Introduction to Software Security

Paolo Falcarin

Ca' Foscari University of Venice

Department of Enviromental Sciences, Informatics and Statistics

paolo.falcarin@unive.it

CM0626 – Software Security

CM0631-2 – Software Security

11 February 2025

About me

- Degree / PhD / Post-Doc at Politecnico di Torino
  - Visiting PhD student at ETH Zurich -2003
  - Visiting Lecturer at Tongji Shanghai – 2009
- Senior Lecturer / Reader at the UEL
  - Visiting researcher at University College London -2012
  - Visiting Lecturer at Hangzhou Dianzi - 2016
- Now Associate Professor at Ca' Foscari University of Ver
  - Visiting professor at University College London -2024
  - Visiting professor at Università Svizzera Italiana -2024

Professional Interests

Ca' Foscari University of Venice

- Software Protection
- Software and Attack Modelling
- AI for Security and Software Engineering

Personal Interests

Ca' Foscari University of Venice

- Travel (20 countries so far..)
- Books, Comics
- Music and Cinema
- Chess
- Football

Course Contents

Ca' Foscari University of Venice

- Intro to Software Protection - MATE scenarios
- Reverse Engineering
- Binary Code Analysis
- Obfuscation
- Tamper-proofing
- Watermarking
- Attack modelling
- E-Voting, human aspects, Cybercrime
- Malware Analysis
- Privacy

Assessment

Ca' Foscari University of Venice

- Written Exam (90 minutes)
- Mandatory project (up to 4 bonus marks)
  - Research paper presentation (individual)

  OR
  - Software Tool presentation (individual)

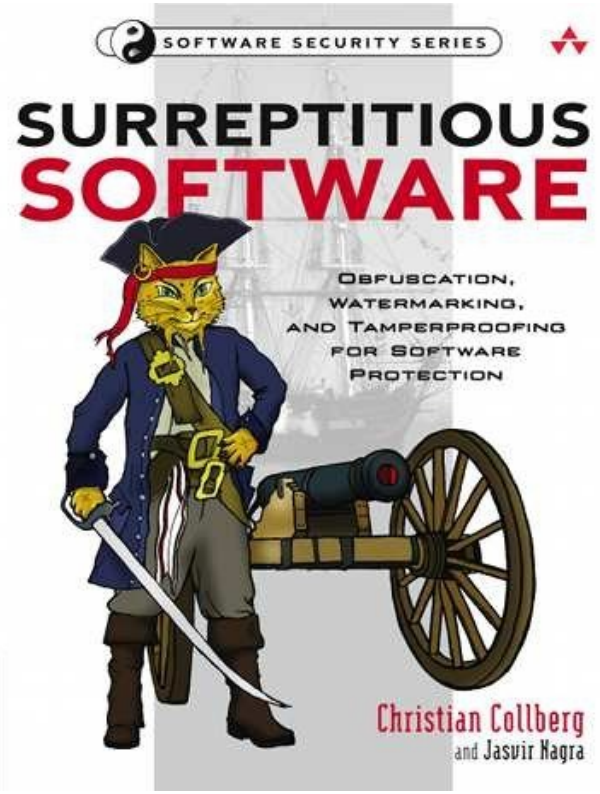
  OR
  - Small practical project (max 2 students per group)

- Projects presentations on **Friday 9th May**

Book and Resources

Ca' Foscari University of Venice

- My slides are mostly taken from Collberg textbook
  - Christian Collberg, Jasvir Nagra; Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection, 2009, Addison-Wesley Professional
- Kali Linux VM on VirtualBox

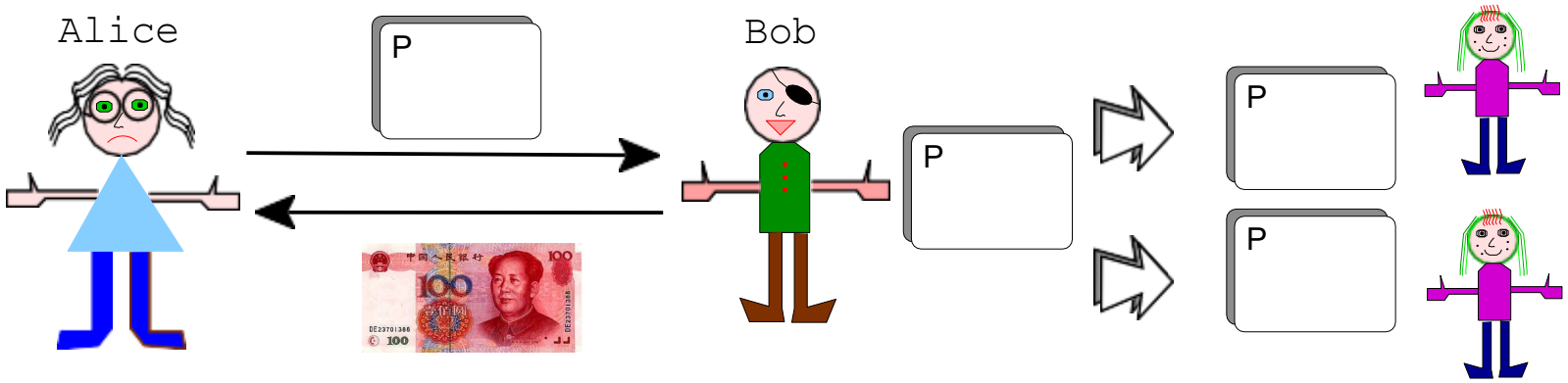


- Few sections taken from:
  - Adam Shostack; Threat Modeling: Designing for Security, 2014, Wiley
  - Chris Eagle, Kara Nance; The Ghidra book: the Definitive Guide, 2020, No Starch Press
  - Vijay Kumar Velu, Robert Beggs; Kali Linux for Advanced Penetration Testing- Third Edition, 2019, Packt Publishing

MATE Scenarios

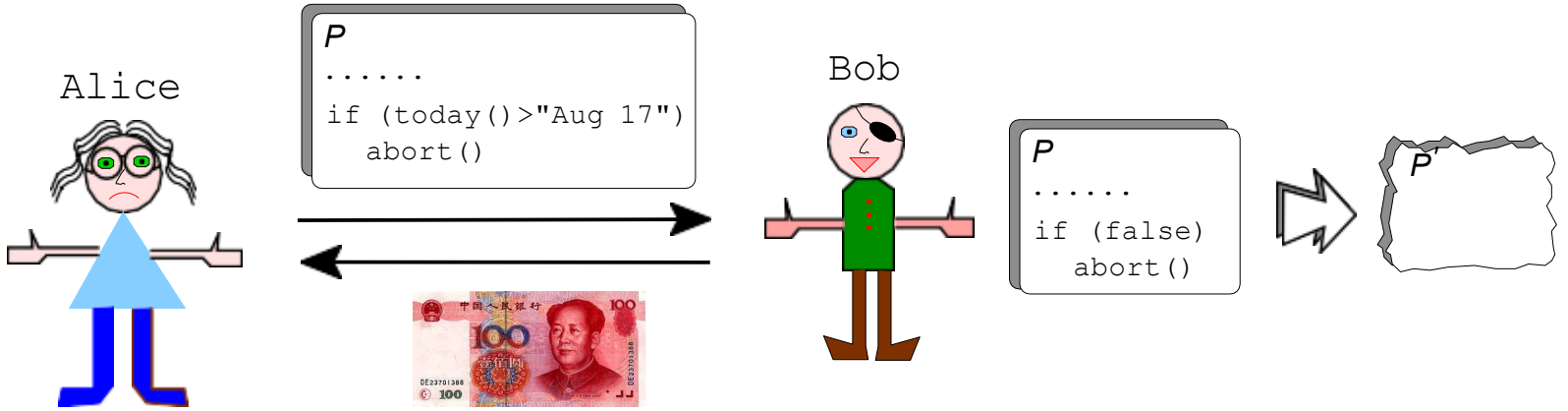
Credits: Christian Collberg

Software piracy



Alice is a software developer.
Bob buys one copy of Alice's program.
Bob illegally sells copies to his friends.

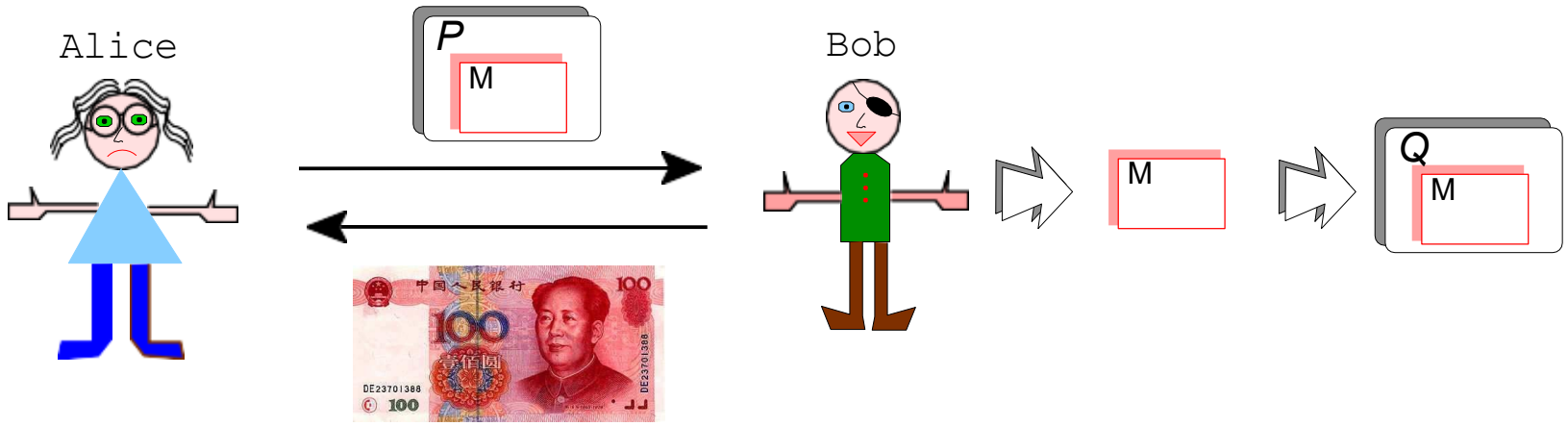
License check tampering



Bob removes license checks to be able to run the program whenever he wants.

Alice protects her program so that it won't run after being tampered with.

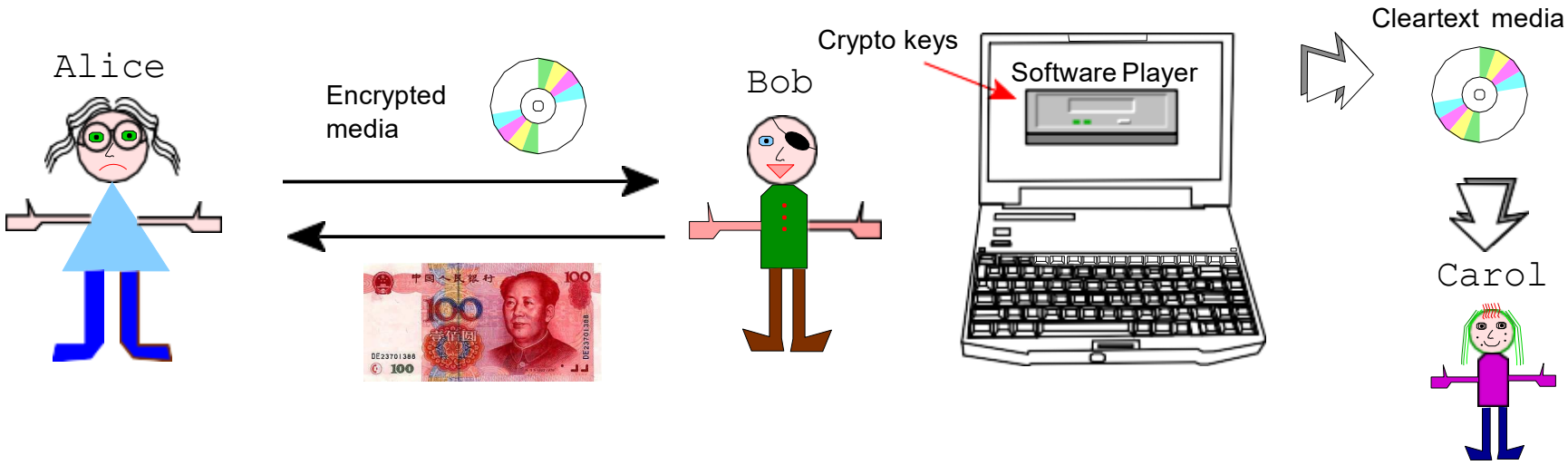
Malicious reverse engineering



Alice's program contains a valuable trade secret (a clever algorithm or design).

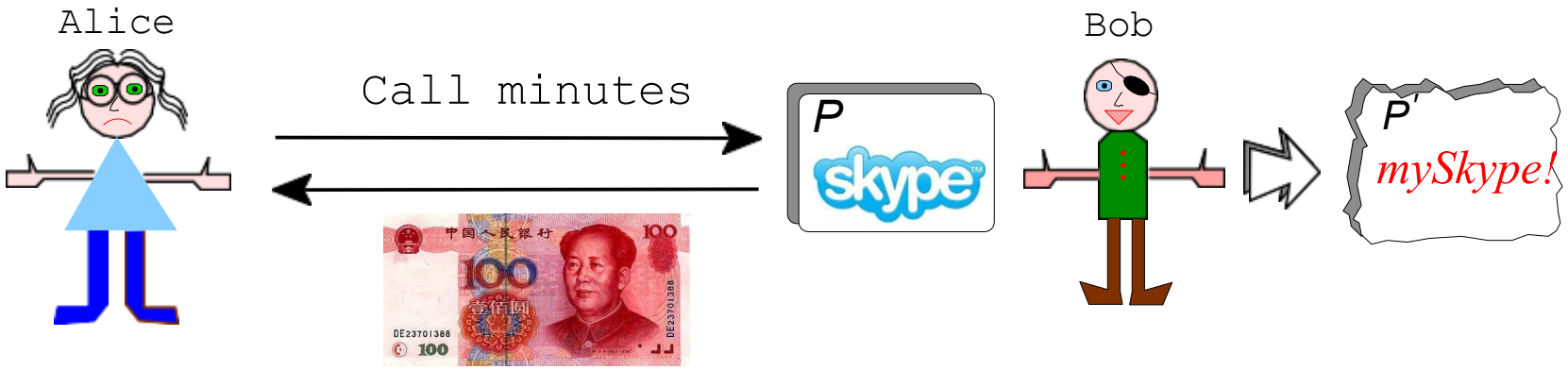
Bob, a rival developer, copies M into his own program (code lifting).

Digital rights management (DRM)



A DRM media player contains cryptographic keys that unlock and play encrypted music files.

