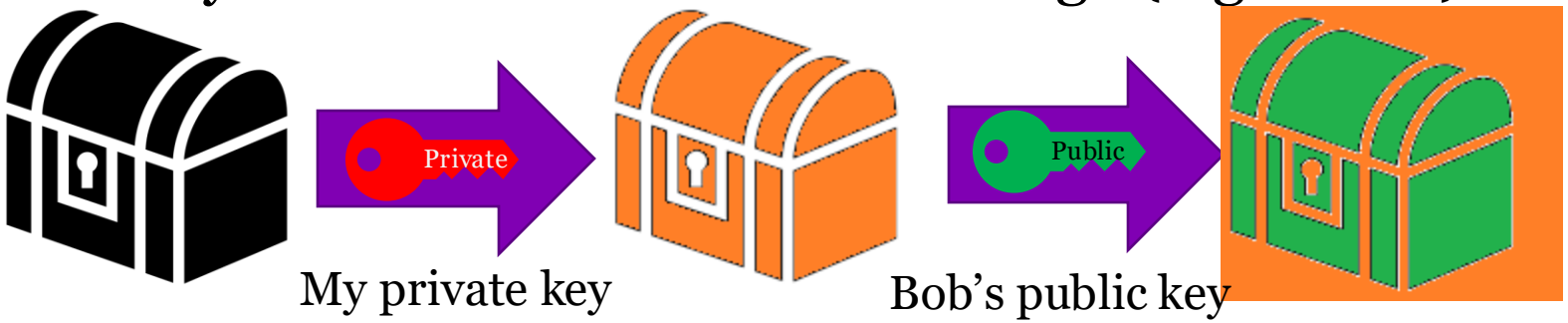
My public key

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2

mQENBFHMcgABCAC9WrYDO6K2L3VHyi4eHN6suHLqMpJ+SO+IUTuLEVnUzIoXAUXH
KozHejfV/9XoG8j933ZtszXKCog3aMESe0E0z6fNGfolvaCe5B4jwq0Jt8NHwb5L
B2dnq0CplgXcN2GJxfEHHUaf27COSobCJxPMeshUh4ZHke+g6DatmiEtBpVp41Ot
1zgxdMQkgb2H2xw28RYfYkdDoueteIkOrFLrCy9ZF9KdMhA1eBH94KnwIQshdiZR
QYEX25+M8cKCb++Rc9H6an7EG9WHOFRW40UsY52OfveOyfQPzkkRto7u2339hvHo
B/h+7xLM6FQbOUZQ9BD5w7IQHgYtXJVsUjodABEBAAG0IkthbWkgVmFuaWVhIDxr
dmFuaWVhQGluZi5lZC5hYy51az6JAT8EEwEIACkFAlYKYvECGyMFCQlmAYAHCwkI
BwMCAQYVCAIJCgsEFgIDAQIeAQIXgAAKCRCTdsxl9/HZffG+CACShuKxje3QAqew
GWh8K4gCdiYoxDqJwq3PHxmyhZmQeN/1a1KcOrIjI2b+Q75/5t+EgXOHpR0PIxfG
lZ6zOEpf6A18iFXx3JgQZdwPD0jtBiWNpOyMeBGTgIvEYG3so2VueQoeXcq3dbYp
5vstVxtD+TKHQ5CioIT75P2bzYq/XLT5aIbNQhQDPcTo0DgbRH+FvqsRXr7yeaef
JaPnxX0+1L33t2QY9zctiGyebwrvHMrIPBJ2VYCDzQkJ7uQ5eFh4ZhsMgOmzLQD4
YiGr5weIMFwAvxZOaRxEa9Vf48jiWvrxuJ8YfHWS0hEScNOcYC2P8q20lJwwE26T
lpdtrwCqtB1LYW1pIFZhbmllYSA8a2FtaUB2YW5pZWEuY29tPokBQgQTAQIALAIb
IwUJCWYBgAcLCQgHAwIBBhUIAgkKCwQWAgMBAh4BAheABQJWCmMeAhkBAAoJEJN2
zGX38dl9JJAIAIW0rxrlYsrmKS6CbW8MgTxxTDOXaCt1b7F0W0QZHskIUQhEcE+a
XBYib1A5uHaatLfyjeXaD3qMEoZnQHoYMGE0GKu00wWsbhfoQzHPgwzRLkD1i75M
BIbaww0KWoVB9e4AkMakXJCnF5BXeo6AHRL2v15V205DikVnlCRXocKtu8b7LnkM
cLn70L0br1de1uyKoNzbSnO/vpKDJp0/EY5yUeV90IypZy/6wFQBehg1sXye6zmO
9wb9uUsu9+/P8pz4JILMDSevjfT7zSRSl/YP3fOfZ6N4bc+KOdwPM7u5Iyoeu9zh
pzibv3ge7VhH2xIWz8vYZ/2xT1345tWRRMOJAhwEEwECAAYFAlTnSpEACgkQjyxM
p99tBt2B8A/+OpIzOsQbQJB8yxti4I7PpD1weJDf3a81Vhm7JyXE/Xy66ypfdt3w
XmFRUuIrwezY1NebWNCRQHzQvRv/VJwjbTUx+Q3HsjIkKlHbE7iCiQXXtTRkoEny
2nudcjGI2v03C3B2JCucEw6esF1x79PI/lPv2+6tgUBKmDfOpsB2vbtqrHnmAYKL
4lQBFH1YSJgnzwo2Jkh0hcHdF90Zem1eMeiDEeVkH63893N8Swk5fBKdTj+SKZ/L
rQElBBlpMR9BmeY6bPvWRuycVKonIMR80G9iFABxjTpWBL8aGk6EeVK5EqYDGvkd
ZIarK84r+KU1KD5IfgOCN7nhwgy7VImE68caZHSRiPWZP1fVVMhydiRJv8WsoUs6
INfVU3nxH+ZYthPbY0T86leGSchBT5K/fBQvbjhrRTbTFwvjzSifb9efWylDi994
nzP6cNorir3GIpsT8gPgBB2/NjxaWiM6y3X1az1vRnsunQHuyKkFWPZwnEvDJYaC
NN/3jWcbhLFwKBDsaHps2+1meFP0oJFvNetzp2bjT9a9pXaQ6KhOmo5DnhLcaV97
bFBpsUuBGaYZTSS05x1RdXHqpEbgap8dtuHhVvJw9QYDQBJro K4aKyG9qqMD8cta
Pl/FAdyAqwH8Nw9efqAK+RQxSVUaue9BYEnbIRpsDK6MkP3YMFmu5ki5AQ0EUcxy
AAEIALyXYy8G2ZaTDJpdGcRhmIqOOSUlzPV7/5E5BbYKBNu4KU3nX+JLVcF5jxPQ
42c7i/WRVxE1BJTiarKGsEvCi94TTXSIUKAt3T1oGBtXmGvqbGBq8ljSGl1UTwdF
5yu50JyRSf2fqRND6P/2eHNXejDUtdvhUXIUt8h9MuUO/ipD0DnwIvMnAATJHA+R
Zqw6oNpyjRGzvr3iuWUwe4PtyJDI3ELAFkbp/NAc5TIuVHRHNOWNplcIJhM5zHuB
QQb3G/EsCn2PQZ5w5SDzavF2SpvQfDqxYpDaTLAXtF+wsJL5iaUjxwRgJPOdbCZf
2Tozd7h9MXtGJDlPKJ8eLG8ogcMAEQEAAYkBJQQYAQIADwUCUcxyAAIbDAUJCWYB
gAAKCRCTdsxl9/HZfS+hB/9BJqSmIgcoHFXnb1PVIKxekzL8+WVm5Pk/EgMQSLZ2
HX4p3ial5PEPcYgUw9YnaG4i00dwJGw5/daTWRrTzcnKd8Yq0P+DU0t96HZDSu3m
mCzE9NVAQYboFbVmGOx0eo627UBSvFqaXvAxBDYkoR8B0TnKhrQFwXkZVb30hKwD
TgAFjOGlZiE6uAdST231tFaq0bizYfe5AVXRqro20xBqNbaJNqs3SW0D831Syvdv
llOBx83/R0gg7hUkI6F2vzXicWmUwFSXRrggCSbLosHsP6isBWwvlHeRmna/aQab
YKG3gbV9iyczAS31gbogVLAZqNSWhp8vVIEE28Fyf/Ed
=x5FK
-----END PGP PUBLIC KEY BLOCK-----

If I do both of those at the same time I can prove that:

1. only I could have sent the message (signature)



My private key

Bob's public key

2. only Bob can read it (encryption)