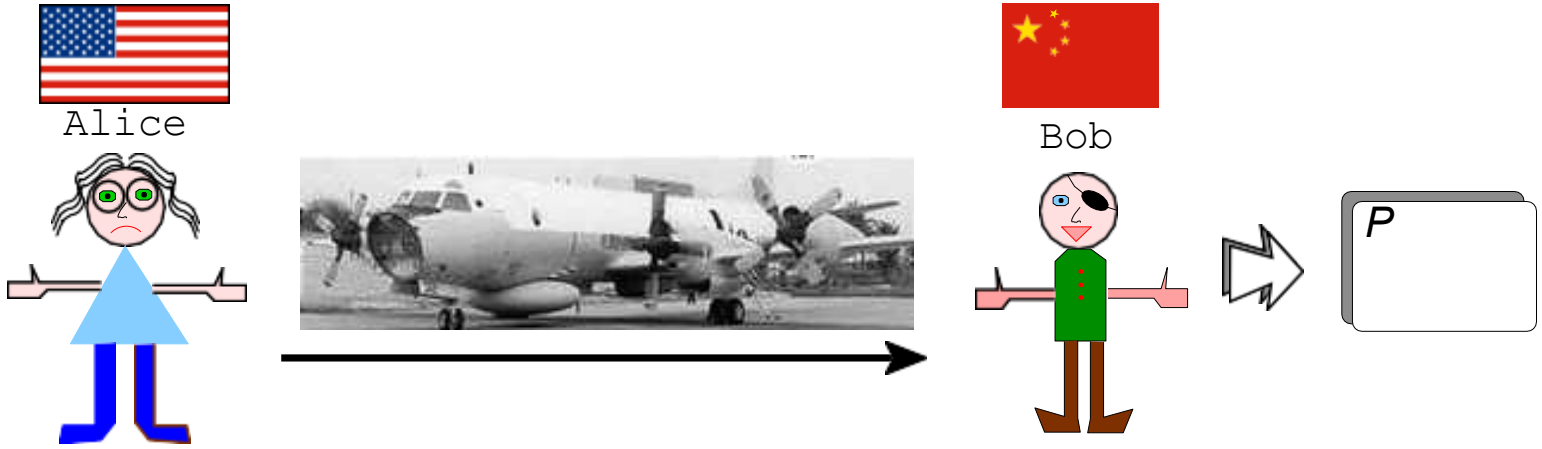
Protocol discovery



Alice sells voice-over-IP call minutes.

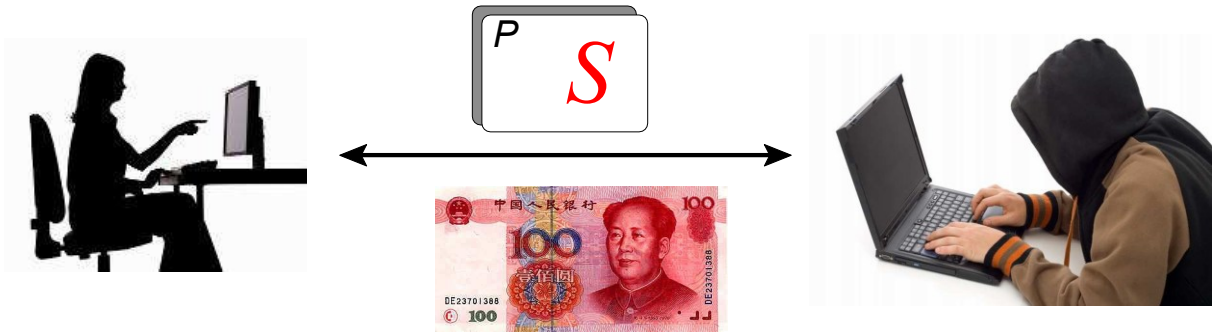
Bob examines the VoIP client to discover proprietary protocols to build his own rival client.

Protecting military software



The military want to be able to track classified software.
In 2001, an EP-3 spy/reconnaissance plane landed on Hainan Island in China after a collision.
The crew was unable to destroy all equipment.

The Man-At-The-End Problem

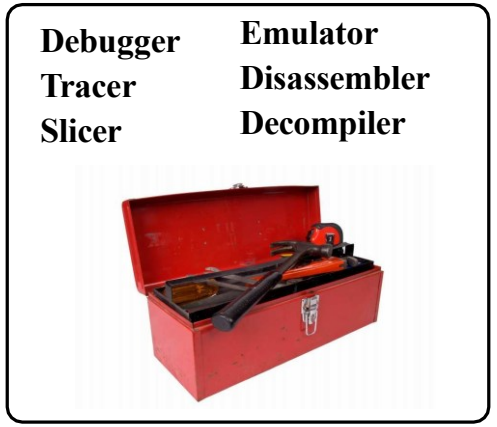


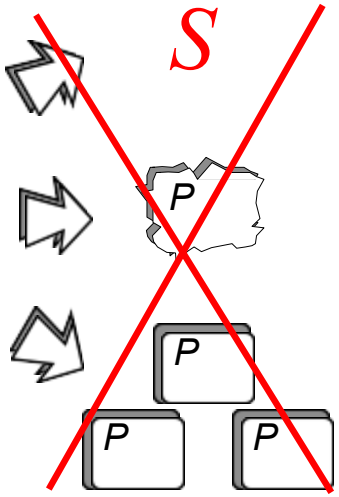
The Man-At-The-End Problem

The Man-At-The-End Problem

The Man-At-The-End Problem

Ca' Foscari University of Venice

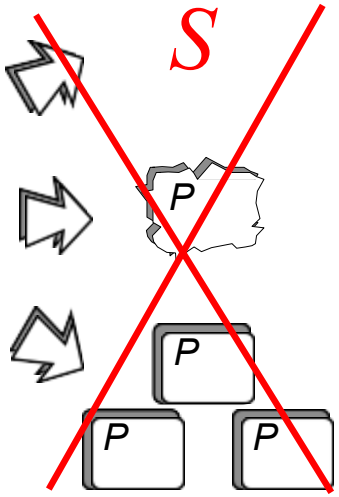




The Man-At-The-End Problem

Ca' Foscari University of Venice

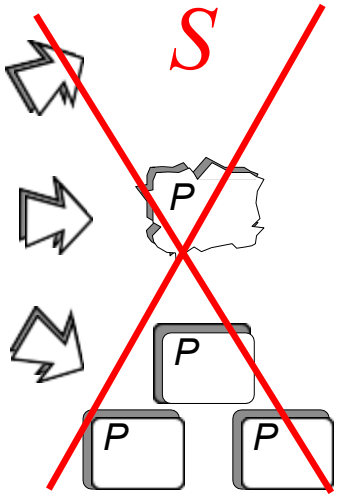
```
1. Static/Dynamic Analysis
2. Modify
3. Test
4. Did it work?
```



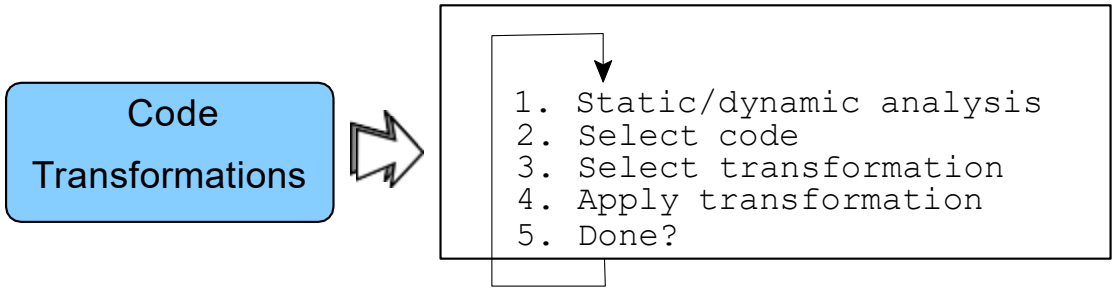
The Man-At-The-End Problem

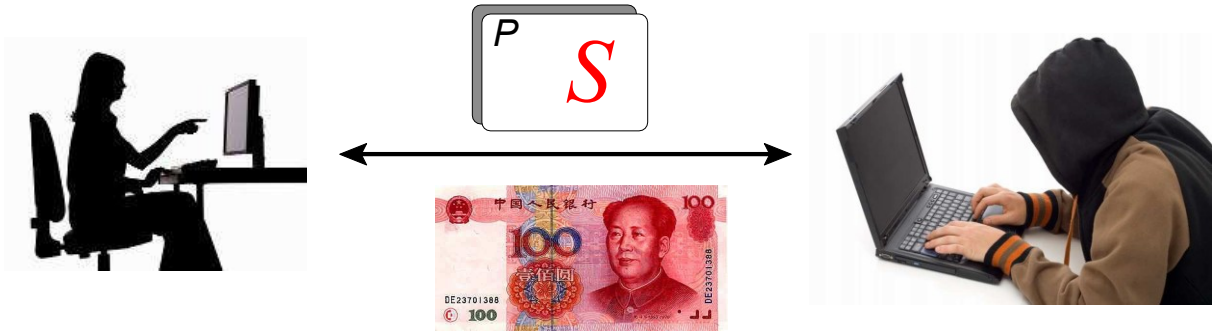
Ca' Foscari University of Venice

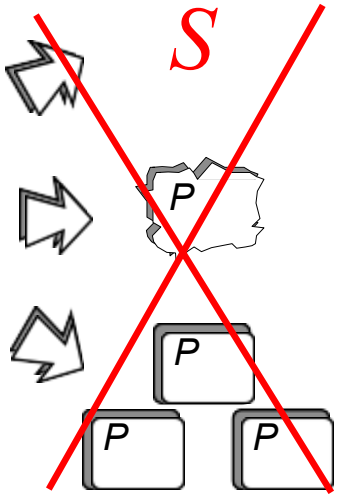
```
1. Static/dynamic analysis
2. Select code
3. Select transformation
4. Apply transformation
5. Done?
```



The Man-At-The-End Problem







MATE attacks

Ca' Foscari University of Venice

Man-At-The-End (MATE) attacks occur in any setting where an adversary has physical access to a device and compromises it by inspecting, reverse engineering, or tampering with its hardware or software.

Exercise

Ca' Foscari University of Venice

- Describe another few situations where a MATE attack can occur!

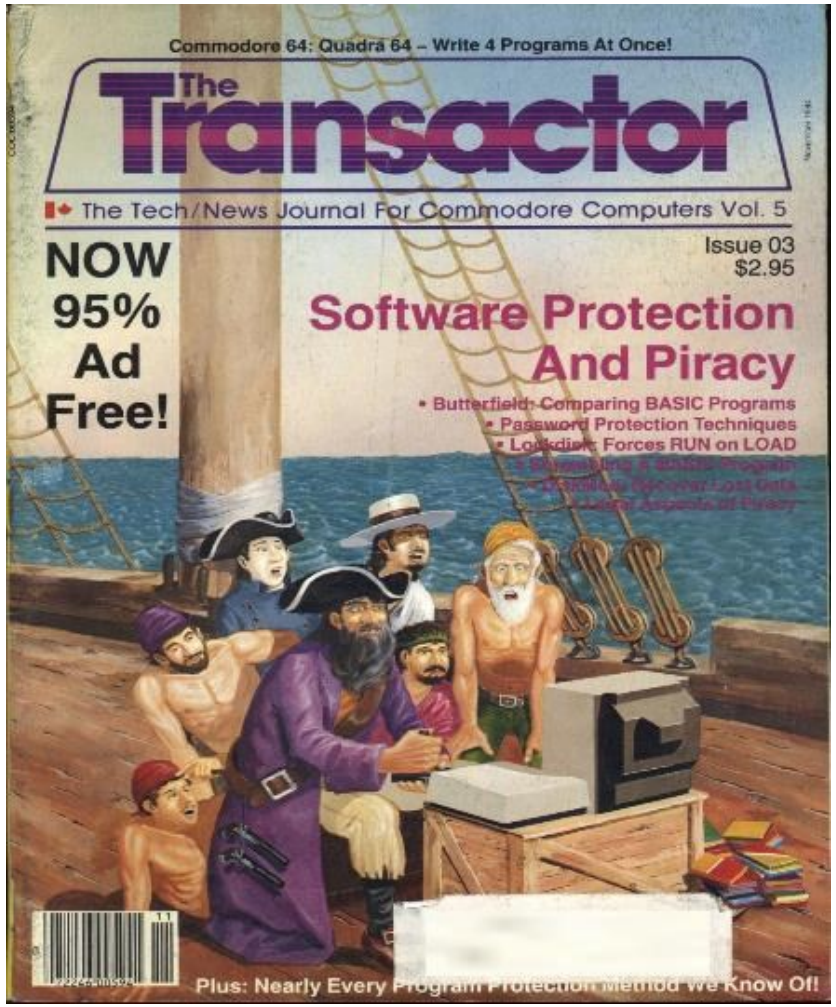
1. ✏️

2. ✏️

Software Protection

What is Software Protection?

Ca' Foscari University of Venice



- Not a New Problem
  - Canadian Journal
  - Commodore 64…
  - Nov. 1984 !!!

- Security research usually protects user and host from malicious programs
- In this case we need to **protect programs from malicious users on untrusted hosts**

Security Attacks

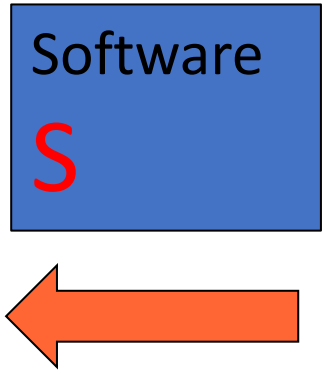
Ca' Foscari University of Venice

- Alice works behind a Network Firewall
- A virus-scanner protects her from viruses, worms, Trojan horses
- Intrusion Detection System analyzes network to detect if "Bob the hacker" is doing something suspicious

Man-At-The-End attacks

Ca' Foscari University of Venice

- Let's invert the situation: Alice sells Bob a program to run containing a secret S
- Bob can gain some economic advantage over Alice by extracting or altering S
- Encryption does not help in this case!!



Protection from What?

Ca' Foscari University of Venice

- Piracy of the software itself
  - Unlicensed copies
- Piracy of data viewed using the software
  - Movies, e-books, digital rights management
- Theft of secrets in the software
  - Crypto keys
- Theft of Intellectual Property
  - Reverse engineering - Code-lifting (Reuse "as is")
- Unauthorized modification
  - Remove protection or add (malicious) functionalities

Man-At-The-End (MATE) Attacker

Ca' Foscari University of Venice

- Can have physical access to a device
- Can inspect and reverse engineer software
- Can tamper with its hardware or software

Applications

Ca' Foscari University of Venice

- Software Licensing
  - Security kernel on each client user device controlled by license server
- Multimedia Content consumption
  - Digital Rights Management on many platforms
- Mobile Apps on SmartPhones
  - No secure hardware => Software-only protections

Example: Code & Content

Ca' Foscari University of Venice

```
set_top_box() {
        if (bob_paid("SkyTV"))
                allow_access();
        if (hardware_is_tampered()
        ||
        software_is_tampered()
        ||
        bob_is_curious()
        ||…)
        punish_Bob();
}
```

Example: Content

Ca' Foscari University of Venice

```
int DigitalRightsMgmt () {
        movie_data=download();
        key=0x38…;
        movie =decrypt(key, movie_data);
        play(movie);
}
```

Example: License

Ca' Foscari University of Venice

```
int main () {
        if (
                printf("License expired!");
                exit;
        }
}
```

false

Possible Hacks and Tools

Ca' Foscari University of Venice

- Extract Code!
- Discover Algorithms!
- Find Design!
- Find Keys!
- Modify Code!

Static Analysis
Dynamic Analysis
Disassembly
Decompilation
Slicing
Debugging
Emulation

Software Piracy

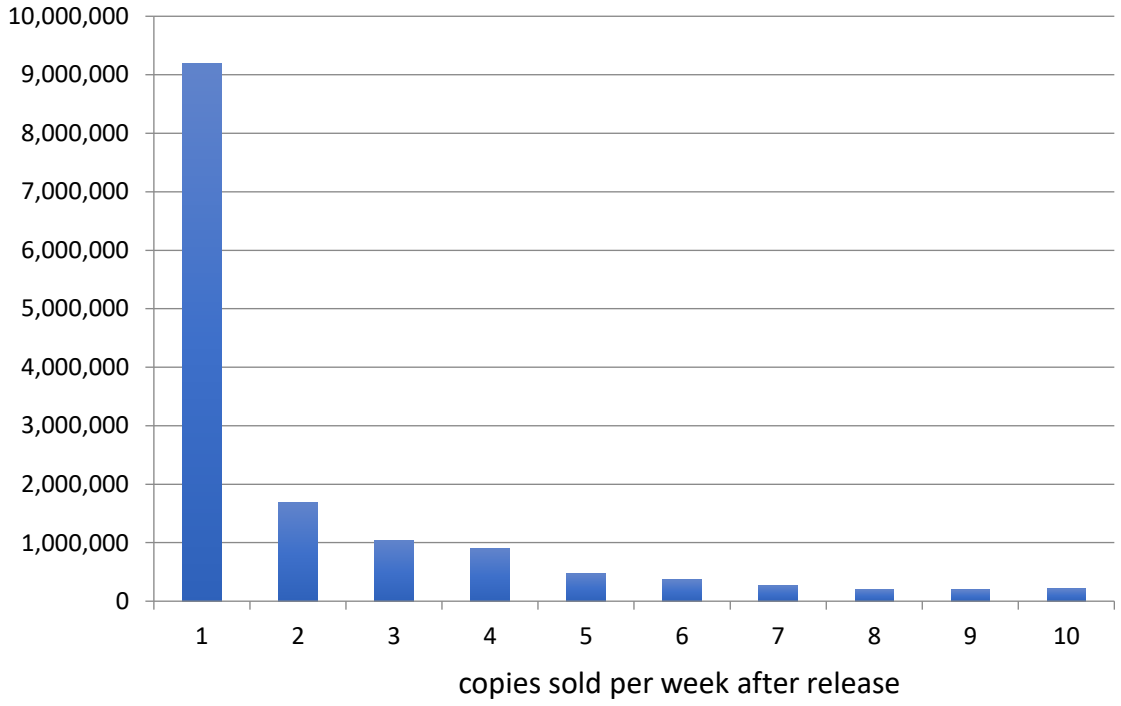
Ca' Foscari University of Venice

- Alice is a software developer.
- Bob buys one copy of Alice's application and sells copies to third parties.

- => Alice watermarks / fingerprints her program

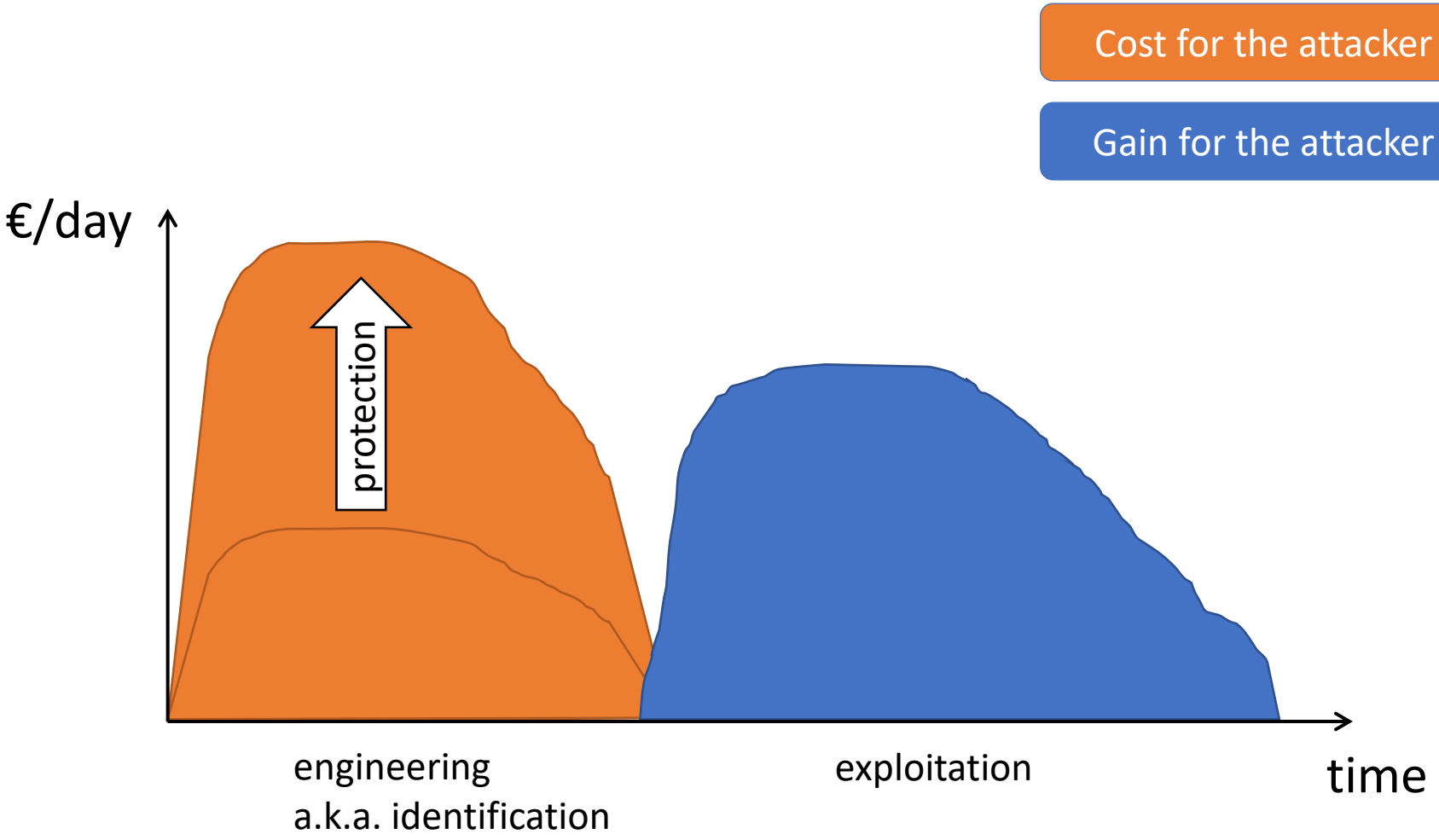
Economics of MATE attacks

What is the value of protection?



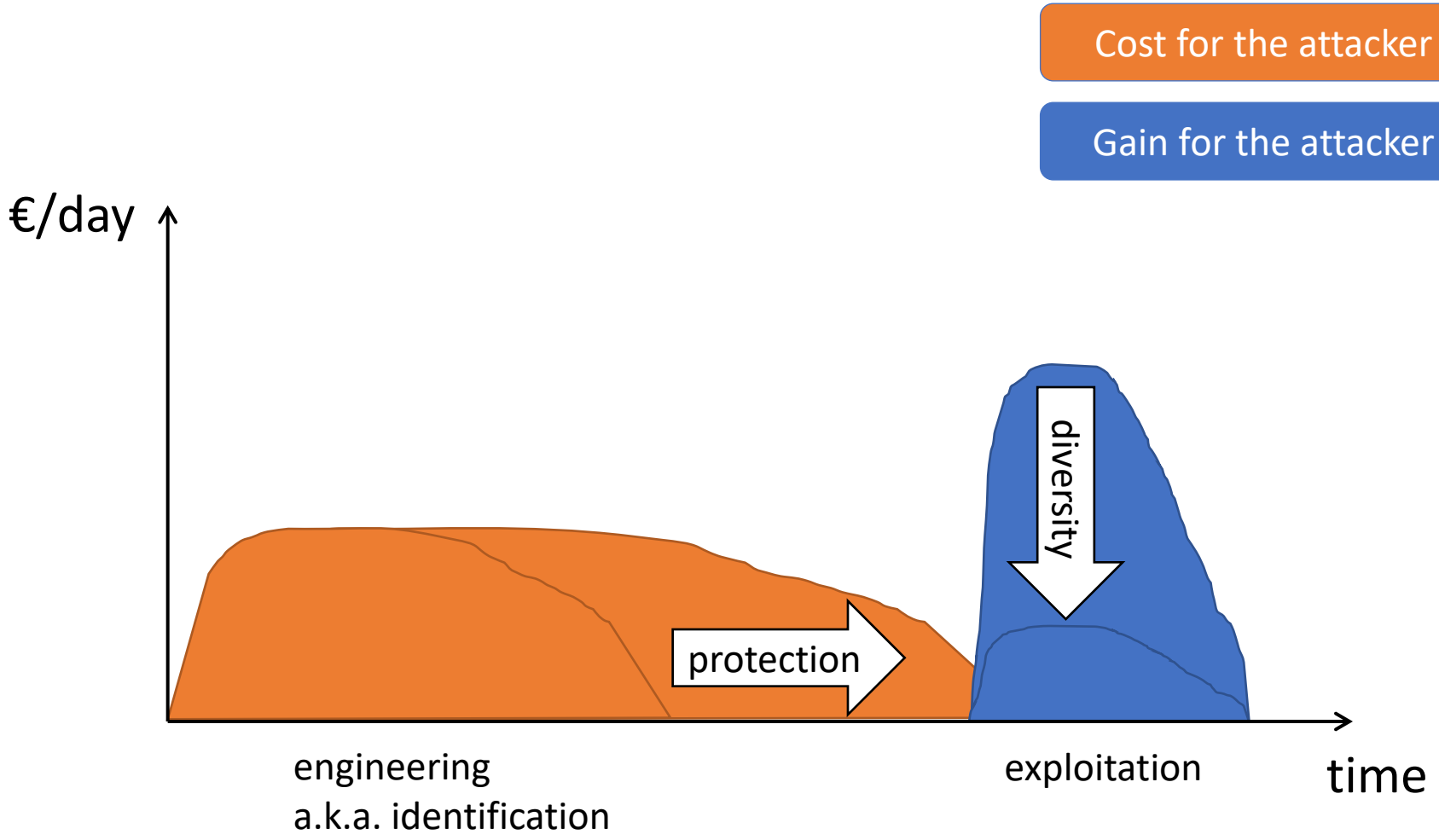
Economics of MATE attacks

Ca' Foscari University of Venice



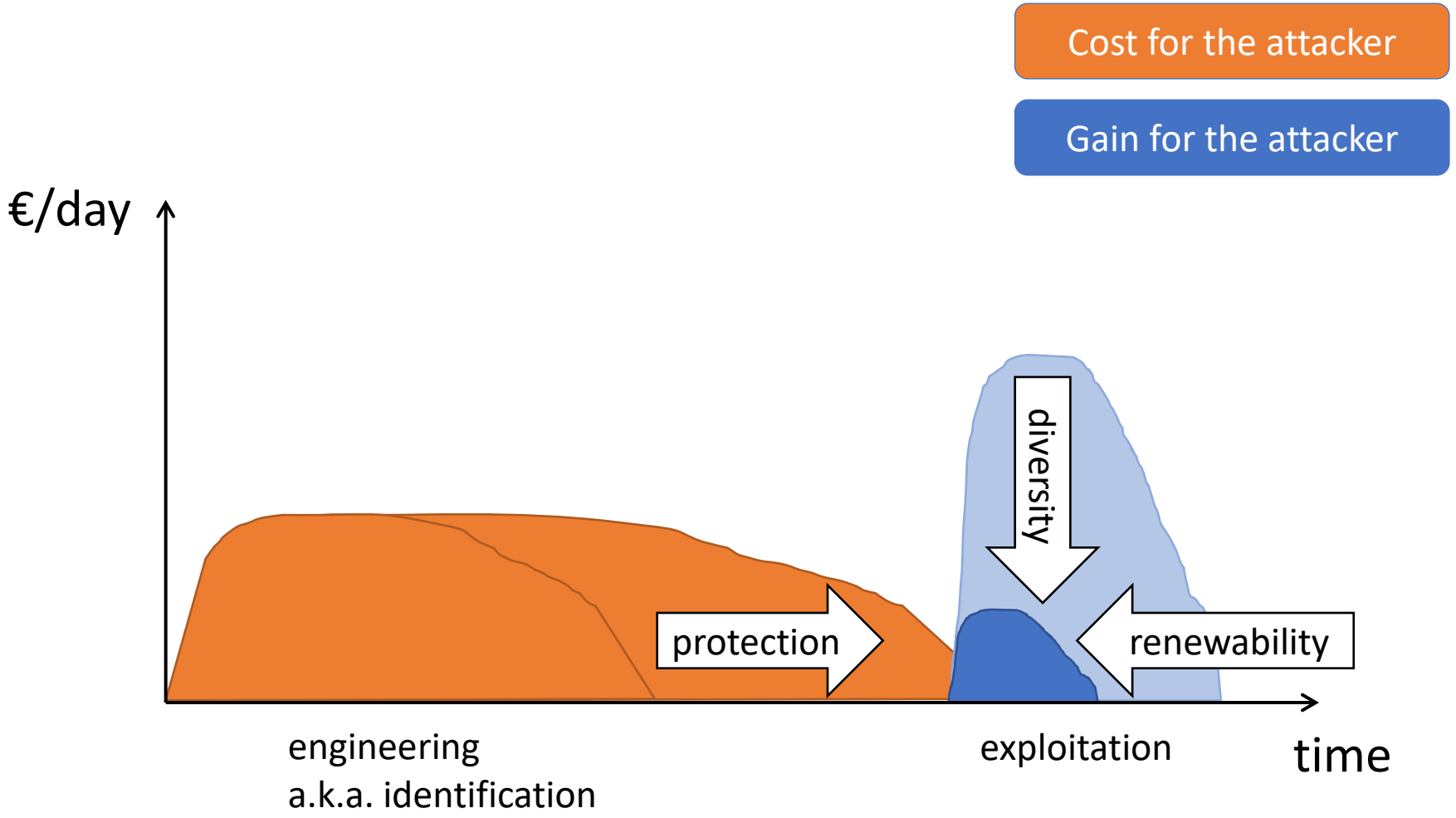
Economics of MATE attacks

Ca' Foscari University of Venice

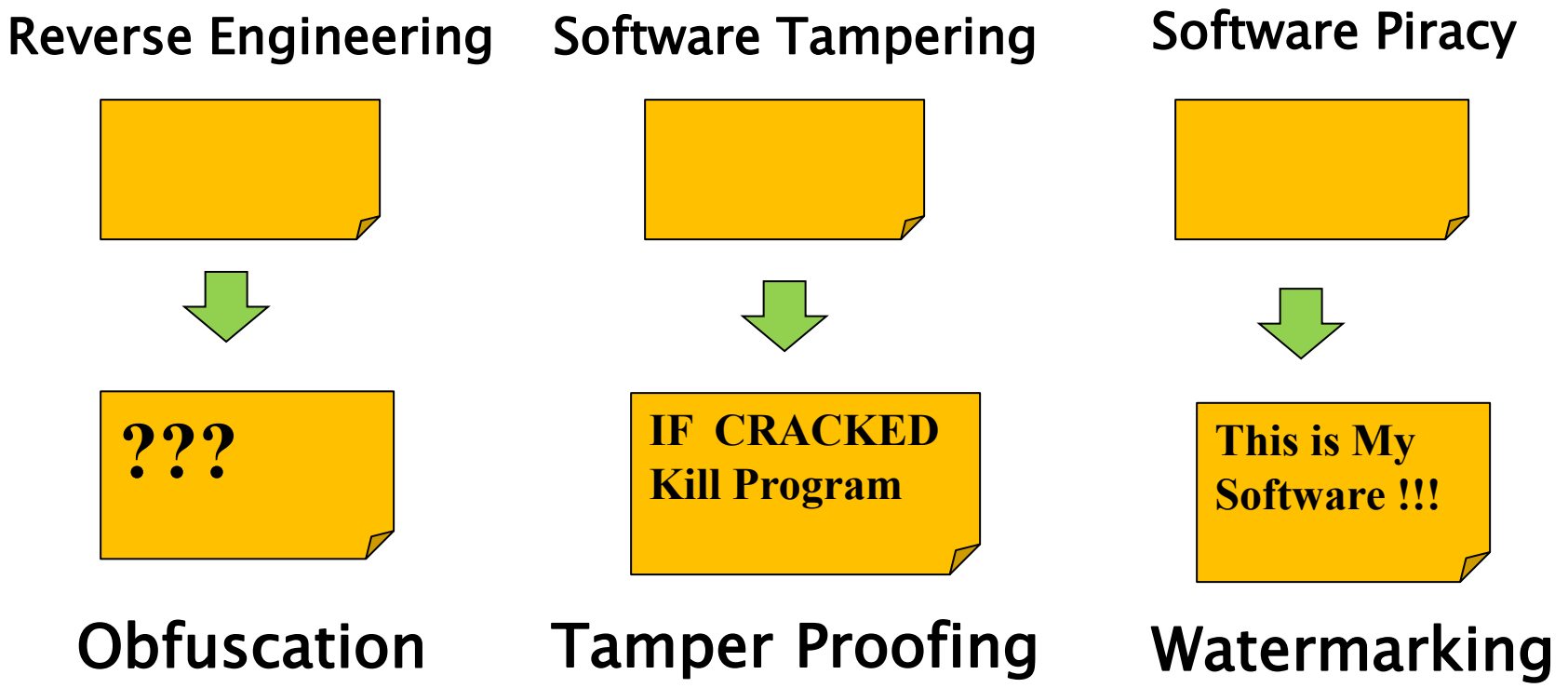


Economics of MATE attacks

Ca' Foscari University of Venice



MATE attacks => Solutions ?