My public key

Bob's private key

My public key

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2

mQENBFHMcgABCAC9WrYDO6K2L3VHyi4eHN6suHLqMpJ+SO+IUTuLEVnUzIoXAUXH
KozHejfV/9XoG8j933ZtszXKCog3aMESe0E0z6fNGfolvaCe5B4jwq0Jt8NHwb5L
B2dnq0CplgXcN2GJxfEHHUaf27COSobCJxPMeshUh4ZHke+g6DatmiEtBpVp41Ot
1zgxdMQkgb2H2xw28RYfYkdDoueteIkOrFLrCy9ZF9KdMhA1eBH94KnwIQshdiZR
QYEX25+M8cKCb++Rc9H6an7EG9WHOFRW40UsY52OfveOyfQPzkkRto7u2339hvHo
B/h+7xLM6FQbOUZQ9BD5w7IQHgYtXJVsUjodABEBAAG0IkthbWkgVmFuaWVhIDxr
dmFuaWVhQGluZi5lZC5hYy51az6JAT8EEwEIACkFAlYKYvECGyMFCQlmAYAHCwkI
BwMCAQYVCAIJCgsEFgIDAQIeAQIXgAAKCRCTdsxl9/HZffG+CACShuKxje3QAqew
GWh8K4gCdiY0xDqJwq3PHxmyhZmQeN/1a1KcOrIjI2b+Q75/5t+EgXOHpR0PIxfG
lZ6zOEpf6A18iFXx3JgQZdwPD0jtBiWNpOyMeBGTgIvEYG3so2VueQoeXcq3dbYp
5vstVxtD+TKHQ5CioIT75P2bzYq/XLT5aIbNQhQDPcTo0DgbRH+FvqsRXr7yeaef
JaPnxX0+1L33t2QY9zctiGyebwrvHMrIPBJ2VYCDzQkJ7uQ5eFh4ZhsMgOmzLQD4
YiGr5weIMFwAvxZOaRxEa9Vf48jiWvrxuJ8YfHWSohEScNOcYC2P8q20lJwwE26T
lpdtrwCqtB1LYW1pIFZhbmllYSA8a2FtaUB2YW5pZWEuY29tPokBQgQTAQIALAIb
IwUJCWYBgAcLCQgHAwIBBhUIAgkKCwQWAgMBAh4BAheABQJWCmMeAhkBAAoJEJN2
zGX38dl9JJAIAIW0rxrlYsrmKS6CbW8MgTxxTDOXaCt1b7F0W0QZHskIUQhEcE+a
XBYib1A5uHaatLfyjeXaD3qMEoZnQHoYMGE0GKu00wWsbhfoQzHPgwzRLkD1i75M
BIbaww0KWoVB9e4AkMakXJCnF5BXeo6AHRL2v15V205DikVnlCRXocKtu8b7LnkM
cLn70L0br1de1uyKoNzbSnO/vpKDJp0/EY5yUeV90IypZy/6wFQBehg1sXye6zn0
9wb9uUsu9+/P8pz4JILMDSevjfT7zSRSl/YP3fOfZ6N4bc+KOdwPM7u5Iyoeu9zh
pzibv3ge7VhH2xIWz8vYZ/2xT1345tWRRMOJAhwEEwECAAYFAlTnSpEACgkQjyxM
p99tBt2B8A/+OpIzOsQbQJB8yxti4I7PpD1weJDf3a81Vhm7JyXE/Xy66ypfdt3w
XmFRUuIrwezY1NebWNCRQHzQvRv/VJwjbTUx+Q3HsjIkKlHbE7iCiQXXtTRkoEny
2nudcjGI2v03C3B2JCucEw6esF1x79PI/lPv2+6tgUBKmDfOpsB2vbtqrHnmAYKL
4lQBFH1YSJgnzwo2Jkh0hcHdF90Zem1eMeiDEeVkH63893N8Swk5fBKdTj+SKZ/L
rQElBBlpMR9BmeY6bPvWRuycVKonIMR80G9iFABxjTpWBL8aGk6EeVK5EqYDGvkd
ZIarK84r+KU1KD5IfgOCN7nhwgy7VImE68caZHSRiPWZP1fVVMhydiRJv8WsoUs6
INfVU3nxH+ZYthPbY0T86leGSchBT5K/fBQvbjhrRTbTFwvjzSifb9efWylDi994
nzP6cNorir3GIpsT8gPgBB2/NjxaWiM6y3X1az1vRnsunQHuyKkFWPZwnEvDJYaC
NN/3jWcbhLFwKBDsaHps2+1meFP0oJFvNetzp2bjT9a9pXaQ6KhOmo5DnhLcaV97
bFBpsUuBGaYZTSS05x1RdXHqpEbgap8dtuHhVvJw9QYDQBJr0K4aKyG9qqMD8cta
Pl/FAdyAqwH8Nw9efqAK+RQxSVUaue9BYEnbIRpsDK6MkP3YMFmu5ki5AQ0EUcxy
AAEIALyXYy8G2ZaTDJpdGcRhmIqOOSUlzPV7/5E5BbYKBNu4KU3nX+JLVcF5jxPQ
42c7i/WRVxE1BJTiarKGsEvCi94TTXSIUKAt3T10GBtXmGvqbGBq8ljSGl1UTwdF
5yu50JyRSf2fqRND6P/2eHNXejDUtdvhUXIUt8h9MuUO/ipD0DnwIvMnAATJHA+R
Zqw6oNpyjRGzvr3iuWUwe4PtyJDI3ELAFkbp/NAc5TIuVHRHNOWNplcIJhM5zHuB
QQb3G/EsCn2PQZ5w5SDzavF2SpvQfDqxYpDaTLAXtF+wsJL5iaUjxwRgJPOdbCZf
2Tozd7h9MXtGJDlPKJ8eLG8ogcMAEQEAAYkBJQQYAQIADwUCUcxyAAIbDAUJCWYB
gAAKCRCTdsxl9/HZfS+hB/9BJqSmIgcoHFXnb1PVIKxekzL8+WVm5Pk/EgMQSLZ2
HX4p3ial5PEPcYgUw9YnaG4i00dwJGw5/daTWRrTzcnKd8Yq0P+DU0t96HZDSu3m
mCzE9NVAQYboFbVmGOx0eo627UBSvFqaXvAxBDYkoR8BoTnKhrQFwXkZVb30hKwD
TgAFjOGlZiE6uAdST231tFaq0bizYfe5AVXRqro20xBqNbaJNqs3SW0D831Syvdv
llOBx83/R0gg7hUkI6F2vzXicWmUwFSXRrggCSbLosHsP6isBWwvlHeRmna/aQab
YKG3gbV9iyczAS31gbogVLAZqNSWhp8vVIEE28Fyf/Ed
=x5FK
-----END PGP PUBLIC KEY BLOCK-----

Encryption properties we want:

1. The communication between you and the other party is **confidential** and has **not been changed**
   - No one can read what you sent
   - No one can change what you sent
2. **Knowing who** you are communicating with
   - You are talking to who you think you are talking to and not someone else

LINKING KEYS AND IDENTITIES

But we still have a problem:

All that assumes that we know which key goes with which person.

One of the founding problems in Usable Security and Privacy.

Even now we have no good answer.

How do we solve the identity problem?

Idea: Have the humans do the linking of identity to cryptographic keys.

We could post the public key somewhere highly public and verifiable it came from us.

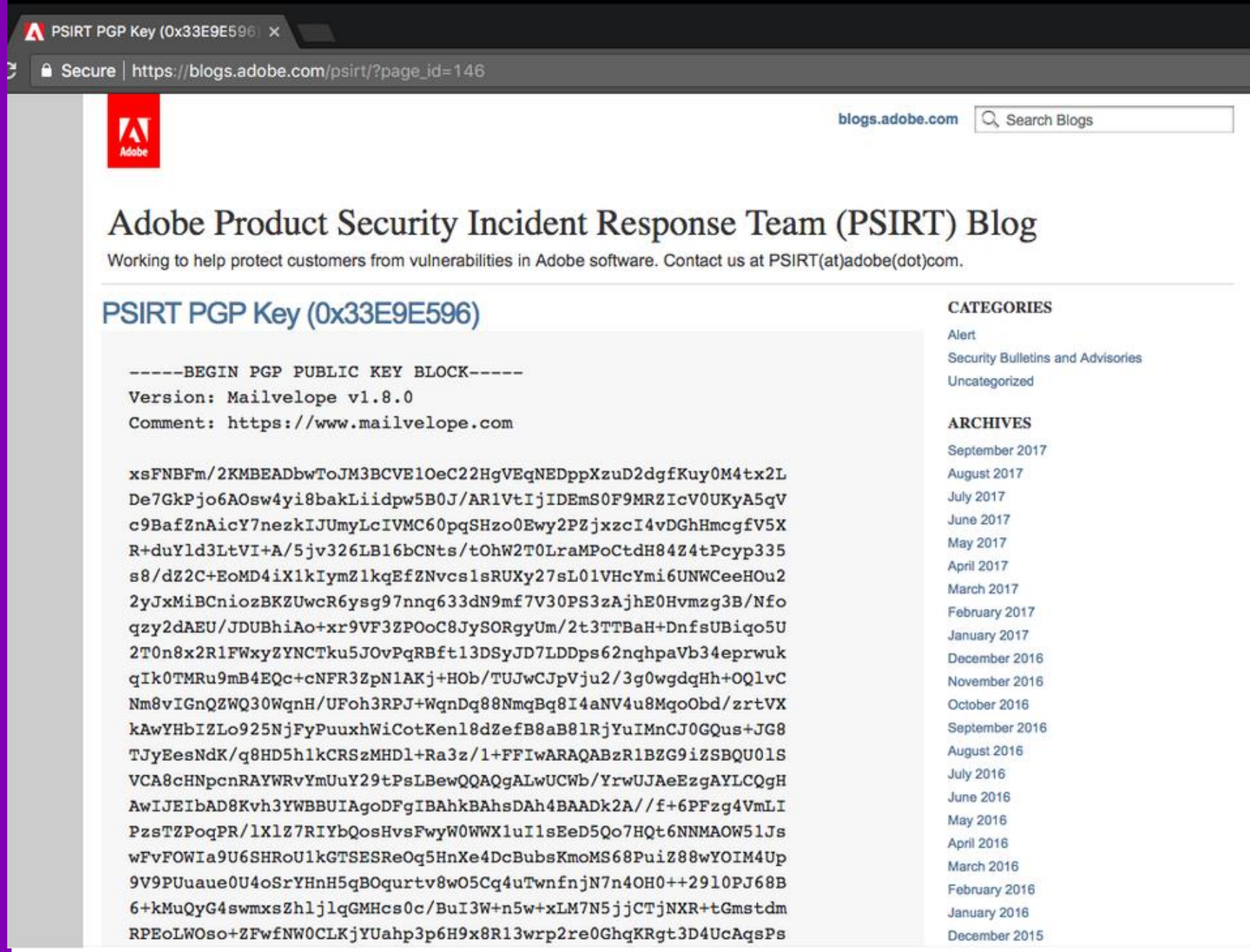


Photo credit: Juho Nurminen @jupenur

Other people can then compare the keys on their computers to the highly visible copy.

xsFNBFm/2KMBEADbwToJM3BCVE1OeC22HgVEqNEDppXzuD2dgfKuy0M4tx2L
De7GkPjo6AOsw4yi8bakLiidpw5B0J/AR1VtIjIDEmS0F9MRZIcV0UKyA5qV
c9BafZnAicY7nezkIJUmyLcIVMC60pqSHzo0Ewy2PZjxzcI4vDGhHmcgfV5X
R+duYld3LtVI+A/5jv326LB16bCNts/tOhW2T0LraMPoCtdH84Z4tPcyp335
s8/dZ2C+EoMD4iX1kIymZ1kqEfZNvcs1sRUXy27sL01VHcYmi6UNWCeeHOu2
2yJxMiBCniozBKZUwcR6ysg97nnq633dN9mf7V30PS3zAjhE0Hvmzg3B/Nfo
qzy2dAEU/JDUBhiAo+xr9VF3ZPOoC8JySORgyUm/2t3TTBaH+DnfsUBiqo5U
2T0n8x2R1FWxyZYNCTku5JOvPqRBft13DSyJD7LDDps62nqhpaVb34eprwuk
qIk0TMRu9mB4EQc+cNFR3ZpN1AKj+HOb/TUJwCJpVju2/3g0wgdqHh+OQlvC
Nm8vIGnQZWQ30WqnH/UFoh3RPJ+WqnDq88NmqBq8I4aNV4u8MqoObd/zrtVX
kAwYHbIZLo925NjFyPuuxhWiCotKenl8dZefB8aB8lRjYuIMnCJ0GQus+JG8
TJyEesNdK/q8HD5h1kCRSzMHDl+Ra3z/1+FFIwARAQABzR1BZG9iZSBQU0lS
VCA8cHNpcnRAYWRvYmUuY29tPsLBewQQAQgALwUCWb/YrwUJAeEzgAYLCQgH
AwIJEIbAD8Kvh3YWBBUIAgoDFgIBAhkBAhsDAh4BAADk2A//f+6PFzg4VmLI
PzsTZPoqPR/lXlZ7RIYbQosHvsFwyW0WWX1uI1sEeD5Qo7HQt6NNMAOW51Js
wFvFOWIa9U6SHRoU1kGTSESReOq5HnXe4DcBubsKmoMS68PuiZ88wYOIM4Up
9V9PUuaue0U4oSrYHnH5qBOqurtv8wO5Cq4uTwnfnjN7n4OH0++29l0PJ68B
6+kMuQyG4swmxsZhljlqGMHcs0c/BuI3W+n5w+xLM7N5jjCTjNXR+tGmstdm
RPEoLWOso+ZFwfNW0CLKjYUahp3p6H9x8R13wrp2re0GhqKRgt3D4UcAqsPs