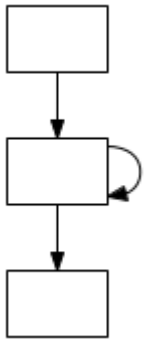
- Techniques are code transformations
- Tools compile unprotected programs to protected programs

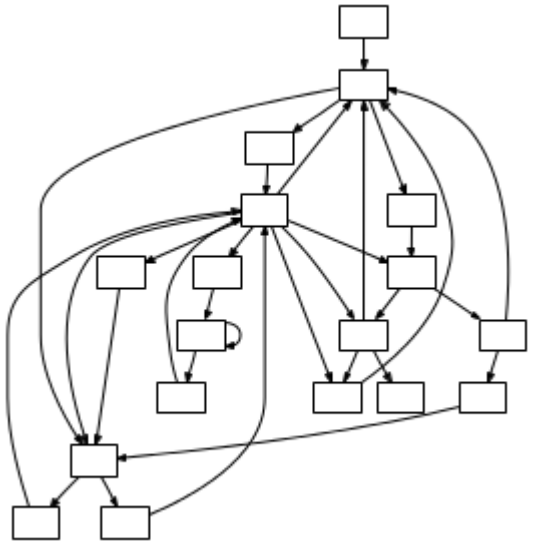
Obfuscation

Ca' Foscari University of Venice

- Obfuscation transforms a program into a new program which:
  - Preserve same functionality of original program
  - More time consuming to reverse engineer
  - More difficult to use automated tools
  - Minimum overhead

original





obfuscated

Obfuscated Language?...

Ca' Foscari University of Venice

- Aoccrdrinig to rscheearch at Cmabridge Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoatnt tihng is taht the frist and lsat ltteer be at the rghit pclae.
- The rset can be ttaol mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.

Code Obfuscations

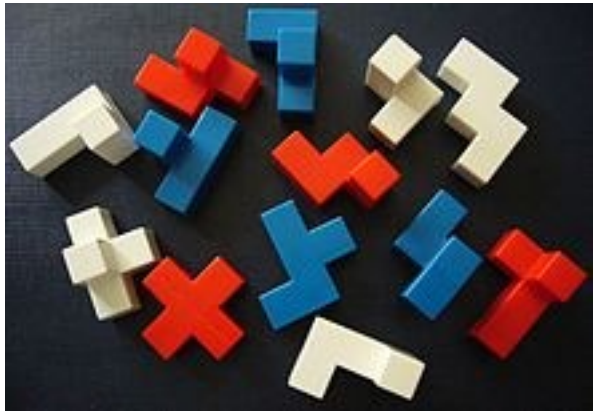
Ca' Foscari University of Venice

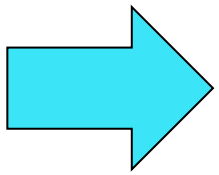
- Lexical transformations
  - Modify variable names (ID-Renaming)
- Control-flow transformations
  - Opaque Predicates -Redundant Code
  - Increase Indirection Levels
- Data transformations
  - Modify data structures
- Anti-disassembly (Mess with binary code)
- Anti-debugging
- Code/Data Encryption

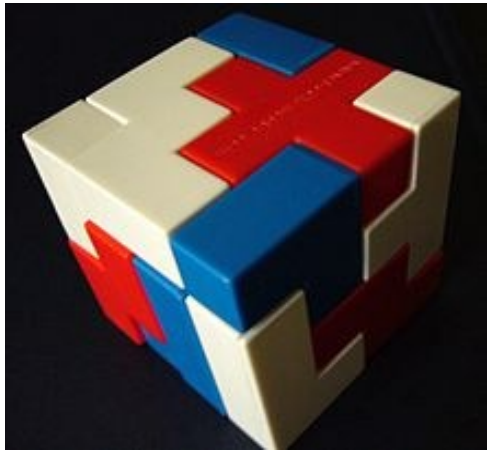
Obfuscated C Code Contest

Ca' Foscari University of Venice

- IOCCC stands for "International Obfuscated C Code Contest": http://www.ioccc.org/
- It is an annual contest to see who can write the most unreadable, but legal C program.
- Example: Bedlam Cube solver







Example of Obfuscated C Code

Ca' Foscari University of Venice

```
/* <body  bgcolor  = 0 >
                    <img src= cube.gif><!-- */
                  #define _ ] [ /* ; ; ; ;  */
                 #include <string.h>  /* ; ; */
                #include  <stdio.h>  /* ; ; ; */
                #define K(o,O) L o=0; o<O ; o++)
                #define  H  unsigned  long  long
                #define /* ; ; ; ; ; ; } */ W ]=
                #define /* ; ; ; ; ; ; */ L for(
                #define /* ; ; ; ; ; ; */ J if (
                #include <stdlib.h> /*; ; ; ; */
                #define Z (j*3+j/9*3+2)%10+j/9*9
                int s[ ] = { 186, 94, 1426, 3098
                ,1047 , 122 , 1082 , 3083 , 1039
                , 569 , 527 , 1054 ,  531  } ;
                #define P( o , O ,I) K(C,o)\
                fputc ( ( O )[ C ] -I, G);
                #define Y strncpy( /* */


      int j,k,l,v,c,C,O[64],n[
    64],*o,q[13 _ 13],u,d,f,g[
   W{ 8,7,6,6,6,6 } ; H p [13 _
  #define M memset(E[c]+j*298+v\
*a+88-u*5+586*( u*5+152-i/16*a+C
13 _ 432],r,w,t,b,S[13]; FILE*G;
char E[13 _ 168840],*A="||||||",
*D="   {   ;   wb;  aa; aaaa}a  \
0z00Zzz} { z0z}      ",Q[64 _ 60
],*F ="+    +",T [43]; int main(
int I,char**V){ int i,h,B,a,m; H
#define R(z){ x=h=0; K(j,27)h|=\
(c>> j&1 )<<z ; ; ; } c=h; ; ; ;
x; J I>1)B=C=atoi(V[1]); J !d)K(
v,13){ h=s[v]; J B<0){ k=h>>18
^h&511; h^= k<<18|k; C=-C; }
K(k,7){ J k==4)R(Z)R(j+(j-
" /@"[j/9]+38)/3*6-6)K(I
```

```
                ,4){ R(Z)K(a,9) x|=((H)c
                  >>a*3&7)<<a*4 +a/3*4; K(a,
                 96){ m=a-37; r=m<0?x>>-m:x<<
                m; J!(x!=(m<0?r<<-m:r>>m)||r&r
               /2&0x888888888888888ULL||r&r>>4&
               0xF000F000F000ULL)){ p[0 _ v _ q
               [v _ 0]++W r; K(j,q[v _ 0]-1)J p
               [0 _ v _ j]==r)q[v _ 0]--; } } }
               } } h=d; x=w; m=q[f _ h]; a=f; u
               |=1<<a; B=u; K(i,m){ w|=S[a W p[
               d _ a _ i]; r: d++; v=*s; t=w; L
               j=1; j<13; j++){ J u>>j&1^1){ b=
               c=k=0; l=q[j _ h]; L; k<l; k++){
               r=p[h _ j _ k]; J!(w&r))b|=p[d _
               j _ c++W r; } J!c) goto n; J c
               ==1){ w|=S[j W b; u|=1<<j; J
               d==12){ J--C<1){ K(c,64)O[
               c W-1; x=0; K(c,13){ r=~


      0; K(j,13)J S[j]<r&&S[j]
    >x)r=S[j]; K(i,64){ n[i W-
   1; memset(Q[i],32,59); } K(i
  ,64){ J r>>i&1)O[i W n[i W c ;
K(j,2){ o=j?O:n; k=o[i]; u=i&d ;
v=i&3; a=48; J k+1){ C=k==c; Y T
,"+----+/    /|+-----+ ||    | +\
|    |/+-----+",43); J!C&&v&&o[i-
1]==k)T[6 W T[21 W T[29 W 32 ; J
u&&o[i-4]==k){ Y T,C?"//////":F,
6); J v==3||o[i-3]<0)T[d W T [20
W C?47:32; } J i&a&&o[i-16]==k){
J u==d||o[i-d]<0)Y T+36,C?A:F,6)
; J!C)J v==3|| o[i-15]<0) T[35 W
32; } J C){ Y T+7,"////",4); T
[19 W T[27 W 47; Y T+22,A,4)
; Y T+30,A,4); } K(C,6)Y Q
[9-i/16*3+u/2+C]+v*5-u/2
```

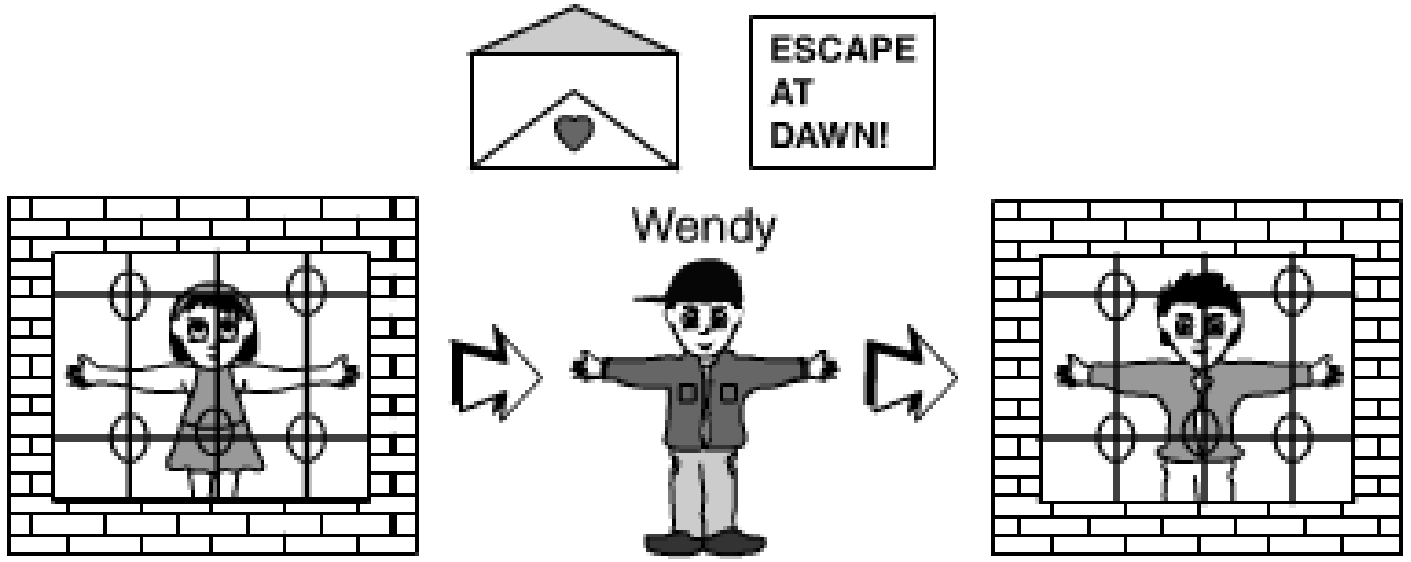
```
              +j*30+g[C],T+"06=EMT"[C]
               -a,20-g[C]-g[5-C]); K(C,a)
              { M)+28,m=k+33,h=20); M+h)-h
             ,k+49+u/4*16,a); } K(C,h){ M)-
            C,k+17,a); M)+a-C,m,C); M+a)+28,
            m,h-C); } } } } K(i,21)puts(Q[i]
            ); x=r; } G=fopen("cube.gif", D+
            11); P(13,"qspbc\213t,J+ **",42)
            K(i,8)P(a,D,h+" H Zbjm "[i]+C)P(
            19,"F$0sjyxhfujWSU(&%%%",37)K(i,
            13){ P(19,"K#.3 ***V****t,J+*1",
            42)K(j,1340){ P(2," !",161)P(126
            ,E[i]+j*126,0)}P(3+i/d,"!\241 ["
            ,32 ) } exit(0); } goto n; }else
            goto r; } J c<v) { f=j; v=c; }
            q[j _ d W c; t|=b; } } J!~t)
            main(0,0); n:u=B; w=x; d=h
            ; } u^=1<<a ; } /*--> */
```

- Example Bedlam Cube Solver http://www.ioccc.org/

Steganography: Prisoners' problem

Ca' Foscari University of Venice

- Alice and Bob are planning a prison break by passing notes through the warden Wendy
- If Wendy finds out they will be put in solitary confinement ☹



Steganography: Prisoners' problem

Ca' Foscari University of Venice

- They can't use cryptography…
  - if Wendy sees a garbled message, she will become suspicious and put an end to communications
- They must hide secrets in innocuous-looking messages…

> Easter is soon, dear! So many flowers! Can you smell them? Are you cold at night? Prison food stinks! Eat well, still! Are you lonely? The prison cat is cute! Don't worry! All is well! Wendy is nice! Need you!):

Watermarking

Ca' Foscari University of Venice

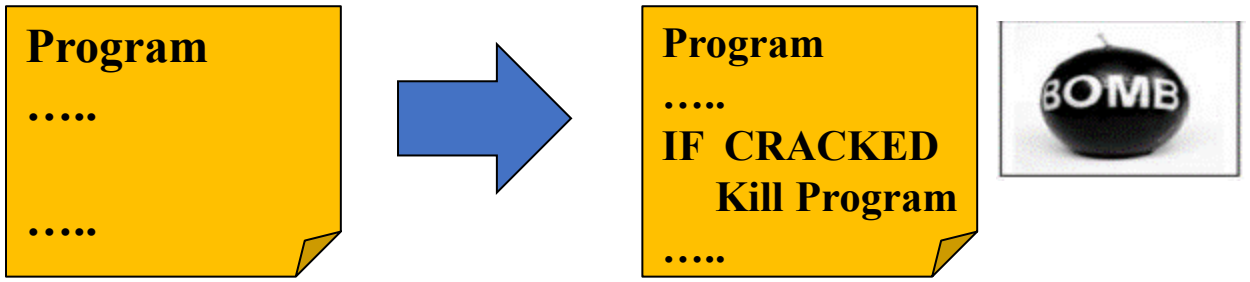
- Watermarking embeds a secret message into a cover message
  - E.g., Secret = copyright, Cover = code or digital image
- → Can prove ownership
- Fingerprinting embeds custom secret messages into cover messages
- → Can trace copyright violator

- Note Well: Watermarking is effective only if software can be inspected

Tamper-Proofing

Ca' Foscari University of Venice

- Tamper-proofing transforms code into a new code which:
  - Has the same semantics on expected input
  - Self-check itself and "explodes" when even binary is slightly modified
- Tamperproofing has two parts:
  - detecting that an attack has occurred
  - reacting to this attack.
- Reaction can be a combination of
  - self-destructing (in whole or in part) or stop service
  - Performance degradation



Tamper-Proofing example


```
int foo () {
        if (today > "Jul 27,2021"){
                printf("License expired!");
                if (hash(foo) != PRECALCULATED_HASH)
                    exit();
        }
}
```

Detect tampering and then:

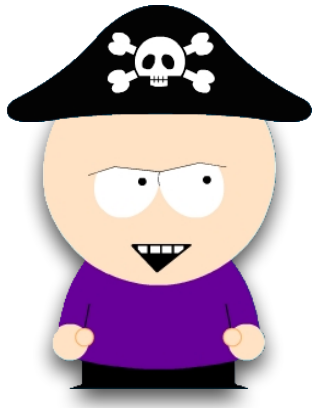
- ***crash the program***
- ***phone home***
- ***refuse to run***
- ***run slower***
- ***make wrong results***

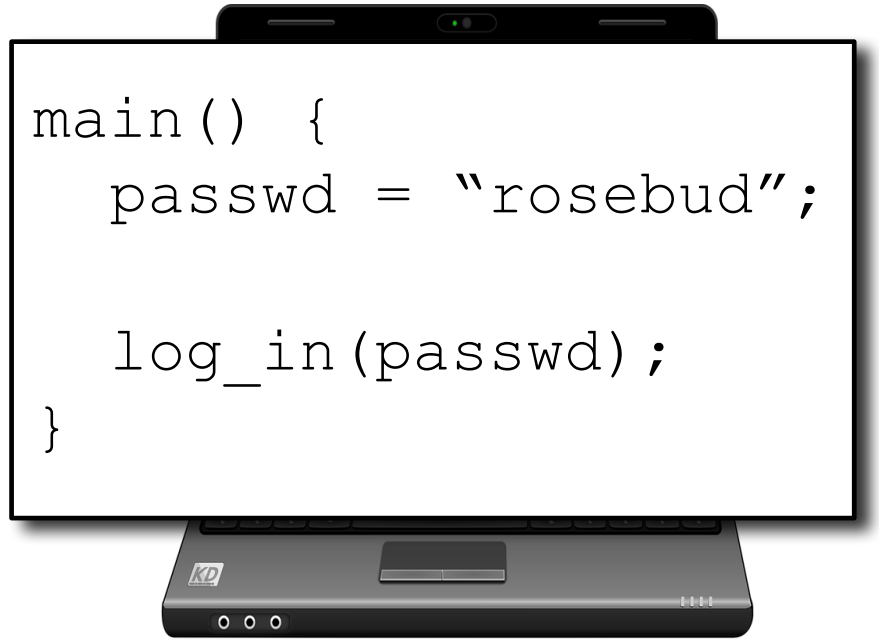
More MATE Scenarios

Credits: Christian Collberg

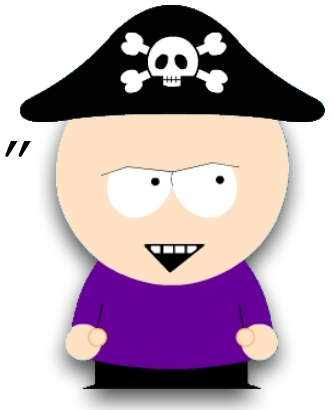
```
main() {
   passwd = "rosebud";

   log_in(passwd);
}
```

```
main() {
   passwd = "rosebud";

   log_in(passwd);
}
```



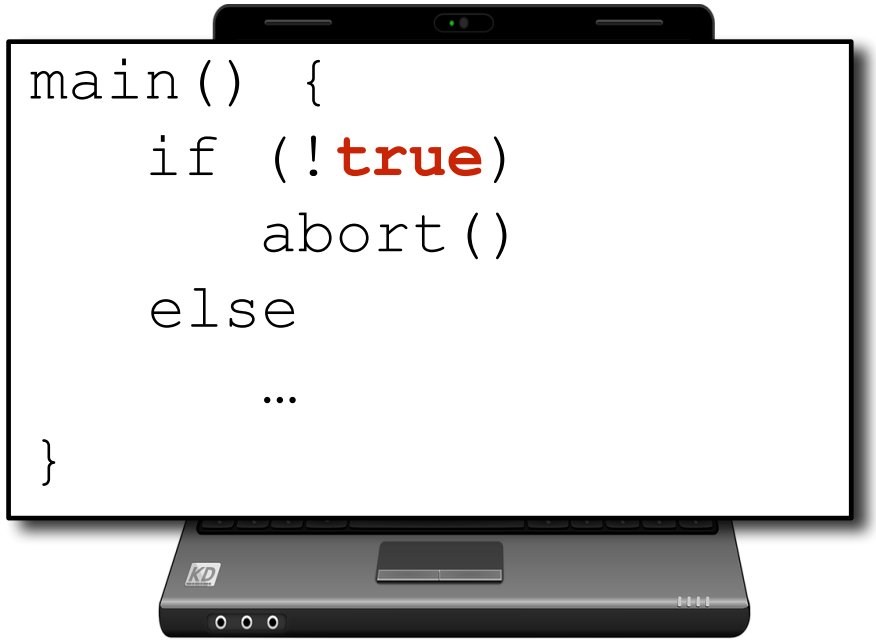


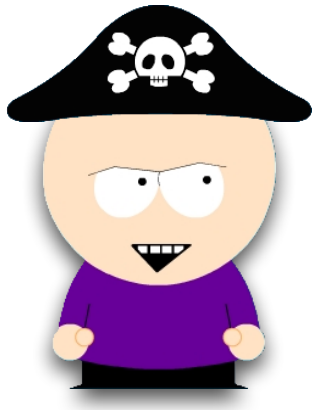
“rosebud”



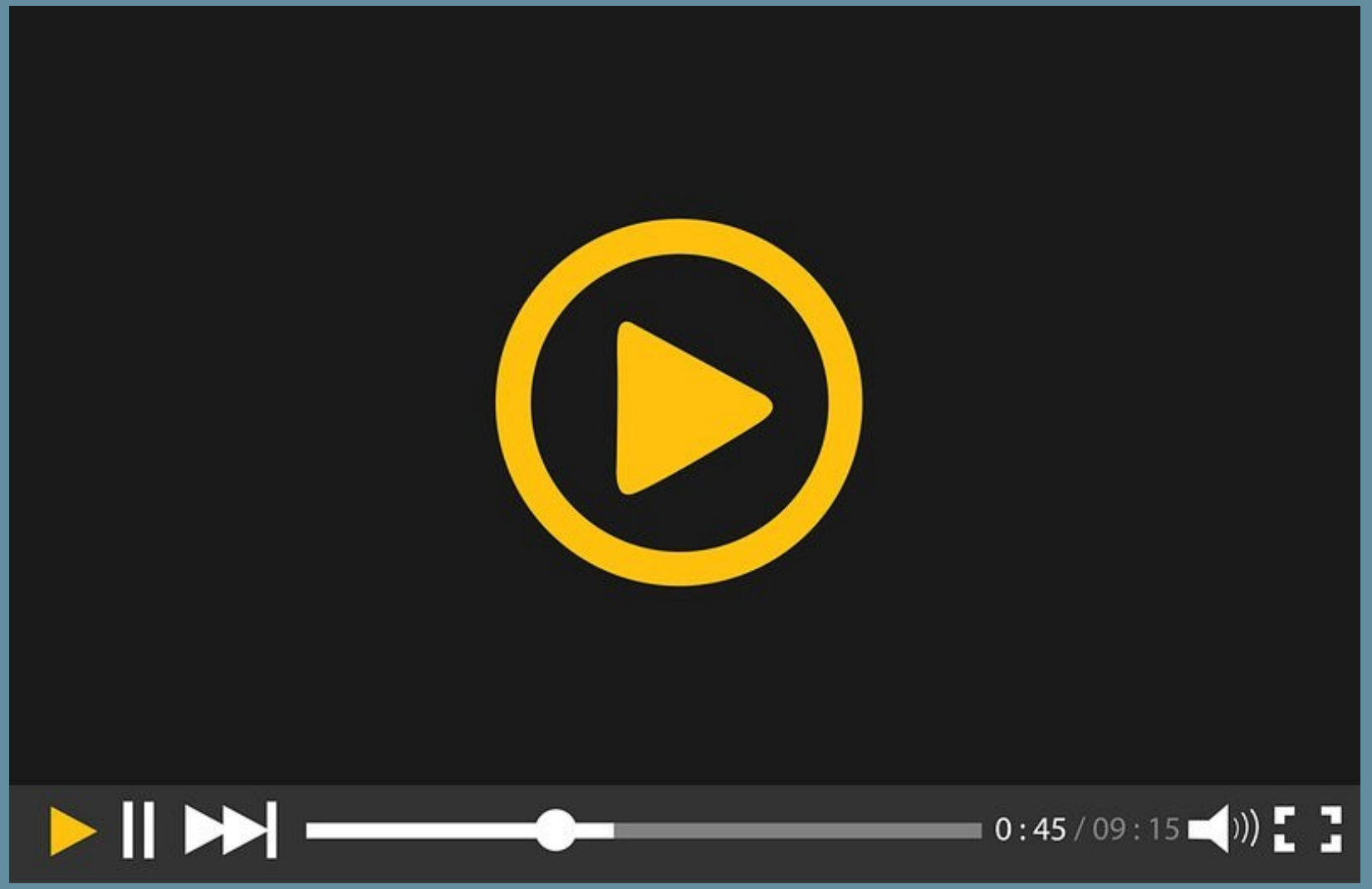
Confidentiality

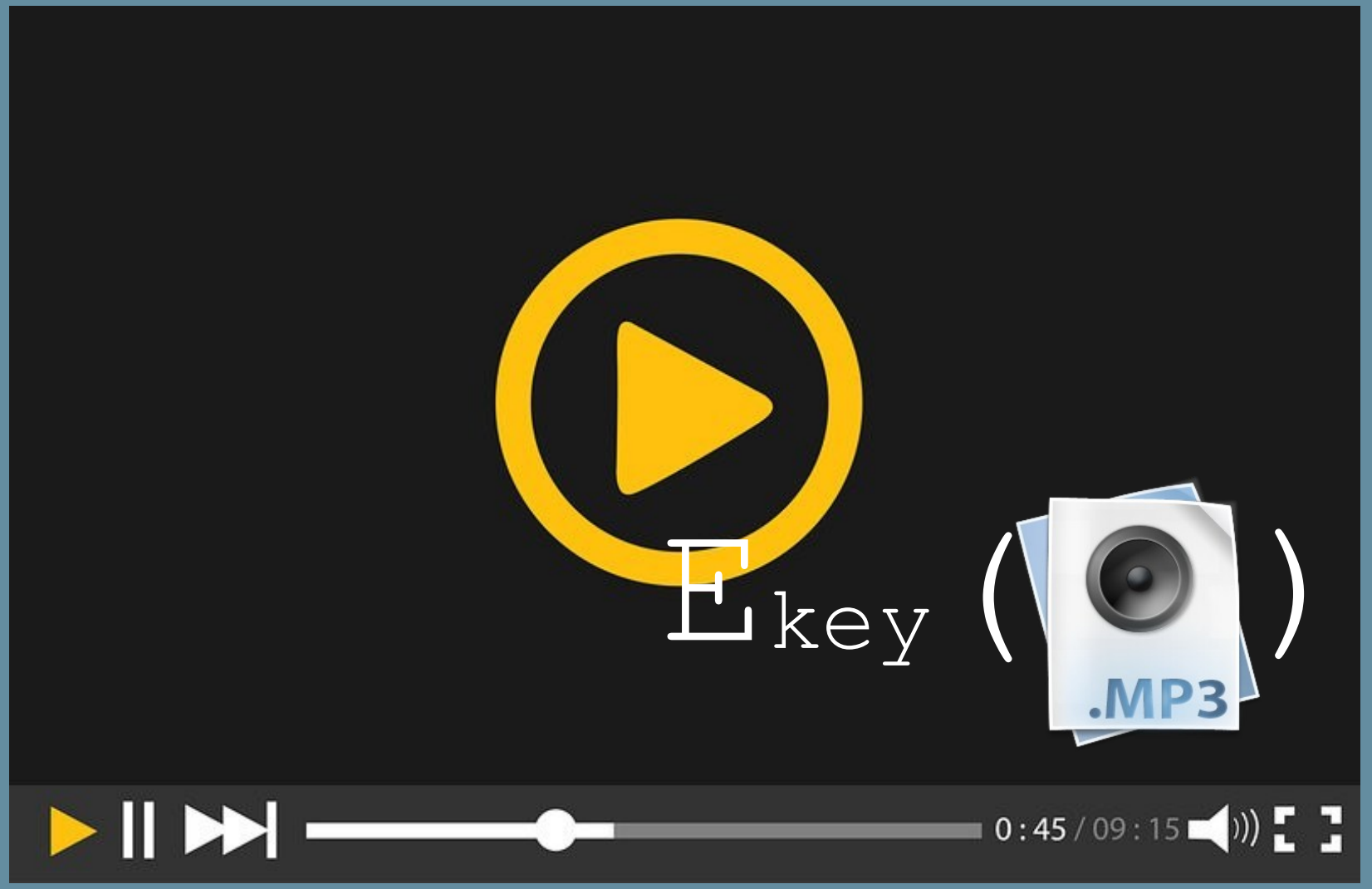
```
main() {
    if (!paid_license)
        abort()
    else
        …
}
```





Integrity


0 : 45 / 09 : 15


Ekey
.MP3
0:45 / 09:15

```
key = 0xaf75b045…

media = download("M.mp3.aes")

if (has_paid("Bob"))
    play(decrypt(key,media))
```