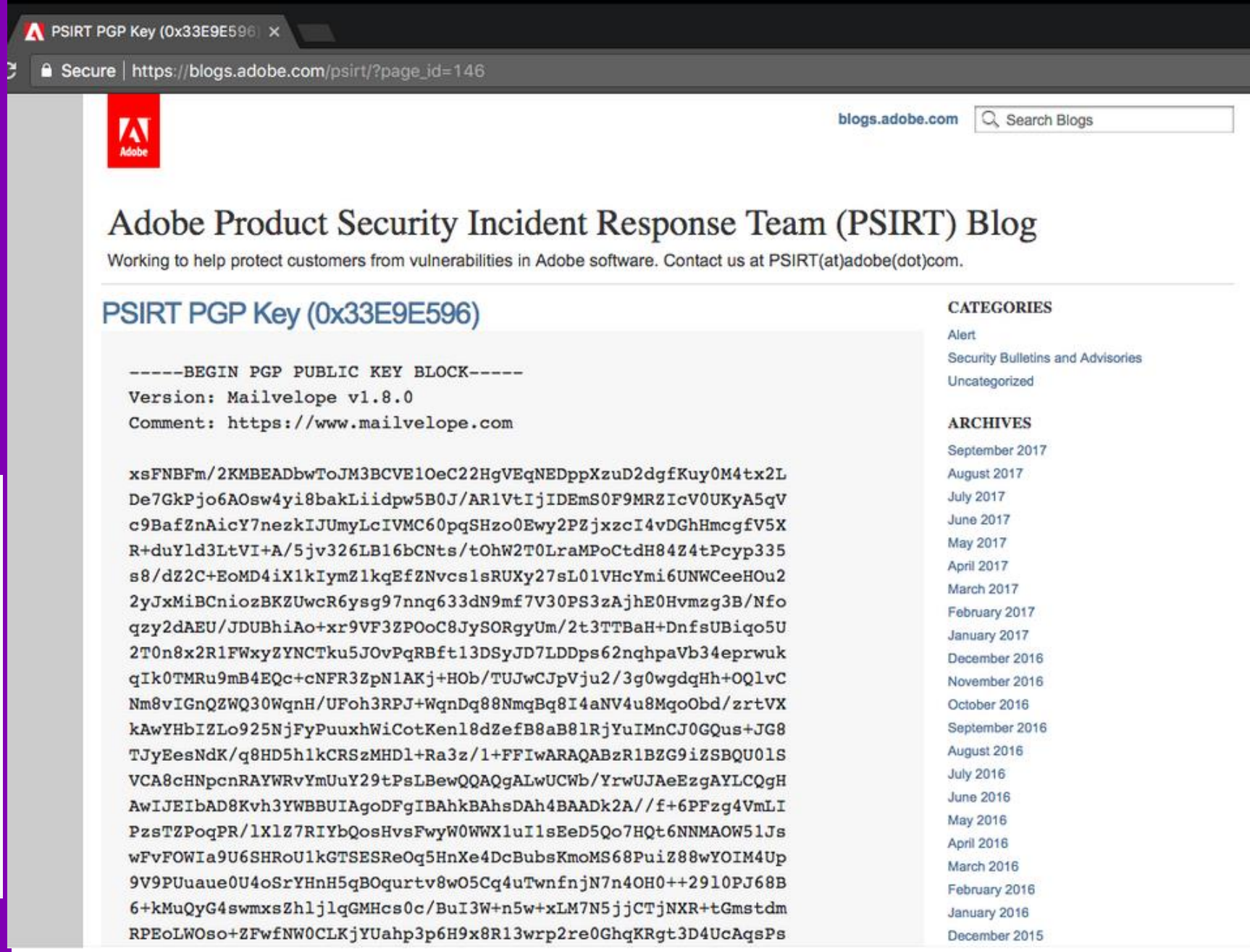
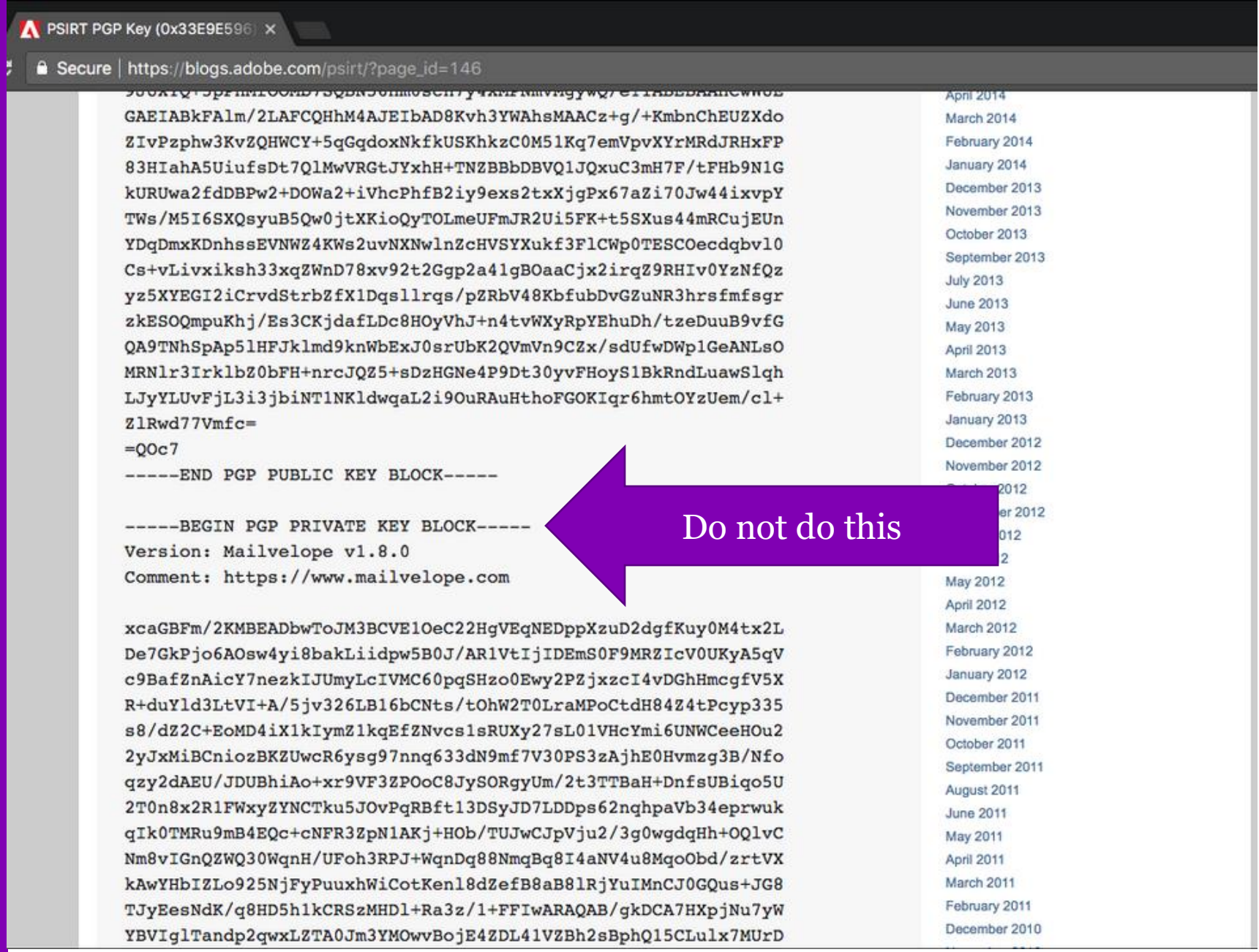


Photo credit: Juho Nurminen @jupenur

Though we must be careful to post ONLY the public key…



Do not do this

Photo credit: Juho Nurminen @jupenur

Nice idea, but it does not scale.

Also a chicken-and-egg problem. How do we find a place guaranteed to be from us without using cryptography?

Idea 2: What if everyone did a few verifications. We could slowly build a web of verifications like:

Alice verified Bob's key
Bob verified Charlie's key
so
Alice can trust Charlie's key

Web of trust

- Alice hand verifies that Bob's public key really does belong to Bob
- Then Alice "signs" the key by encrypting it with her private key.
- Now anyone that has hand verified Alice's key, can also trust Bob's key (if they trust Alice to do verifications).

My public key

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2

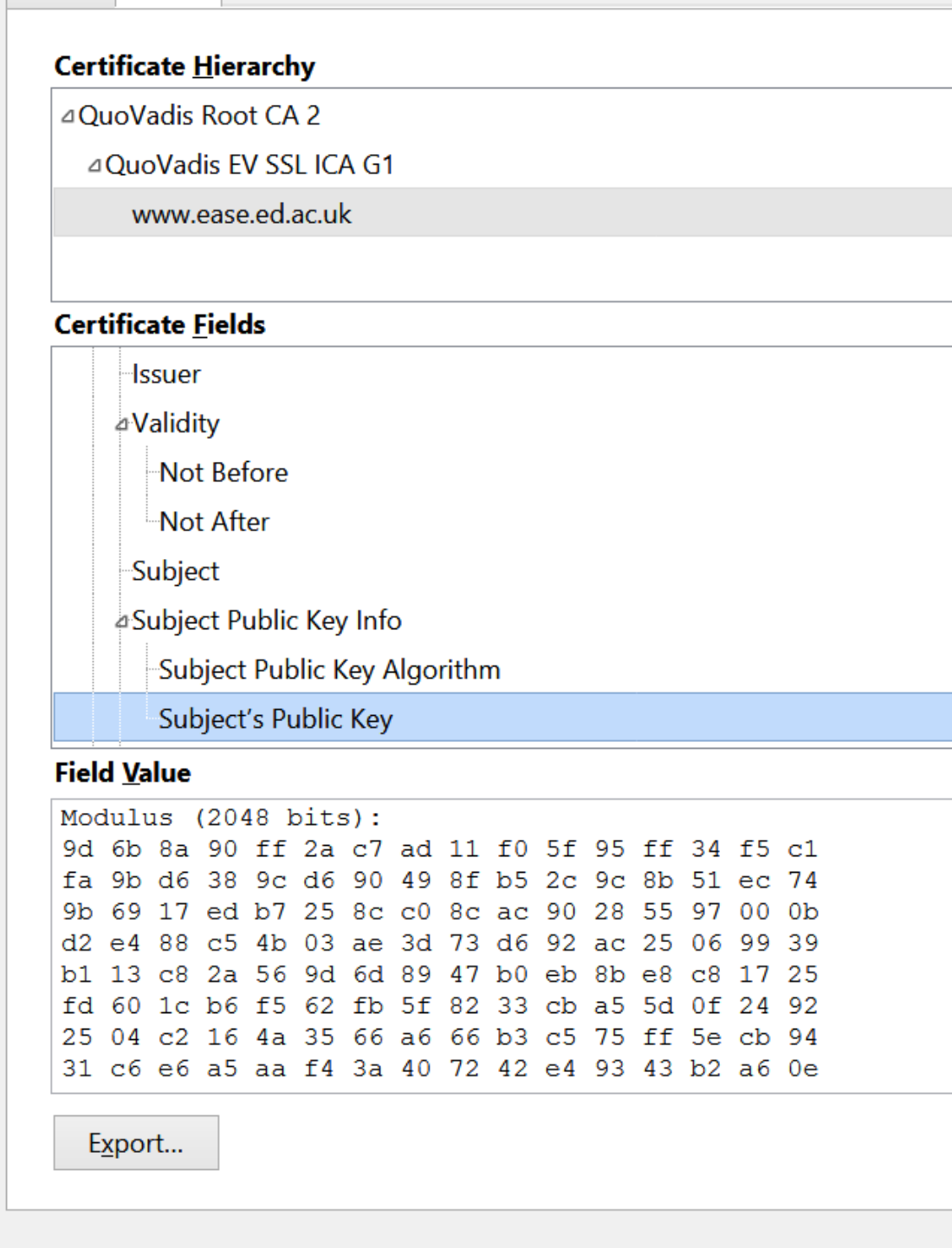
mQENBFHMcgABCAC9WrYDO6K2L3VHyi4eHN6suHLqMpJ+SO+IUTuLEVnUzIoXAUXH
KozHejfV/9XoG8j933ZtszXKCog3aMESe0E0z6fNGfolvaCe5B4jwq0Jt8NHwb5L
B2dnq0CplgXcN2GJxfEHHUaf27COSobCJxPMeshUh4ZHke+g6DatmiEtBpVp41Ot
1zgxdMQkgb2H2xw28RYfYkdDoueteIkOrFLrCy9ZF9KdMhA1eBH94KnwIQshdiZR
QYEX25+M8cKCb++Rc9H6an7EG9WHOFRW40UsY52OfveOyfQPzkkRto7u2339hvHo
B/h+7xLM6FQbOUZQ9BD5w7IQHgYtXJVsUjodABEBAAG0IkthbWkgVmFuaWVhIDxr
dmFuaWVhQGluZi5lZC5hYy51az6JAT8EEwEIACkFAlYKYvECGyMFCQlmAYAHCwkI
BwMCAQYVCAIJCgsEFgIDAQIeAQIXgAAKCRCTdsxl9/HZffG+CACShuKxje3QAqew
GWh8K4gCdiYoxDqJwq3PHxmyhZmQeN/1a1KcOrIjI2b+Q75/5t+EgXOHpR0PIxfG
lZ6zOEpf6A18iFXx3JgQZdwPD0jtBiWNpOyMeBGTgIvEYG3so2VueQoeXcq3dbYp
5vstVxtD+TKHQ5CioIT75P2bzYq/XLT5aIbNQhQDPcTo0DgbRH+FvqsRXr7yeaef
JaPnxX0+1L33t2QY9zctiGyebwrvHMrIPBJ2VYCDzQkJ7uQ5eFh4ZhsMgOmzLQD4
YiGr5weIMFwAvxZOaRxEa9Vf48jiWvrxuJ8YfHWS0hEScNOcYC2P8q20lJwwE26T
lpdtrwCqtB1LYW1pIFZhbmllYSA8a2FtaUB2YW5pZWEuY29tPokBQgQTAQIALAIb
IwUJCWYBgAcLCQgHAwIBBhUIAgkKCwQWAgMBAh4BAheABQJWCmMeAhkBAAoJEJN2
zGX38dl9JJAIAIW0rxrlYsrmKS6CbW8MgTxxTDOXaCt1b7F0W0QZHskIUQhEcE+a
XBYib1A5uHaatLfyjeXaD3qMEoZnQHoYMGE0GKu00wWsbhfoQzHPgwzRLkD1i75M
BIbaww0KW0VB9e4AkMakXJCnF5BXeo6AHRL2v15V205DikVnlCRXocKtu8b7LnkM
cLn70Lobr1de1uyKoNzbSnO/vpKDJp0/EY5yUeV90IypZy/6wFQBehg1sXye6zn0
9wb9uUsu9+/P8pz4JILMDSevjfT7zSRSl/YP3fOfZ6N4bc+KOdwPM7u5Iyoeu9zh
pzibv3ge7VhH2xIWz8vYZ/2xT1345tWRRMOJAhwEEwECAAYFAlTnSpEACgkQjyxM
p99tBt2B8A/+OpIzOsQbQJB8yxti4I7PpD1weJDf3a81Vhm7JyXE/Xy66ypfdt3w
XmFRUuIrwezY1NebWNCRQHzQvRv/VJwjbTUx+Q3HsjIkKlHbE7iCiQXXtTRkoEny
2nudcjGI2v03C3B2JCucEw6esF1x79PI/lPv2+6tgUBKmDfOpsB2vbtqrHnmAYKL
4lQBFH1YSJgnzwo2Jkh0hcHdF90Zem1eMeiDEeVkH63893N8Swk5fBKdTj+SKZ/L
rQElBBlpMR9BmeY6bPvWRuycVKonIMR80G9iFABxjTpWBL8aGk6EeVK5EqYDGvkd
ZIarK84r+KU1KD5IfgOCN7nhwgy7VImE68caZHSRiPWZP1fVVMhydiRJv8WsoUs6
INfVU3nxH+ZYthPbY0T86leGSchBT5K/fBQvbjhrRTbTFwvjzSifb9efWylDi994
nzP6cNorir3GIpsT8gPgBB2/NjxaWiM6y3X1az1vRnsunQHuyKkFWPZwnEvDJYaC
NN/3jWcbhLFwKBDsaHps2+1meFP0oJFvNetzp2bjT9a9pXaQ6KhOmo5DnhLcaV97
bFBpsUuBGaYZTSS05x1RdXHqpEbgap8dtuHhVvJw9QYDQBJr0K4aKyG9qqMD8cta
Pl/FAdyAqwH8Nw9efqAK+RQxSVUaue9BYEnbIRpsDK6MkP3YMFmu5ki5AQ0EUcxy
AAEIALyXYy8G2ZaTDJpdGcRhmIqOOSUlzPV7/5E5BbYKBNu4KU3nX+JLVcF5jxPQ
42c7i/WRVxE1BJTiarKGsEvCi94TTXSIUKAt3T10GBtXmGvqbGBq8ljSGl1UTwdF
5yu50JyRSf2fqRND6P/2eHNXejDUtdvhUXIUt8h9MuUO/ipD0DnwIvMnAATJHA+R
Zqw60NpyjRGzvr3iuWUwe4PtyJDI3ELAFkbp/NAc5TIuVHRHNOWNplcIJhM5zHuB
QQb3G/EsCn2PQZ5w5SDzavF2SpvQfDqxYpDaTLAXtF+wsJL5iaUjxwRgJPOdbCZf
2Tozd7h9MXtGJDlPKJ8eLG80gcMAEQEAAYkBJQQYAQIADwUCUcxyAAIbDAUJCWYB
gAAKCRCTdsxl9/HZfS+hB/9BJqSmIgcoHFXnb1PVIKxekzL8+WVm5Pk/EgMQSLZ2
HX4p3ial5PEPcYgUw9YnaG4i00dwJGw5/daTWRrTzcnKd8Yq0P+DU0t96HZDSu3m
mCzE9NVAQYboFbVmGOx0eo627UBSvFqaXvAxBDYkoR8B0TnKhrQFwXkZVb30hKwD
TgAFjOGlZiE6uAdST231tFaq0bizYfe5AVXRqro20xBqNbaJNqs3SW0D831Syvdv
llOBx83/R0gg7hUkI6F2vzXicWmUwFSXRrggCSbLosHsP6isBWwvlHeRmna/aQab
YKG3gbV9iyczAS31gbogVLAZqNSWhp8vVIEE28Fyf/Ed
=x5FK
-----END PGP PUBLIC KEY BLOCK-----

Wonderful idea in theory. But verifying those long keys is hard… also I don’t trust most people to do a thorough job of it….

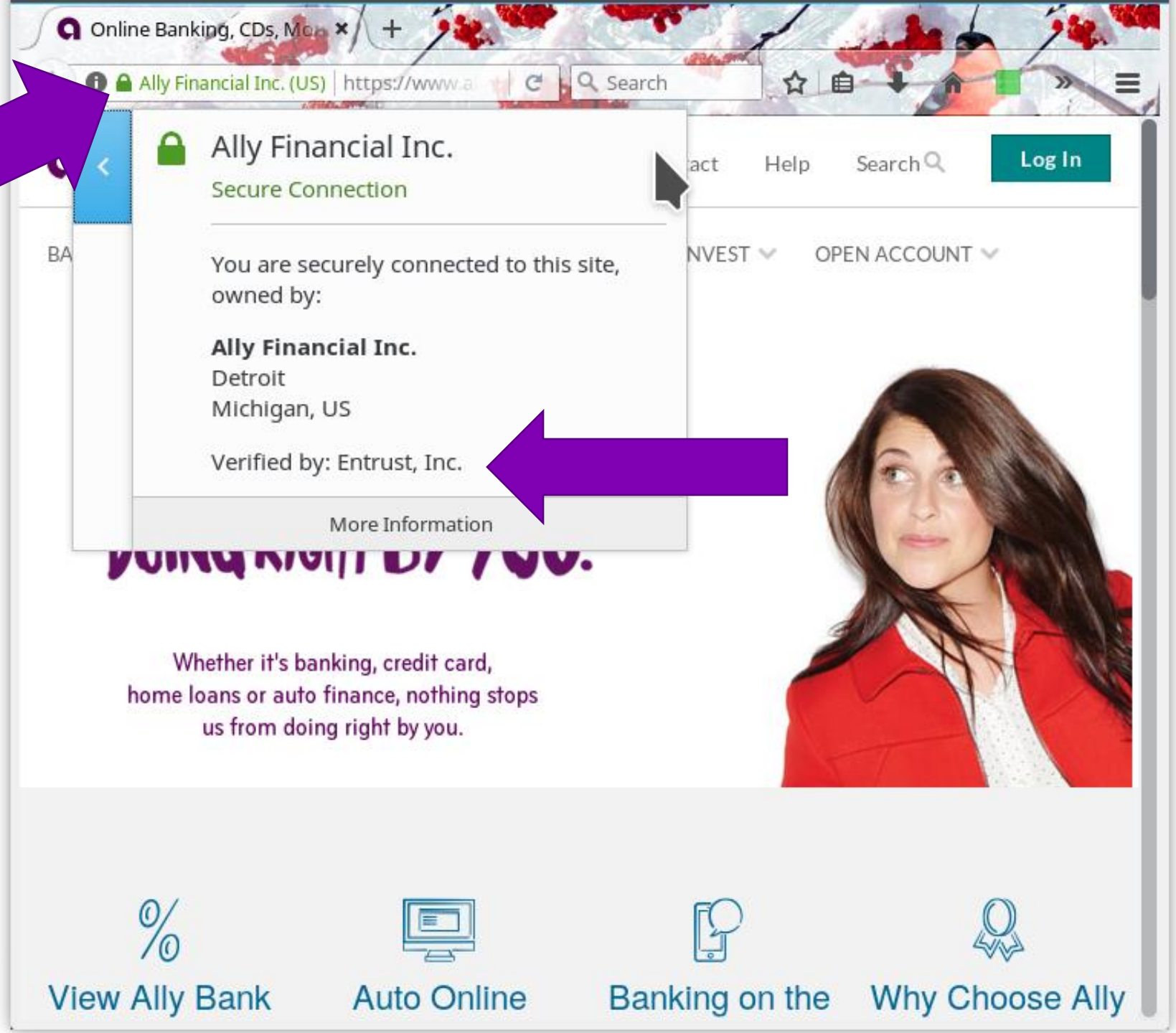
Idea 3: What if a couple of trusted groups did the verifications. Then they could have high standards and everyone could just trust them.