Ekey

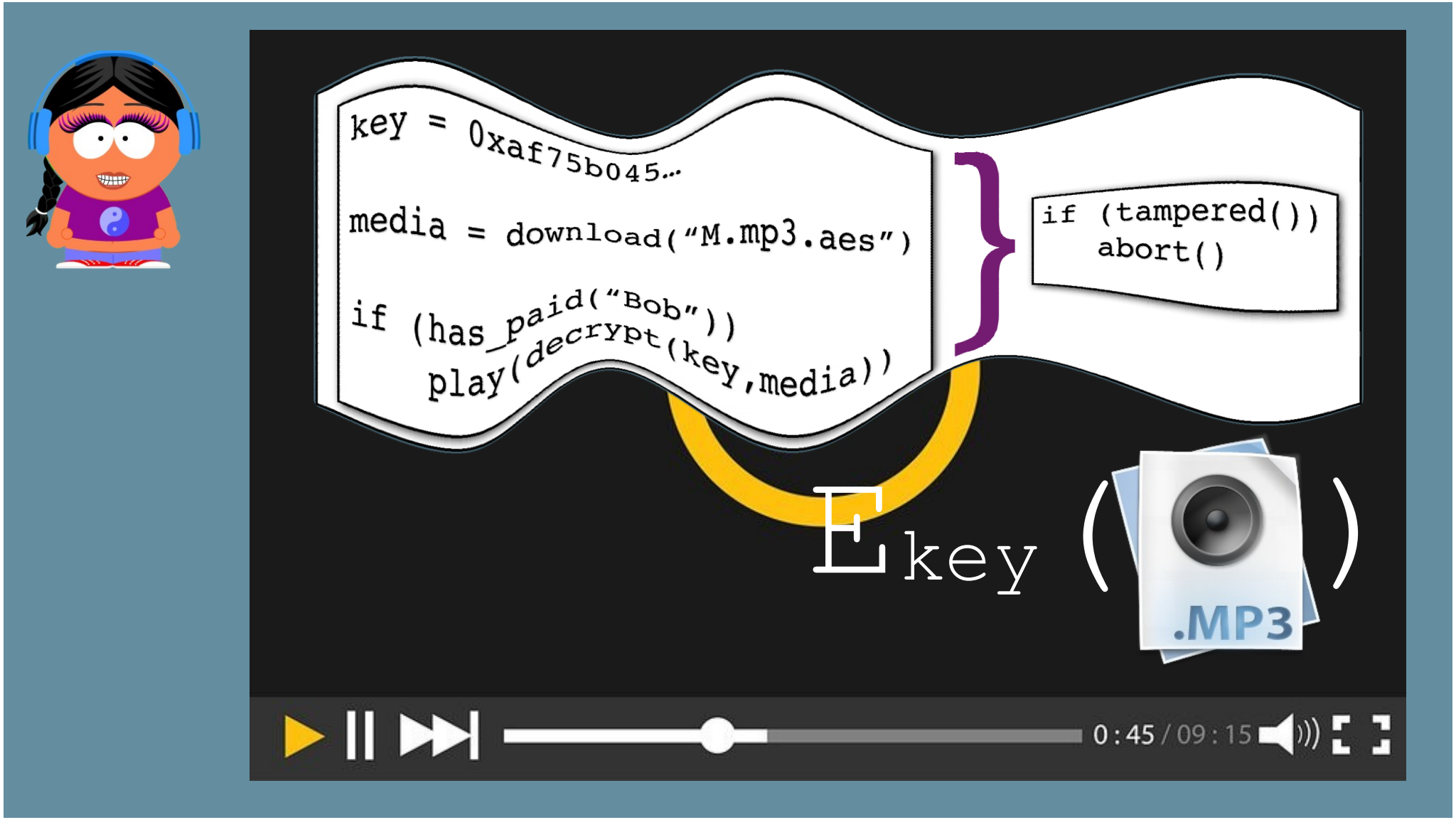
0:45 / 09:15

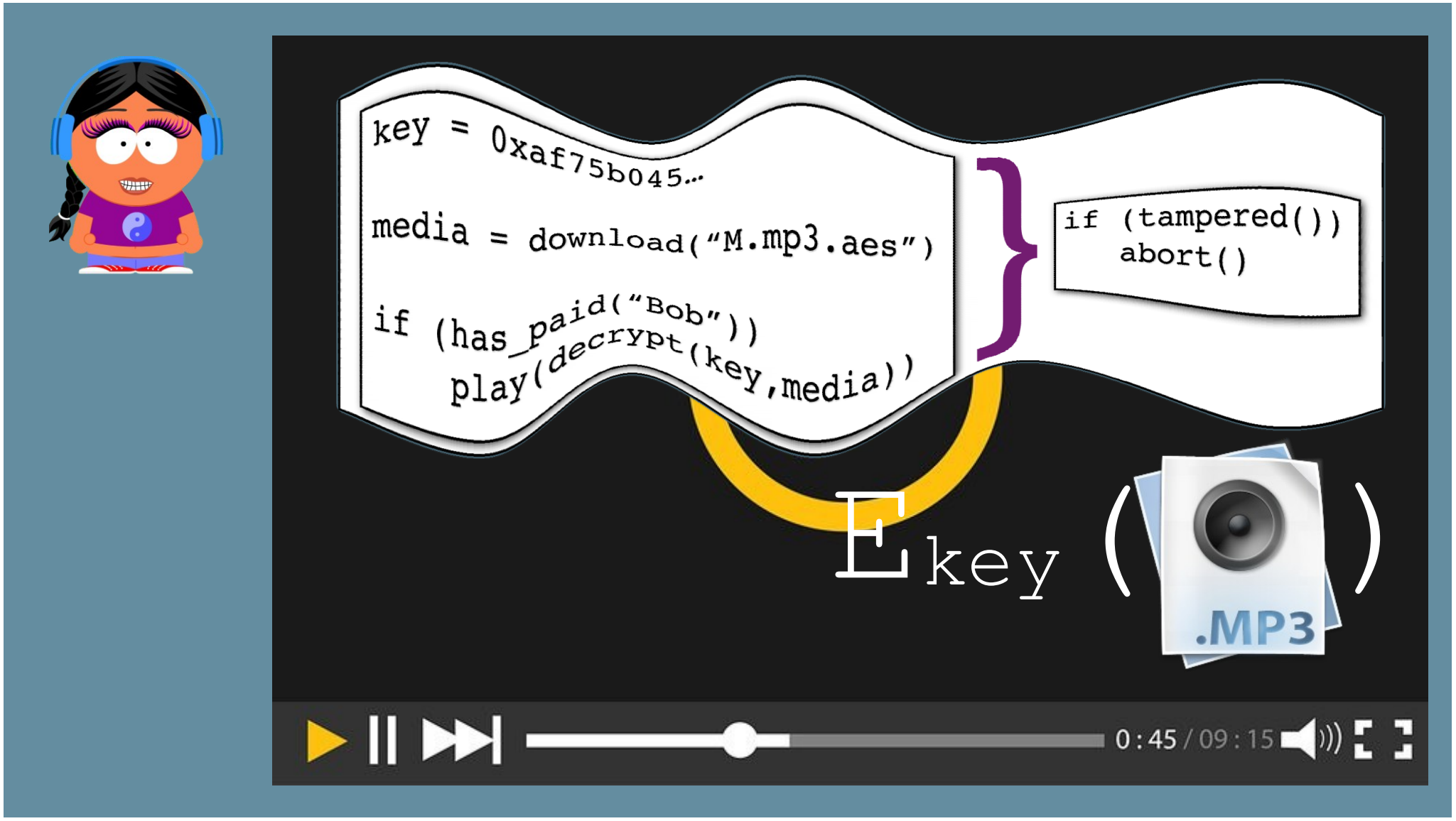
```
key = 0xaf75b045…

media = download("M.mp3.aes")

if (has_paid("Bob"))
    play(decrypt(key,media))
```

```
if (tampered())
    abort()
```

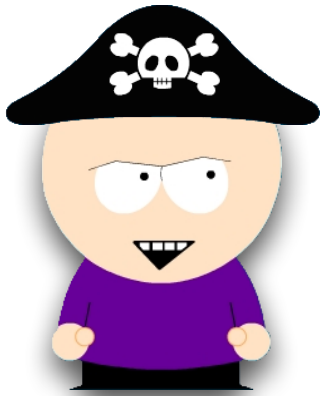
Ekey

0:45 / 09:15


key = 0xaf75b045…
media = download("M.mp3.aes")
if (has_paid("Bob"))
play(decrypt(key,media))
if (tampered())
abort()
Ekey
.MP3
0:45 / 09:15

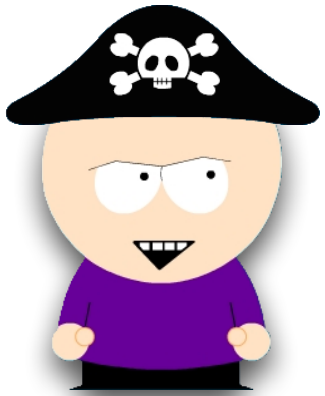

key = 0xaf75b045…
media = download("M.mp3.aes")
if (has_paid("Bob"))
play(decrypt(key,media))
if (tampered())
abort()
Ekey
.MP3
0:45 / 09:15

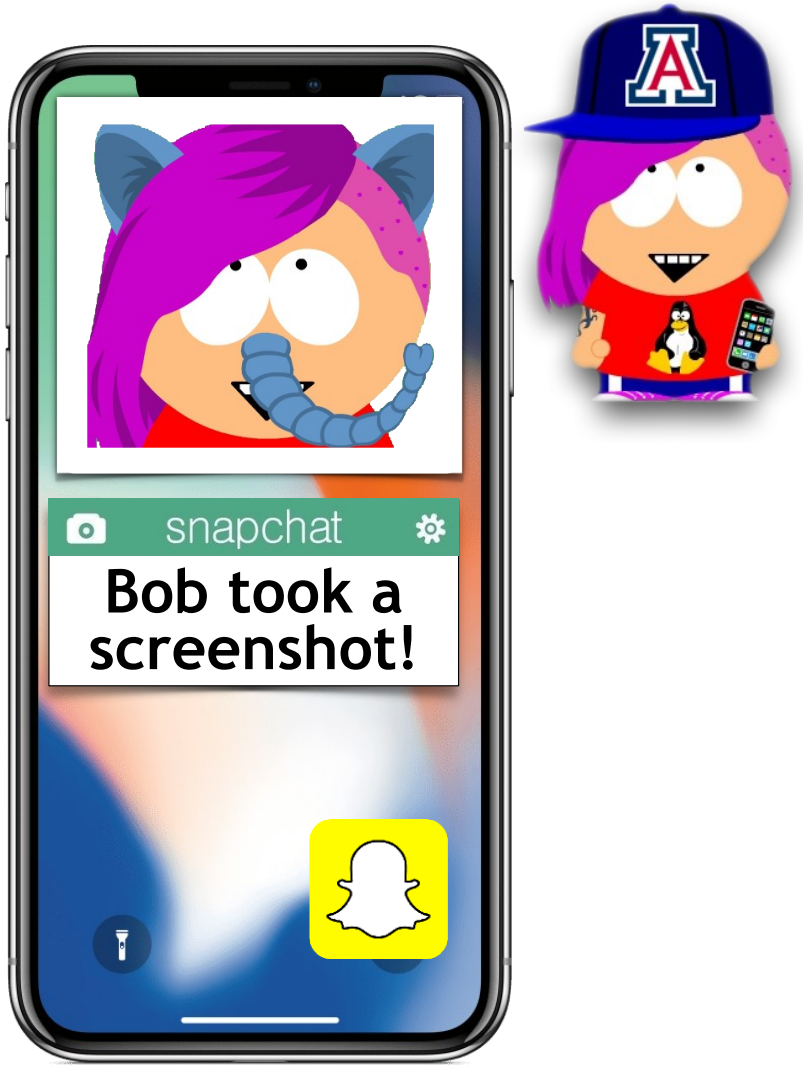

9:41
Tuesday, September 12

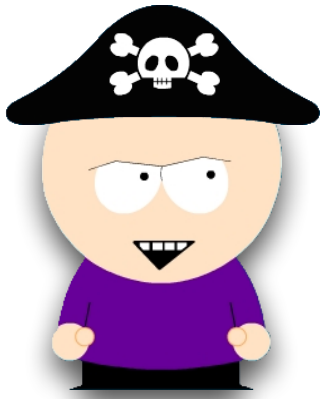


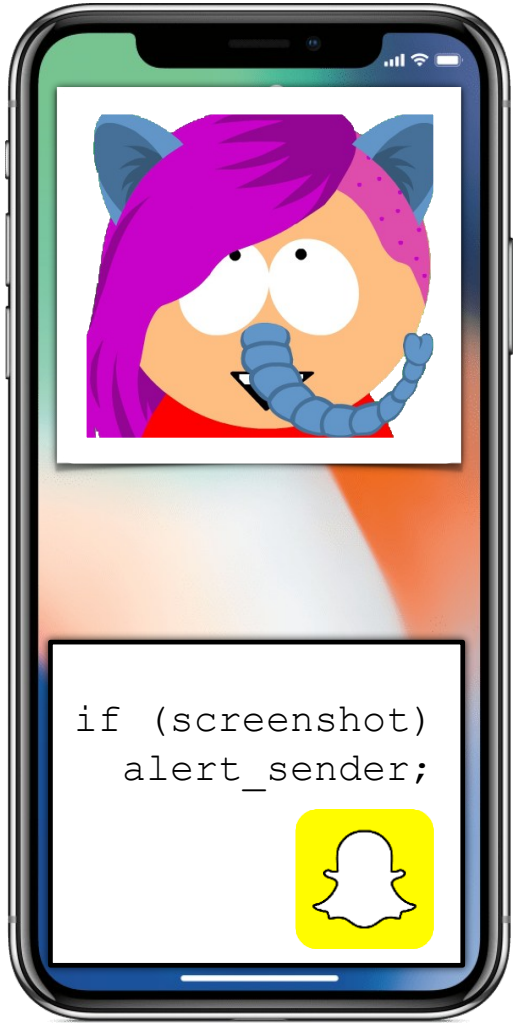

9:41
Tuesday, September 12

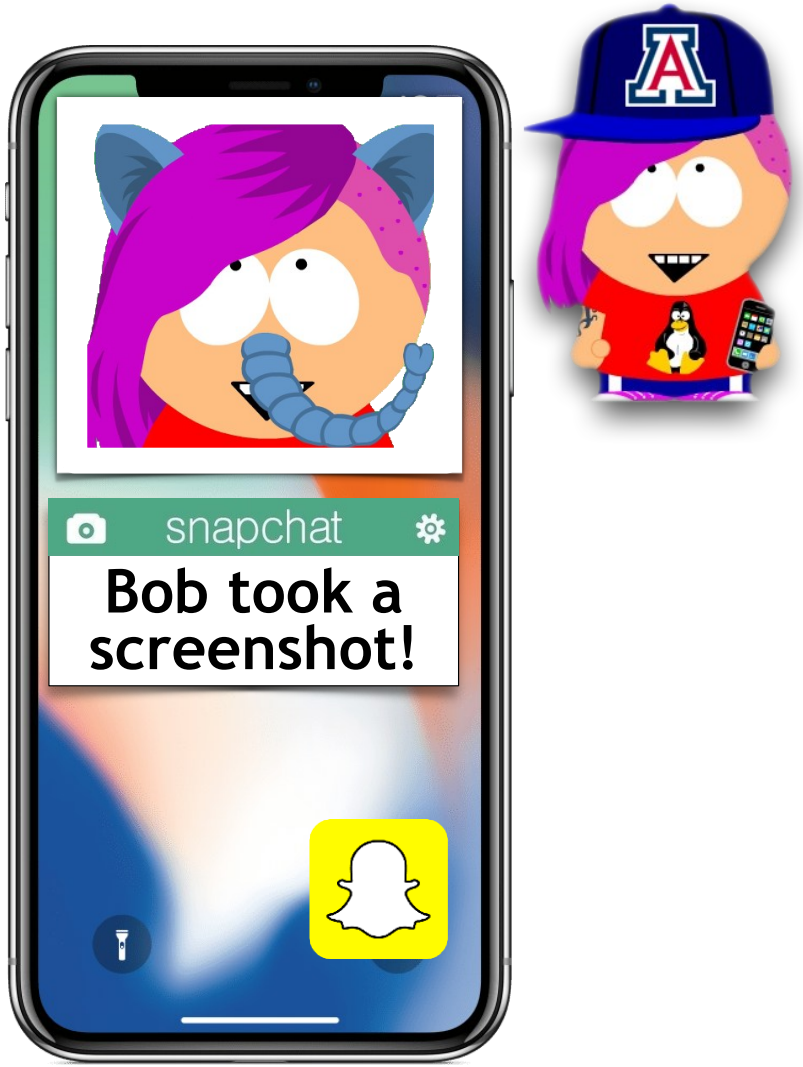

9:41
Tuesday, September 12

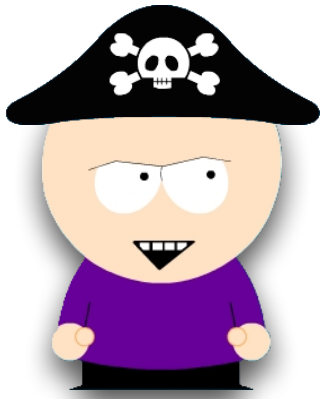



snapchat
Bob took a screenshot!

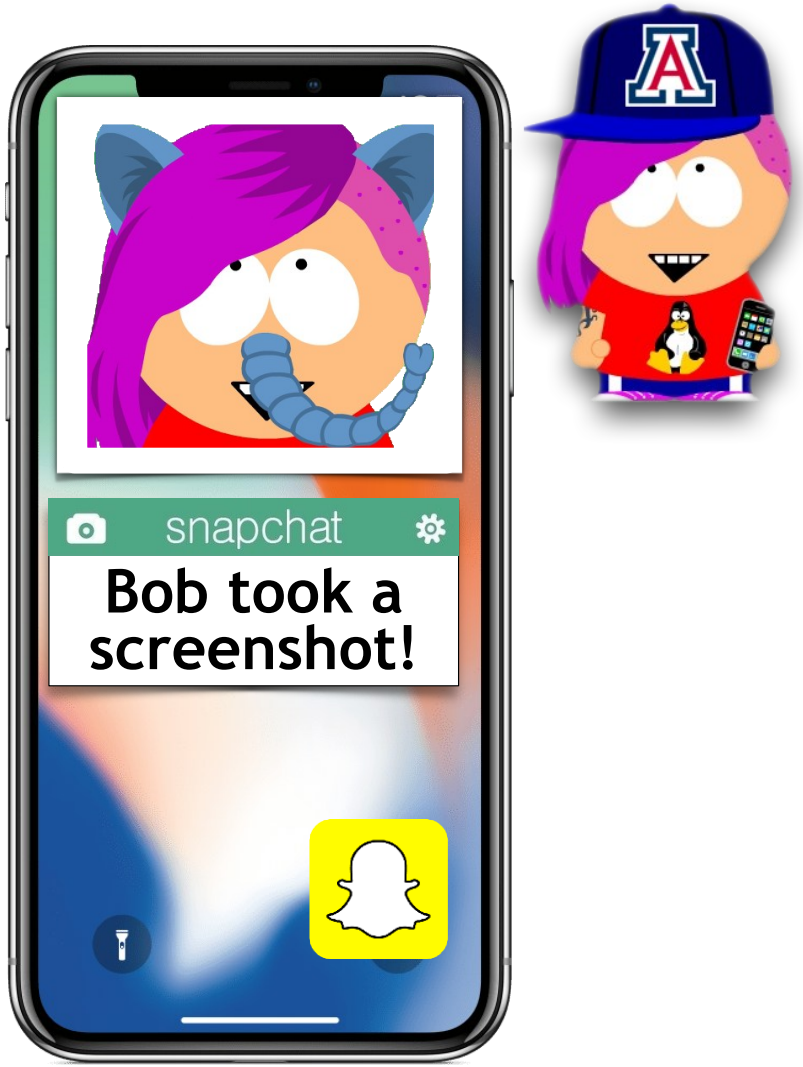


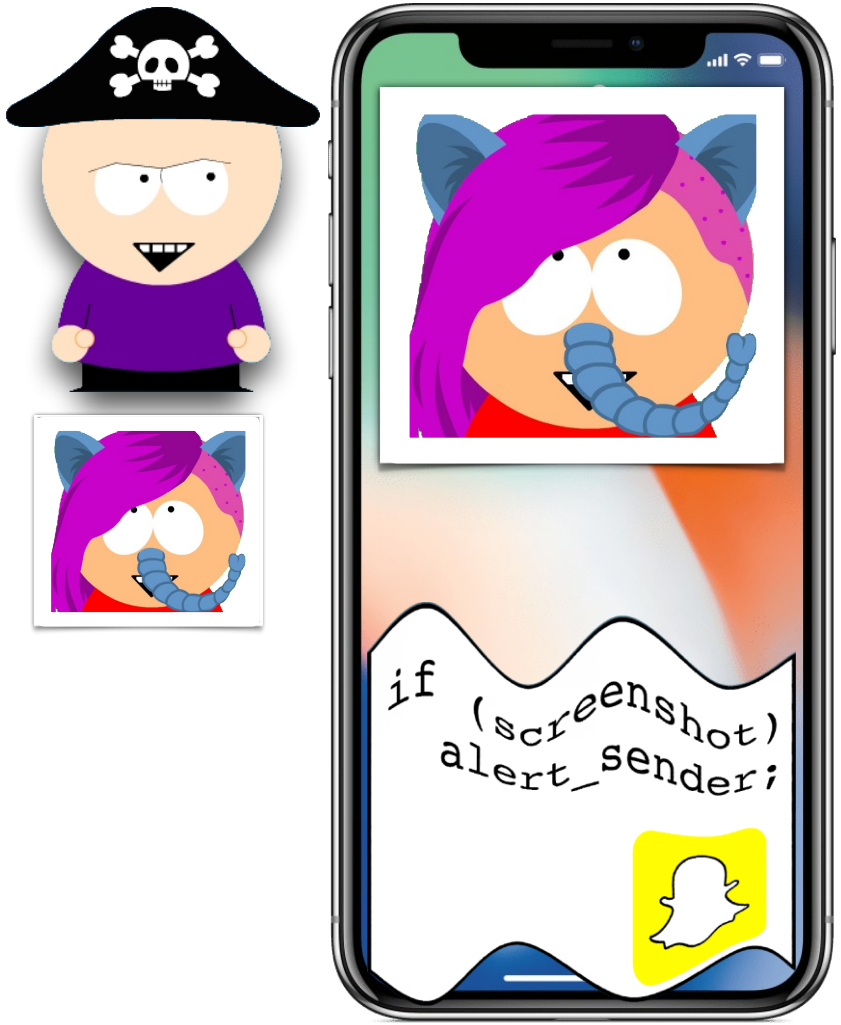

if (screenshot)
alert_sender;

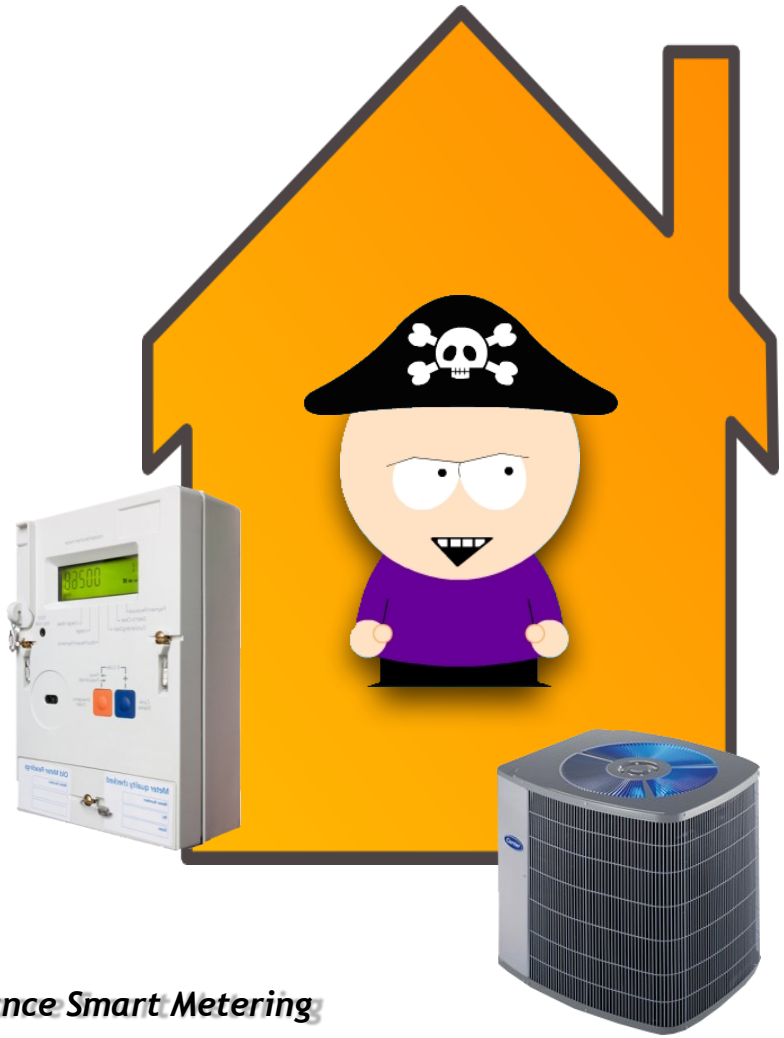

snapchat
Bob took a screenshot!



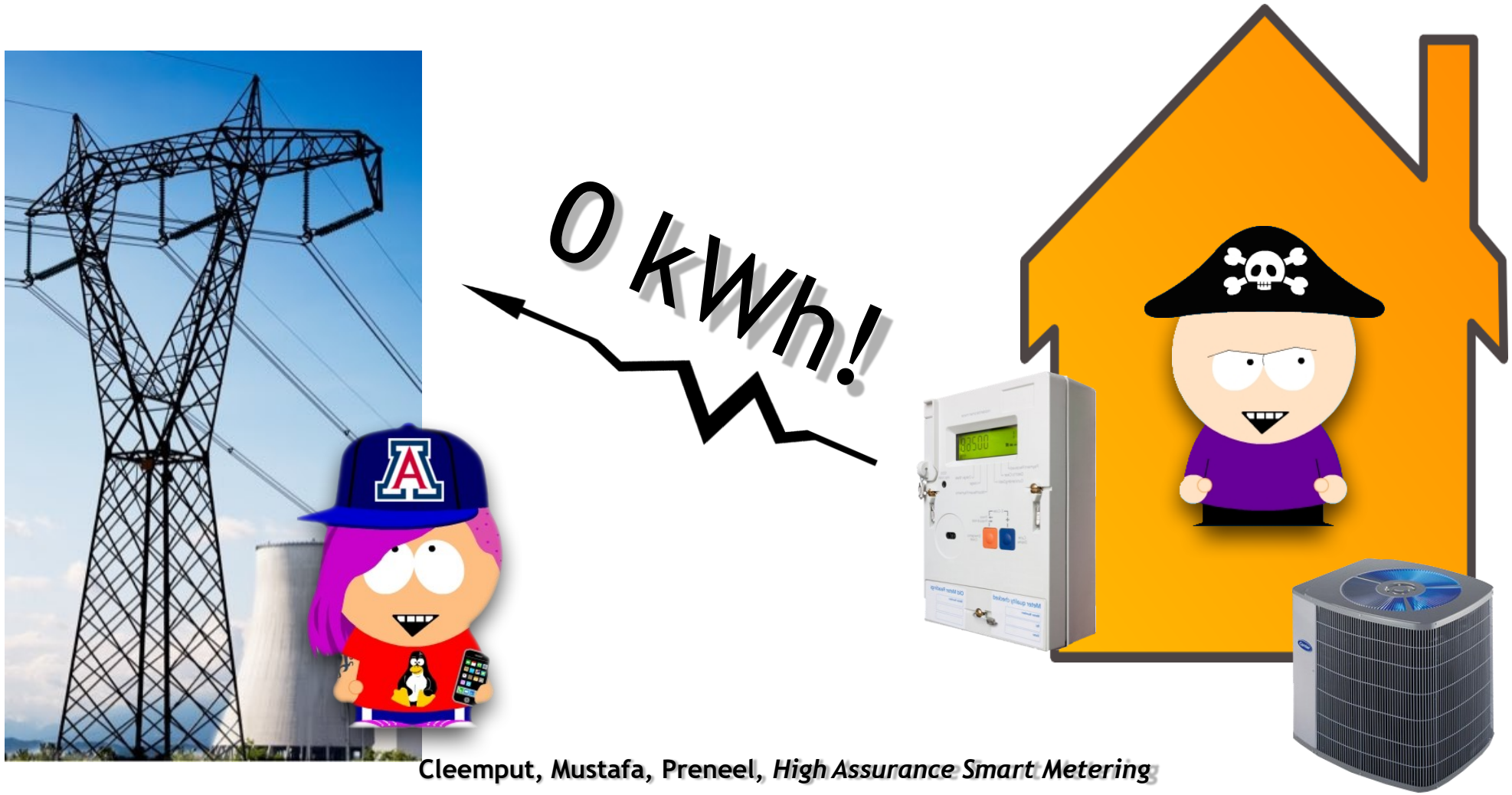

false
if (screenshot)
alert_sender;


snapchat
Bob took a screenshot!

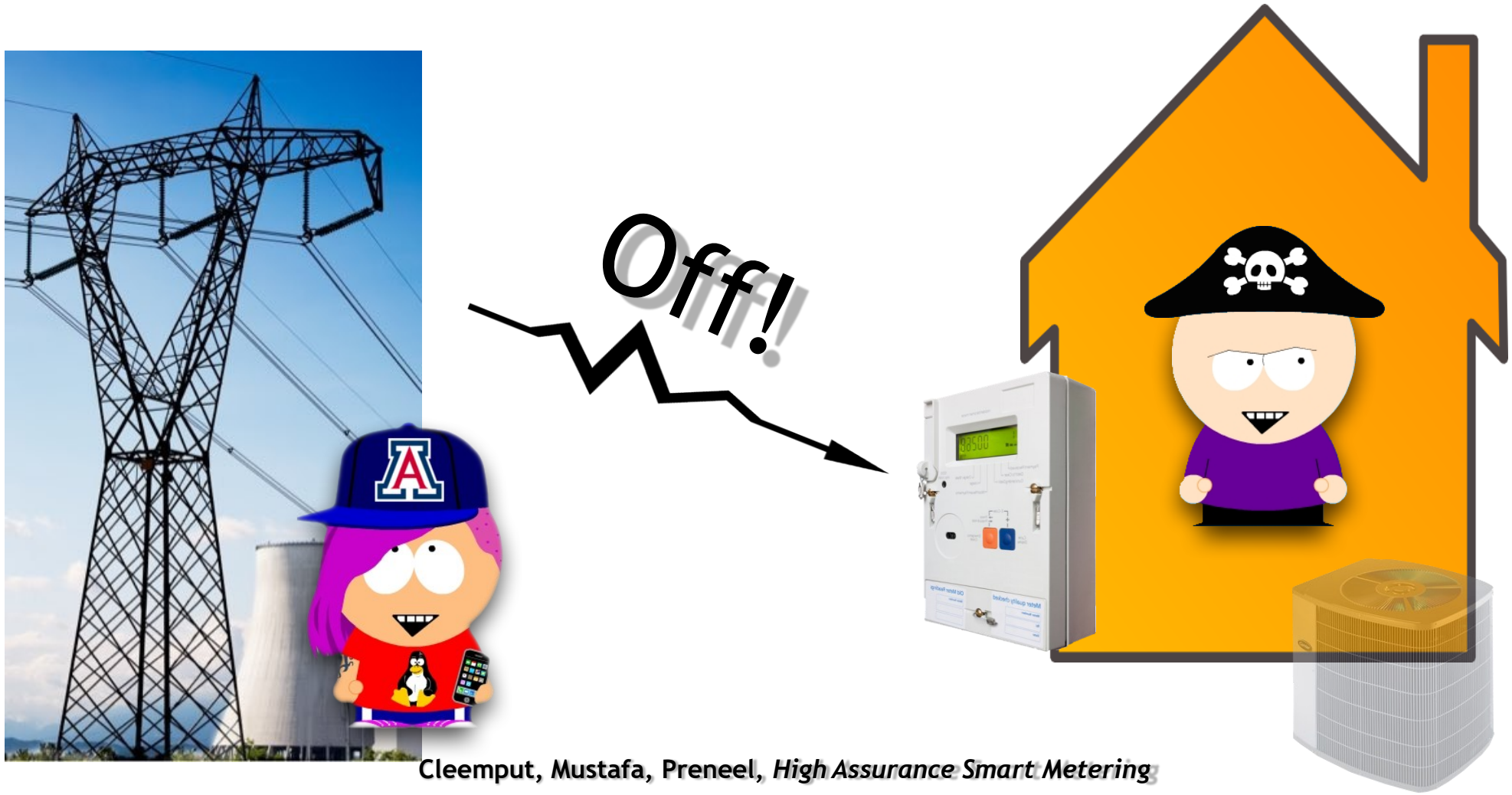

if (screenshot)
alert_sender;