Certificate Authorities

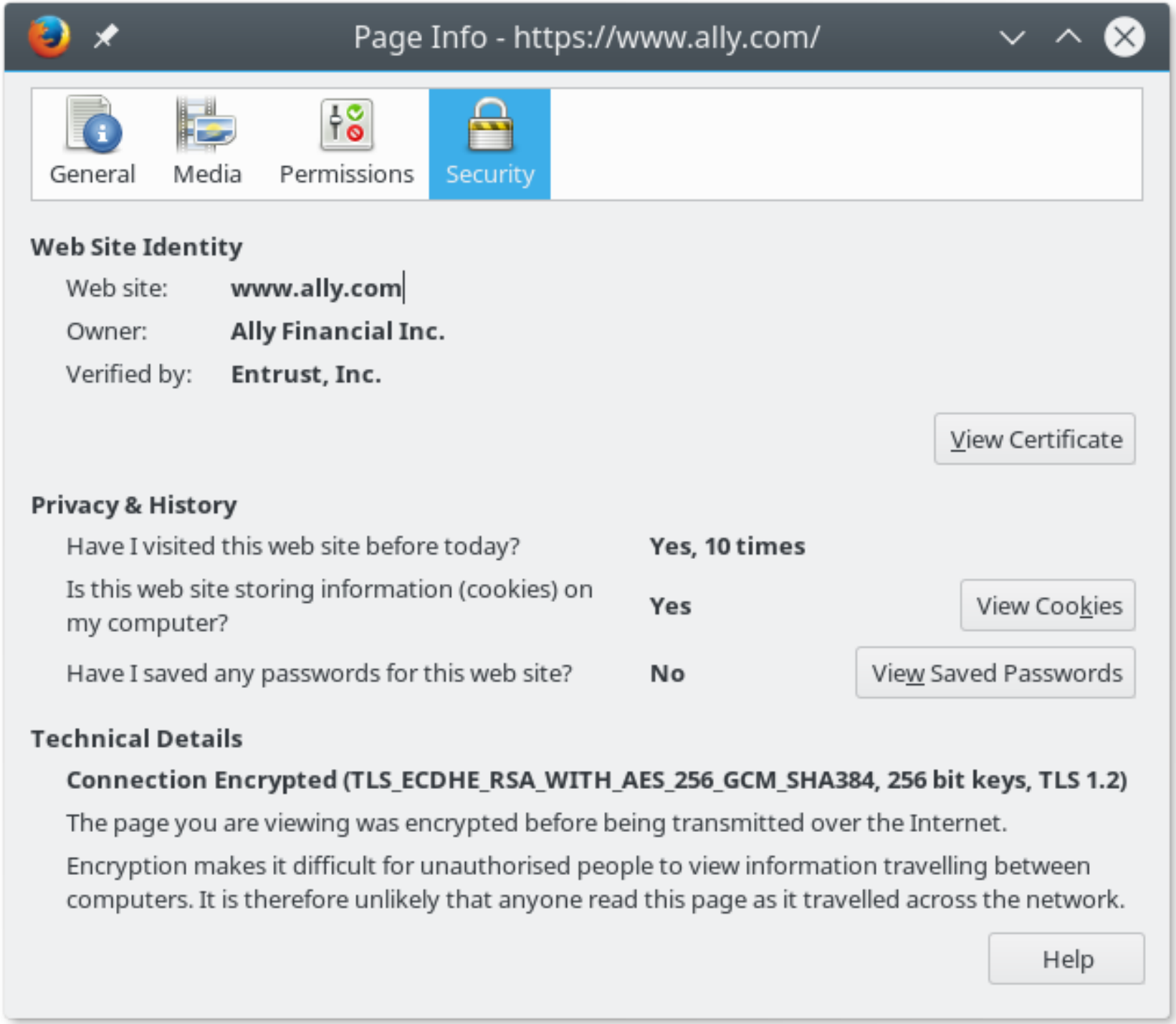
- A certificate authority verifies some properties of a person/organization and issues a “certificate” signed by their private key.
- Certificates can be quite detailed about what has been verified, and what they have been verified to do.

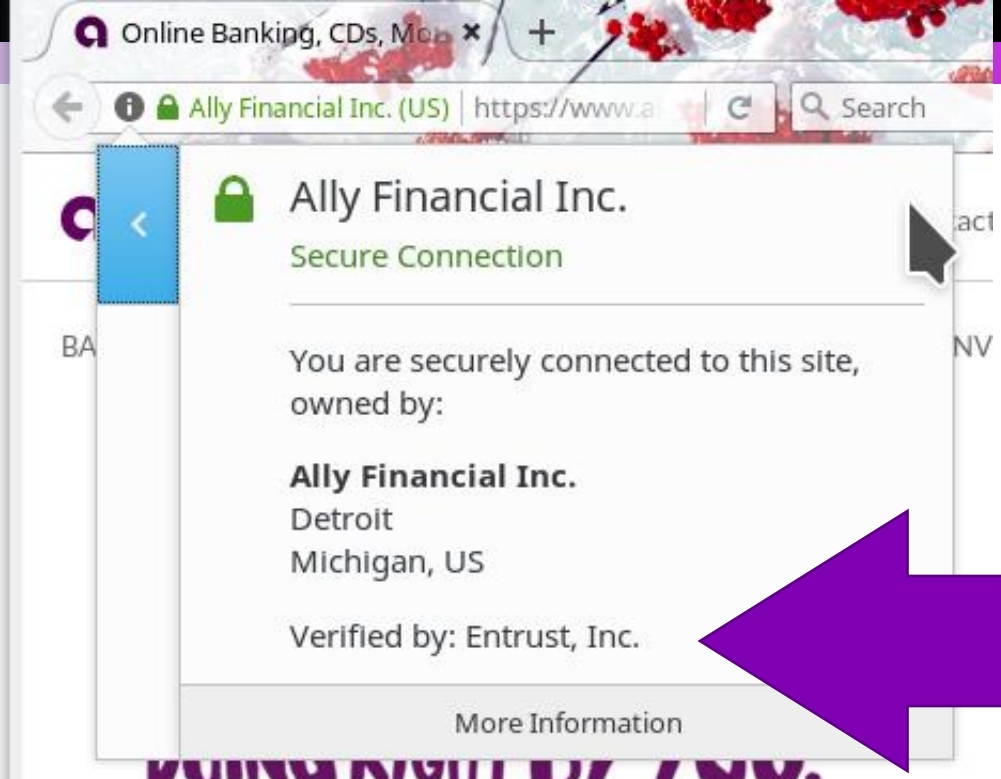


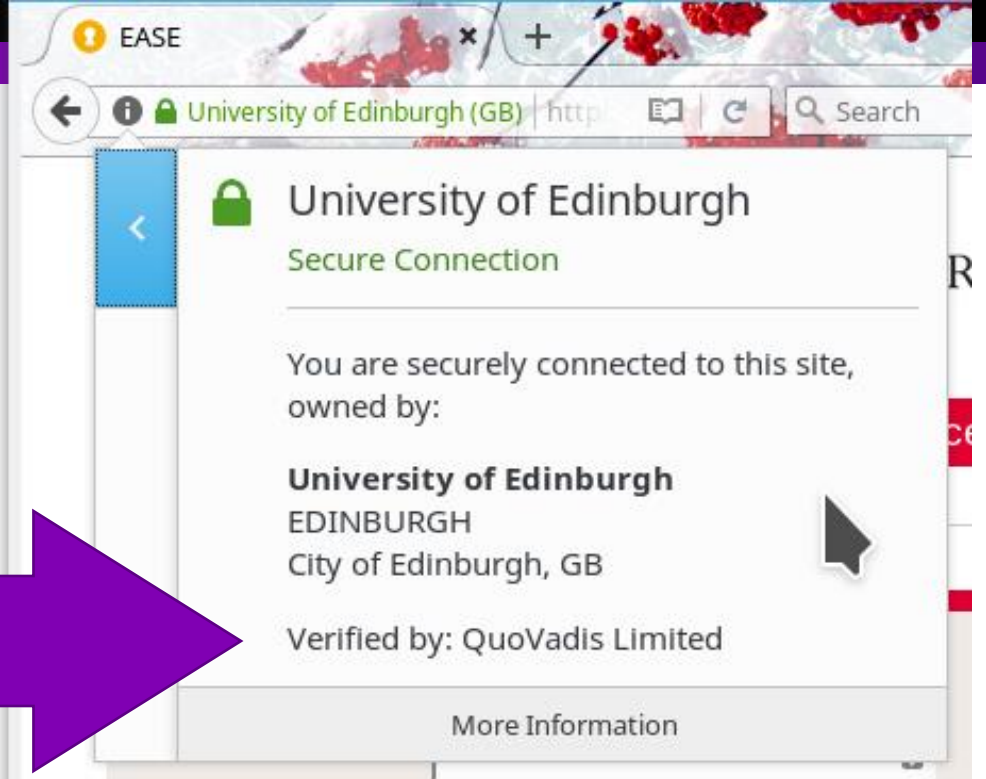
Certificate Authorities are used by browsers to verify identity



You can see lots of details about any encrypted connection.



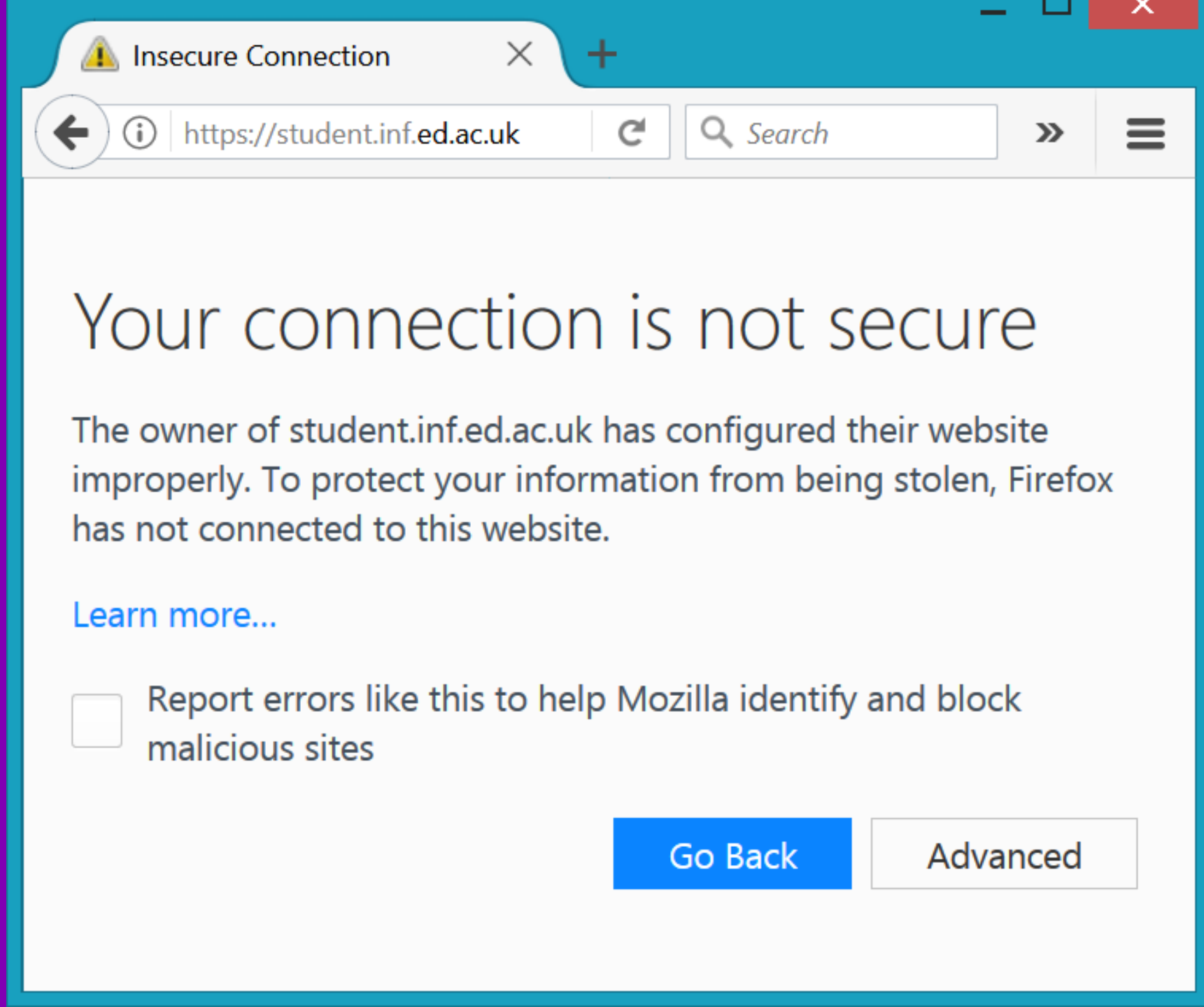




But now don't we just have the same problem again?

How does the browser know which Certificate Authorities to trust?

Clearly some Certificate Authorities are trusted and some are not.



Your operating system and your browser both maintain lists of Certificate Authorities that they trust.

These lists differ between operating systems, browsers, and organizations.

Each organization makes its own trust decisions about Certificate Authorities

INFOWORLD TECH WATCH

By Fahmida Y. Rashid, Senior Writer, InfoWorld | MAR 24, 2017

Google to Symantec: We don't trust you anymore

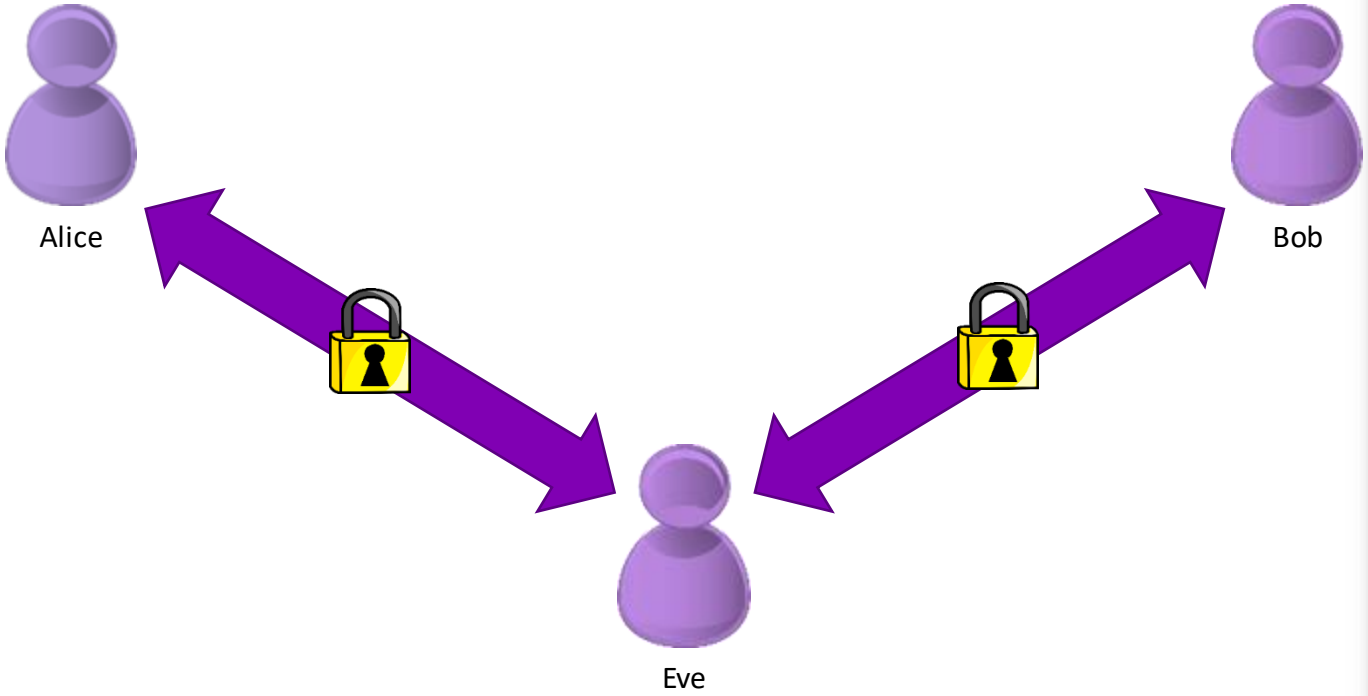
Admins need to consider whether they still want to use Symantec after its repeated mistakes with issuing TLS certificates

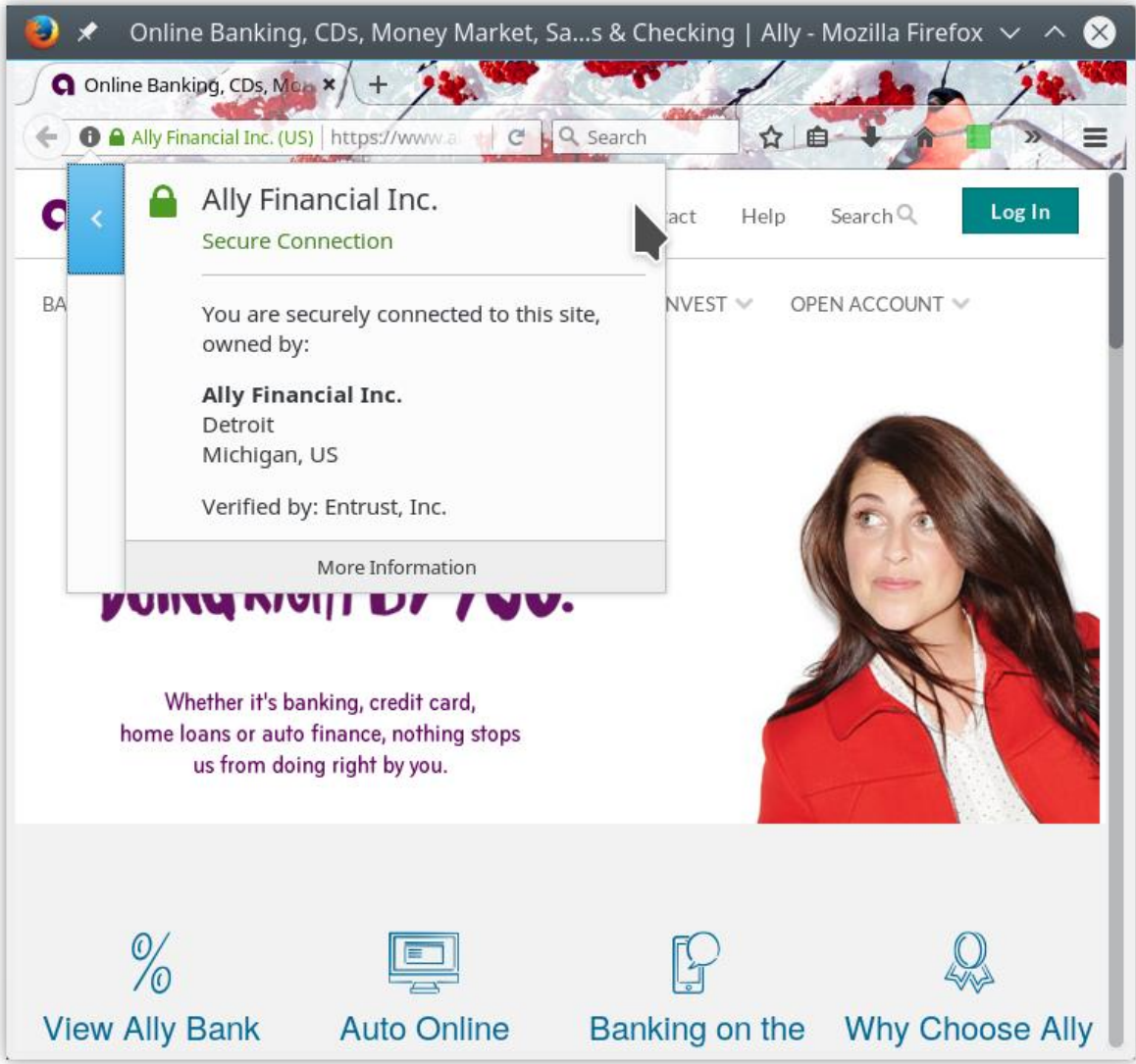
geralt via pixabay

Security teams, network administrators, and operations teams have busy days ahead. Google's Chrome development team is fed up with Symantec as a certificate authority and has announced plans to no longer trust current Symantec certificates.

In the past 18 months, Google has tangled repeatedly with Symantec over the way it issues transport layer security (TLS) certificates, with Symantec promising to do better. The latest incident—an investigation into 127 mis-issued certificates—ballooned into "at least 30,000, issued over a period spanning several years," Ravi Sleevi, a software engineer on the Google

In Conclusion...





Why Johnny Can’t Encrypt

In Proceedings of the 8th USENIX Security Symposium, August 1999, pp. 169-183

Why Johnny Can’t Encrypt: A Usability Evaluation of PGP 5.0

Alma Whitten
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
alma@cs.cmu.edu

J. D. Tygar[1]
EECS and SIMS
University of California
Berkeley, CA 94720
tygar@cs.berkeley.edu

Abstract

User errors cause or contribute to most computer security failures, yet user interfaces for security still tend to be clumsy, confusing, or near-nonexistent. Is this simply due to a failure to apply standard user interface design techniques to security? We argue that, on the contrary, effective security requires a different usability standard, and that it will not be achieved through the user interface design techniques appropriate to other types of consumer software.

To test this hypothesis, we performed a case study of a security program which does have a good user interface by general standards: PGP 5.0. Our case study used a cognitive walkthrough analysis together with a laboratory user test to evaluate whether PGP 5.0 can be successfully used by cryptography novices to achieve effective electronic mail security. The analysis found a number of user interface design flaws that may

1 Introduction

Security mechanisms are only effective when used correctly. Strong cryptography, provably correct protocols, and bug-free code will not provide security if the people who use the software forget to click on the encrypt button when they need privacy, give up on a communication protocol because they are too confused about which cryptographic keys they need to use, or accidentally configure their access control mechanisms to make their private data world-readable. Problems such as these are already quite serious: at least one researcher [2] has claimed that configuration errors are the probable cause of more than 90% of all computer security failures. Since average citizens are now increasingly encouraged to make use of networked computers for private transactions, the need to make security manageable for even untrained users has become critical [4, 9].

Why Enc

In Proceedings of the 8th USENIX Security Symposium, August 1999, pp. 169-183

If an average user of email feels the need for privacy and authentication, and acquires PGP with that purpose in mind, will PGP's current design allow that person to realize what needs to be done, figure out how to do it, and avoid dangerous errors, without becoming so frustrated that he or she decides to give up on using PGP after all?

interface by general standards: PGP 5.0. Our case study used a cognitive walkthrough analysis together with a laboratory user test to evaluate whether PGP 5.0 can be successfully used by cryptography novices to achieve effective electronic mail security. The analysis found a number of user interface design flaws that may

ive when used
rovably correct
ovide security if
to click on the
y, give up on a
re too confused
need to use, or
rol mechanisms
able. Problems
s: at least one
ration errors are

the probable cause of more than 90% of all computer security failures. Since average citizens are now increasingly encouraged to make use of networked computers for private transactions, the need to make security manageable for even untrained users has become critical [4, 9].

Understanding the problem

Definition: Security software is usable if the people who are expected to use it:

1. are reliably made aware of the security tasks they need to perform;
2. are able to figure out how to successfully perform those tasks;
3. don’t make dangerous errors; and
4. are sufficiently comfortable with the interface to continue using it.

Users need to:

- understand that privacy is achieved by encryption, and figure out how to encrypt email and how to decrypt email received from other people
- understand that authentication is achieved through digital signatures, and figure out how to sign email and how to verify signatures on email from other people
- understand that in order to sign email and allow other people to send them encrypted email a key pair must be generated, and figure out how to do so
- understand that in order to allow other people to verify their signature and to send them encrypted email, they must publish their public key, and figure out some way to do so
- understand that in order to verify signatures on email from other people and send encrypted email to other people, they must acquire those people's public keys
- manage to avoid such dangerous errors as accidentally failing to encrypt, trusting the wrong public keys, failing to back up their private keys, and forgetting their pass phrases
- be able to succeed at all of the above within a few hours of reasonably motivated effort

Tested usability using two methods

- Cognitive Walkthrough
  - A set of experts review the experts and make an informed guess about what will be problematic
  - Paired with heuristics – The experts state how the user interface supports or violates common HCI principles (Heuristics)
- Lab Study
  - Ask the participant to perform a set of tasks
  - Very similar to a think aloud, but without the talking aloud part

Cognitive walkthrough outcomes

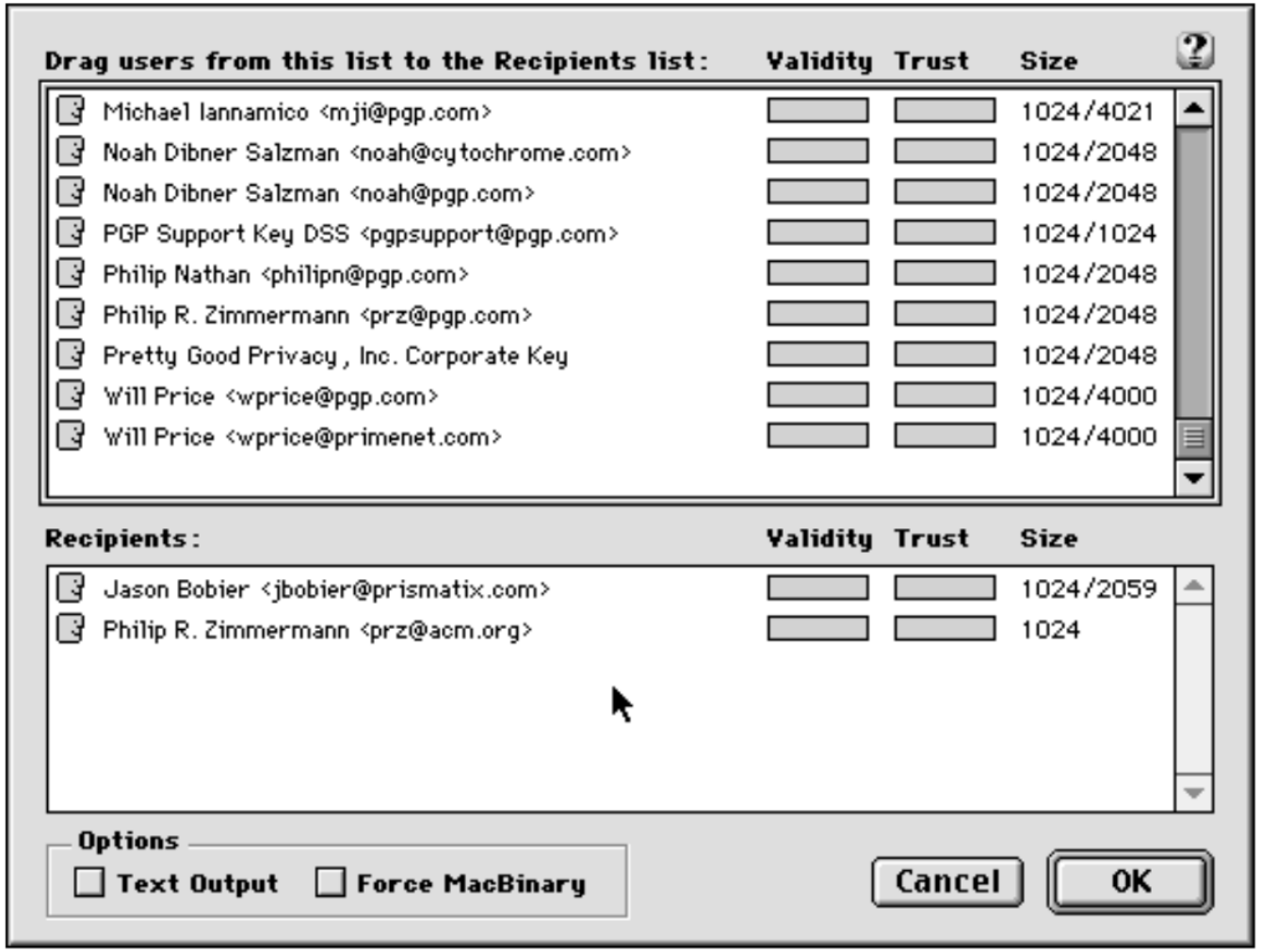
- **Visual metaphors** – Do key and lock pictures make sense?
- **Different key types** – Public vs private keys, or maybe signing and encryption keys?
- **Key server** – Used for sharing keys
- **Key management policy** – Trust and validity ratings
- **Consistency** – Use of the same terms everywhere
- **Too much information** – Information like key size, hashes, and trust
- **Irreversible actions**
  - Accidentally deleting the private key
  - Accidentally publicizing a key
  - Accidentally revoking a key
  - Forgetting the pass phrase
  - Failing to back up the key rings

Lab study

- 12 participants with CS backgrounds
- Participant had to send several emails to team members (the researchers)
  - Creating a key pair
  - Sending their public key to team members
  - Getting team members' public keys
  - Sending the email
  - Decrypting response email

- 3 – emailed the private key to the team member
  - 1 never realized the error
- 1 – forgot their pass phase and had to re-generate keys
- 1 – never figured out how to encrypt
- 7 – used their public keys to encrypt
  - 1 created a separate key pair for each team member
- 3 – successfully sent an encrypted email to the whole team and were able to decrypt an response email

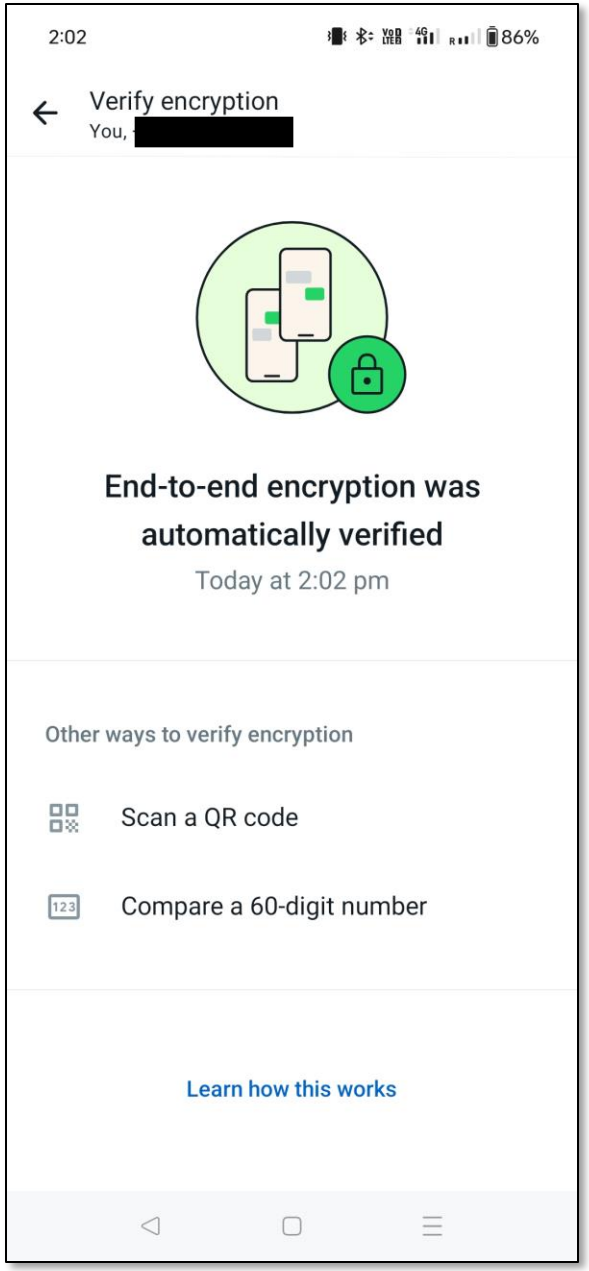
Whitten and Tygar evaluated PGP encryption in 1999, surely it must be more usable now.



"SECURE" MESSAGING

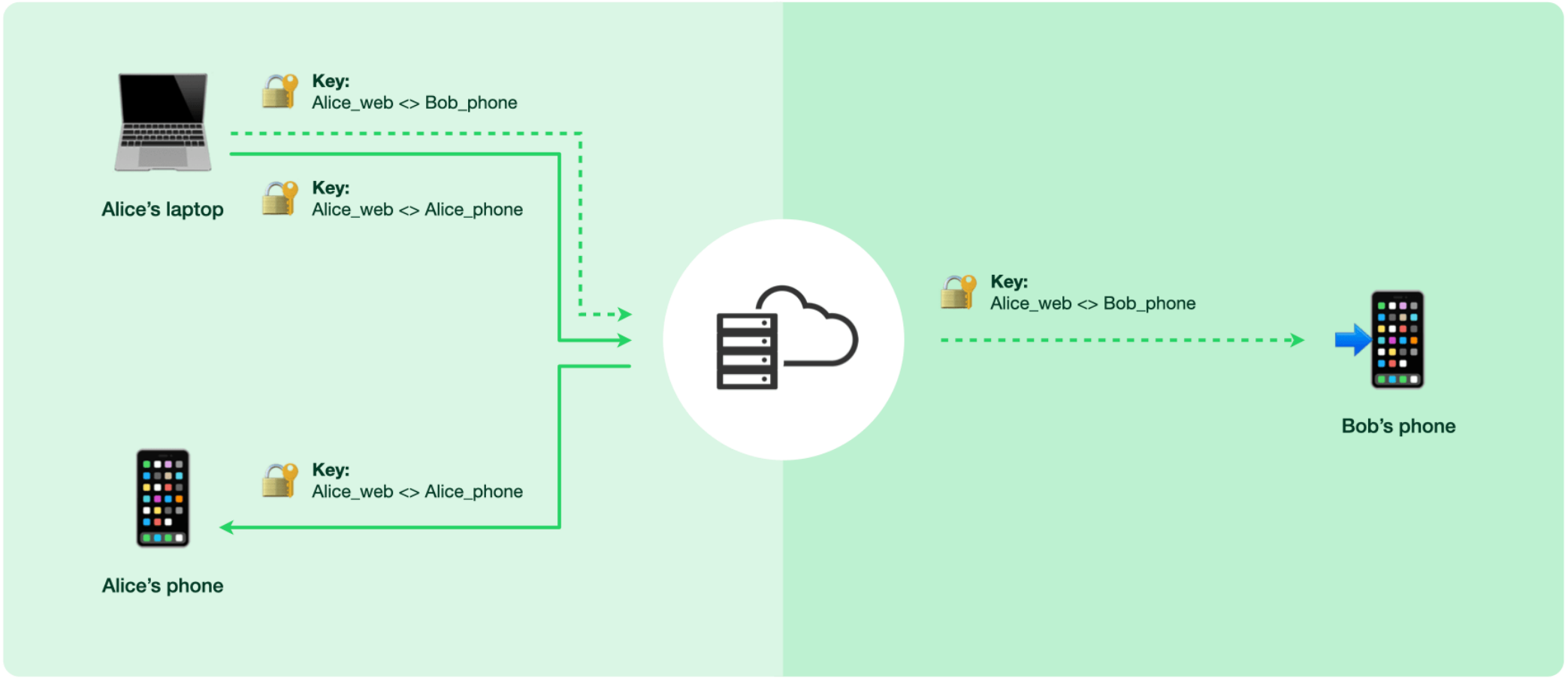
WhatsApp

- All messages, including group chats, are end-to-end encrypted
- The “ends” are the WhatsApp app on both devices
- Keys are managed by WhatsApp itself and shared with the devices as needed



WhatsApp: syncing chats

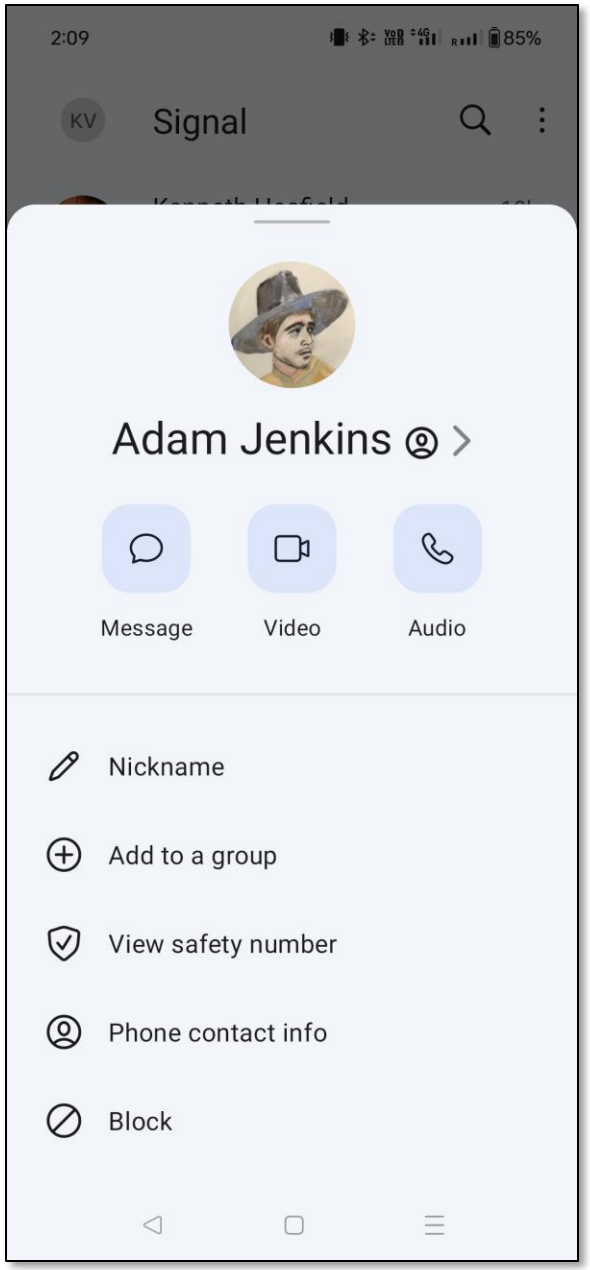
Life of a message: Multi-Device (new)

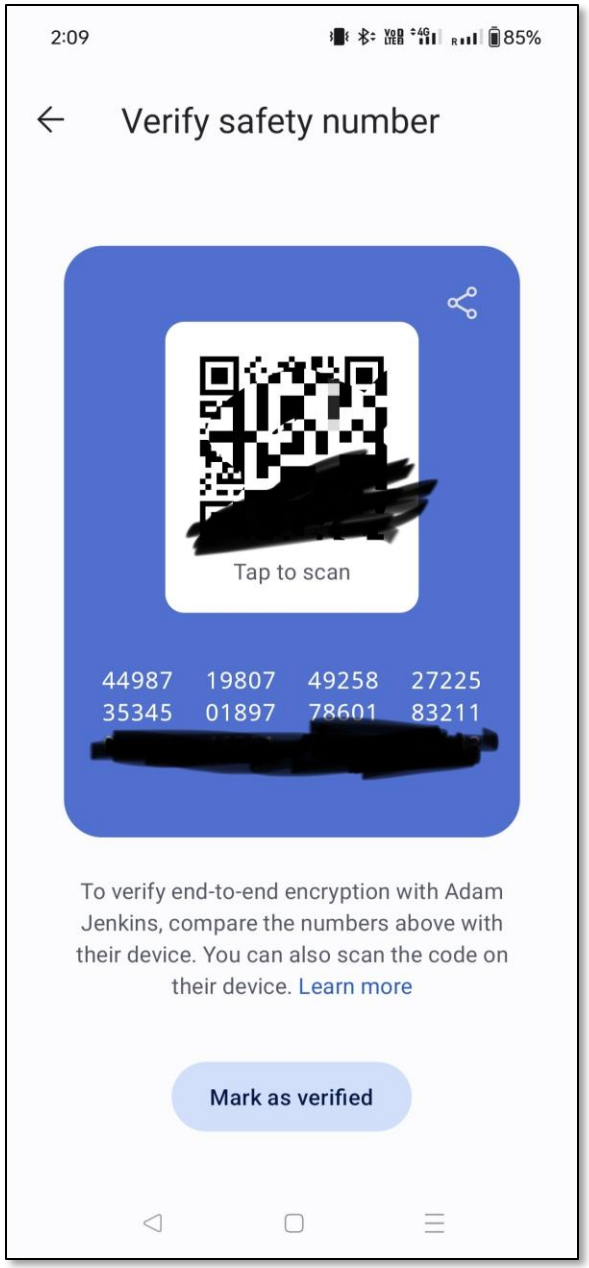


https://engineering.fb.com/2021/07/14/security/whatsapp-multi-device/

Signal

- End to end encrypted
- The “ends” are the apps on both sides





Telegram

- Only Secret chats are end-to-end encrypted
- Secret chats are more restricted than other messaging tools
- Video and audio calls are end-to-end encrypted

The Security Blanket of the Chat World: An Analytic Evaluation and a User Study of Telegram

Ruba Abu-Salma[1,*], Kat Krol[2,*,‡], Simon Parkin[1], Victoria Koh[1], Kevin Kwan[1], Jazib Mahboob[1], Zahra Traboulsi[1], and M. Angela Sasse[1]

[1] University College London (UCL), {ruba.abu-salma.13, s.parkin, victoria.koh.13, kevin.kwan.13, jazib.mahboob.13, zahra.traboulsi.13, a.sasse}@ucl.ac.uk

[2] University of Cambridge, kat.krol@cl.cam.ac.uk

***Abstract*—The computer security community has advocated widespread adoption of secure communication tools to protect personal privacy. Several popular communication tools have adopted end-to-end encryption (e.g., WhatsApp, iMessage), or promoted security features as selling points (e.g., Telegram, Signal). However, previous studies have shown that users may not understand the security features of the tools they are using, and may not be using them correctly. In this paper, we present a study of Telegram using two complementary methods: (1) a lab-based user study (11 novices and 11 Telegram users), and (2) a hybrid analytical approach combining cognitive walk-through and heuristic evaluation to analyse Telegram's user interface. Participants who use Telegram feel secure because they feel they are using a secure tool, but in reality Telegram offers limited security benefits to most of its users. Most participants develop a habit of using the less secure default chat mode at all times. We also uncover several user interface design issues that impact security, including technical jargon, inconsistent use of terminology, and making some security features clear and others not. For instance, use of the end-to-end-encrypted *Secret Chat* mode requires both the sender and recipient be online at the same time, and *Secret Chat* does not support group conversations.**

I. Introduction

Recent events have seen developers offering messaging tools with greater security to support a diverse range of user motivations. These include revelations about mass surveillance and the potential for user tracking in communication tools (e.g., Facebook's tentative plans to use WhatsApp user data [30]). End-to-end (E2E) encryption has been adopted in several messaging tools (e.g., WhatsApp, iMessage), whereas other tools have positioned security as a key selling point (e.g., Telegram, Signal). Security-related features may differ in how much they involve the user, whereas differences in the visibility of security features can create problems and impact user trust in a messaging tool [52], [53]. Telegram[1] is unique in offering separate modes of communication with differing levels of security. However, it may be difficult for users to distinguish between these modes and make effective use of them [31]. Users may explore the functionality of a messaging tool, or identify features that satisfy specific goals (which may or may not relate to security, such as sharing sensitive information with others). Users new to a security tool may also use it in ways that are not anticipated by developers [46].

Here, we explore the motivations and security behaviours of using a messaging tool that claims to be secure, specifically those who have not used Telegram before and those who are familiar with the tool. We combine two research techniques: (1) a novel lab-based user study with 11 novices and 11 participants with prior experience of using Telegram, and (2) a usability inspection bringing together cognitive walk-through and heuristic evaluation, focusing on Telegram's UI. This approach has been applied before in the area of usable security, most notably by Whitten and Tygar [62] to evaluate PGP 5.0. Here, we have planned a lab-based study that uses a set of tasks to elicit user perceptions of Telegram. The usability inspection complements this by allowing us to look at issues not touched upon by those tasks or not reported by our participants.

Prior work has focused on novices, with the admirable goal of identifying barriers to adoption [52], [62]. Studies of secure communication tools have rarely involved non-novices, where these users can identify the motivations for adopting and using security features in practice. Participants brought their mobile devices to the lab. Novices installed Telegram to explore its features by way of a 'sensitive payment information' messaging scenario. Prior users of Telegram were similarly involved in the task, but as an opportunity to see how they have used the tool and the role of Telegram's various security features in these practices, such as the *Secure Chat* mode. In both cases, scenario tasks were used to promote discussion as part of semi-structured interviews. Use of a System Usability Scale (SUS) questionnaire further explored the usability of the tool for novices and users alike. We found that both groups

*Authors contributed equally.

‡The study was conducted while the author was at University College London (UCL).

EuroUSEC '17, April 29, 2017, Paris, France

http://dx.doi.org/10.14722/eurousec.2017.23006

[1] https://telegram.org/

When asked about encryption, six participants (three novices and three users) provided explanations relating to security and safety. These included *"an extra barrier of security"*, *"more time is needed to know the content of the message"*, and *"making chats safe from hacking until they get deleted from the servers."*

Questions