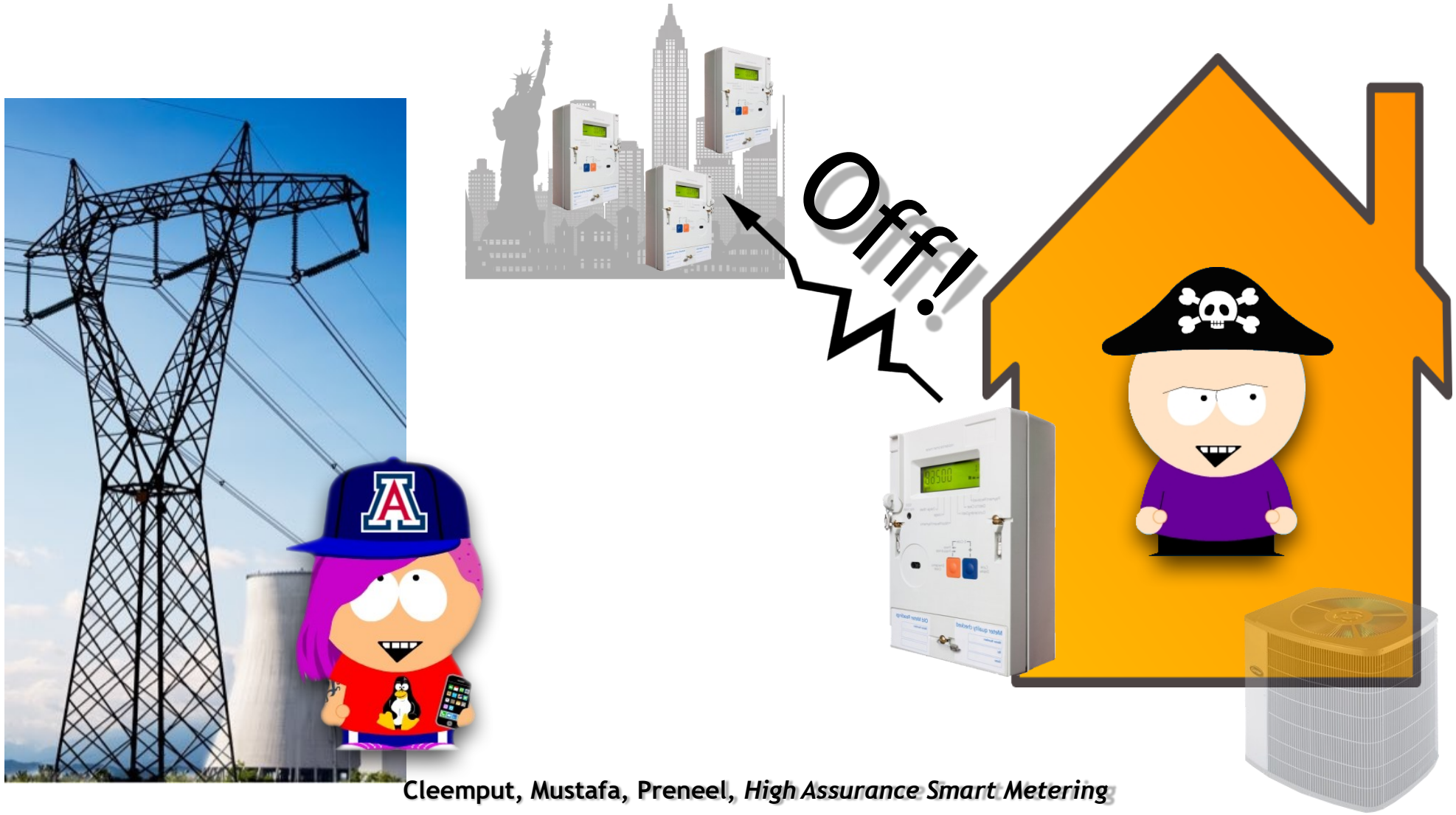
Cleemput, Mustafa, Preneel, *High Assurance Smart Metering*



Cleemput, Mustafa, Preneel, *High Assurance Smart Metering*

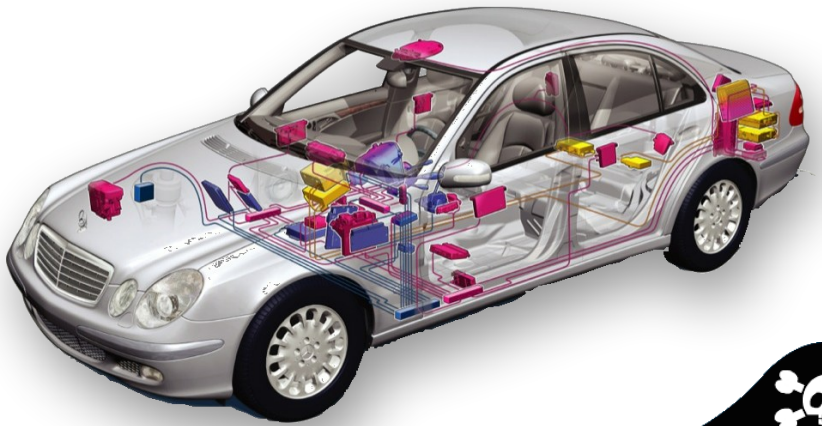


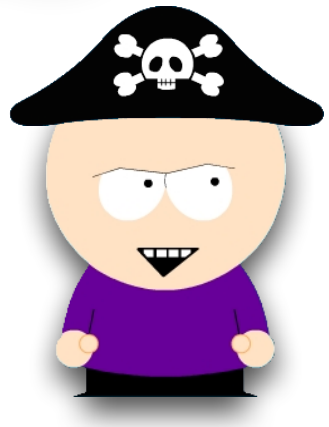
Cleemput, Mustafa, Preneel, *High Assurance Smart Metering*

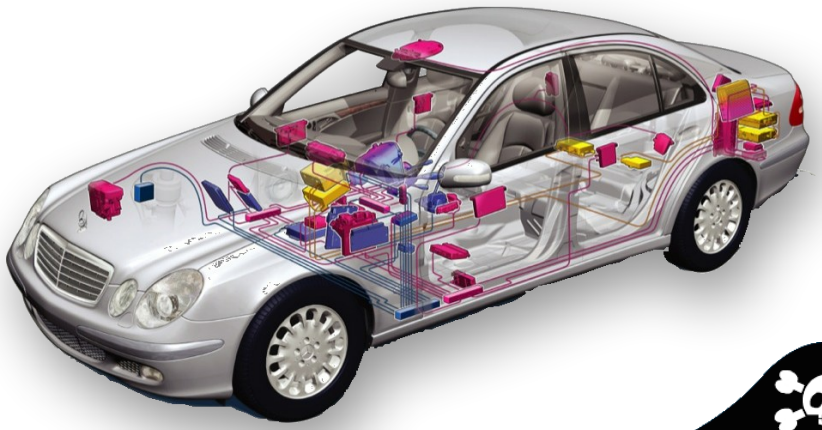


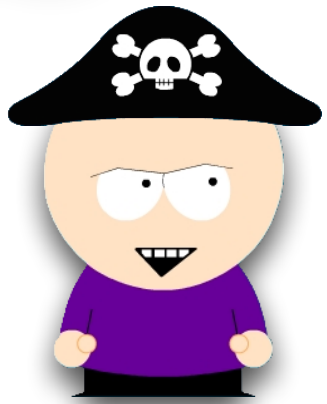
Cleemput, Mustafa, Preneel, *High Assurance Smart Metering*

Cleemput, Mustafa, Preneel, *High Assurance Smart Metering*

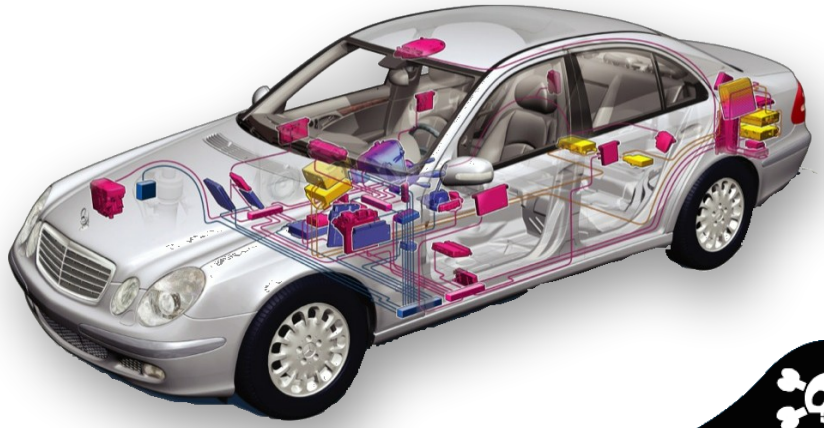


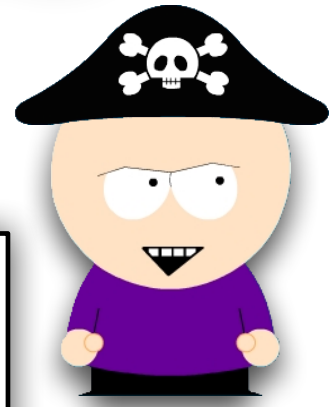




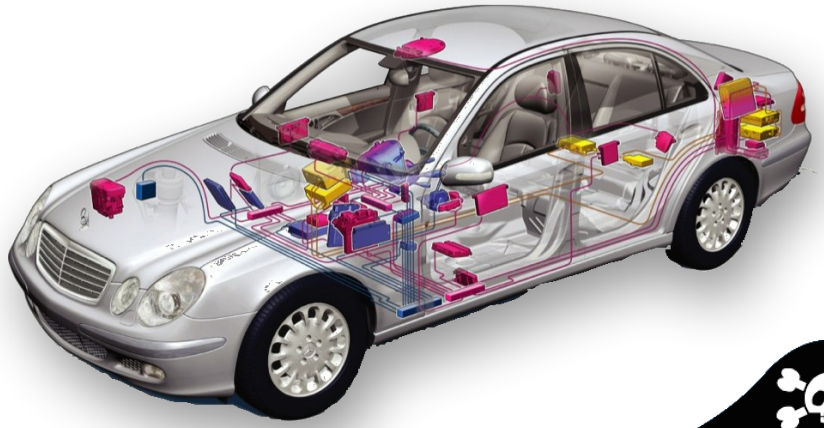


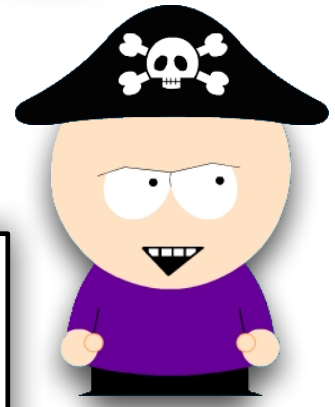

5000





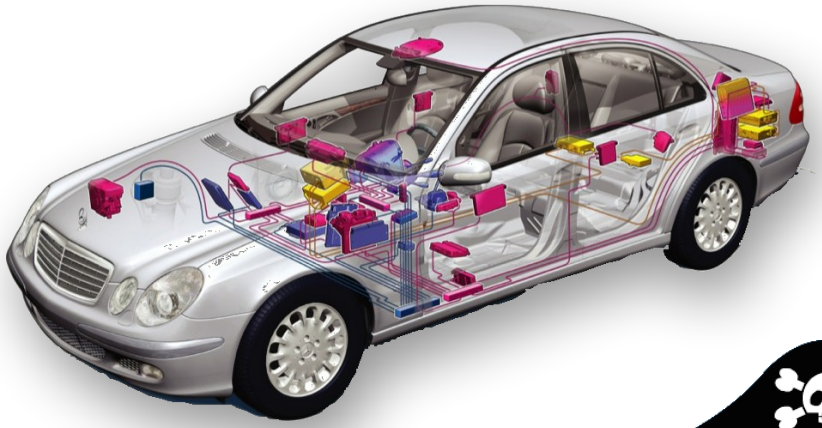
```
                true
if (has_paid_for_upgrade())
    allow_ridiculous_mode = 1;
```

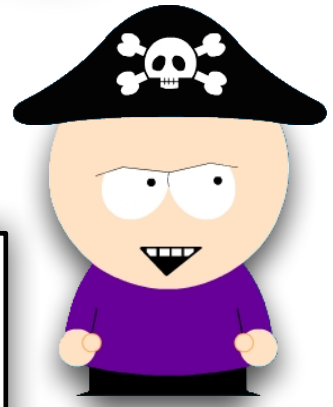




false

```
if (reported_stolen())
    send_GPS_coords();
```

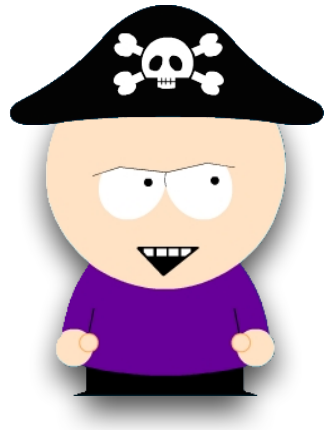


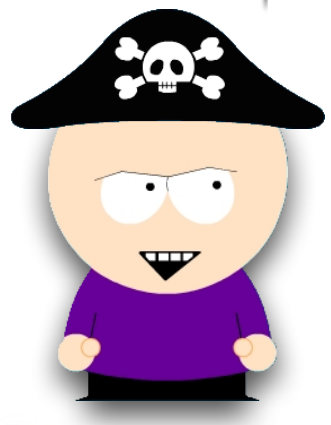


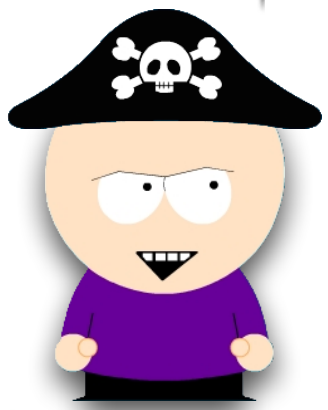
```
battery_management() {
    valuable trade secrets
}
```

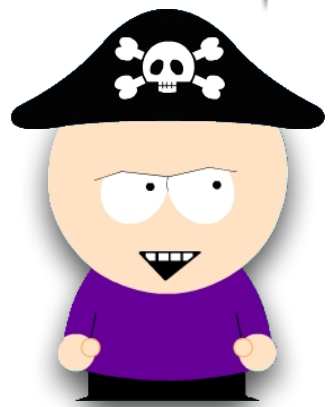
What else can Bob do?

Ca' Foscari University of Venice

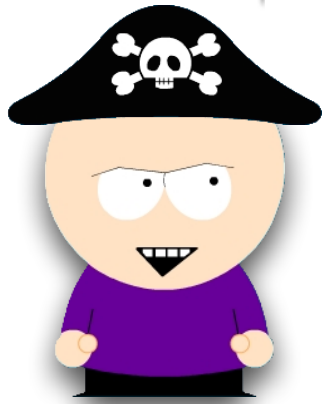








Switch off air conditioning

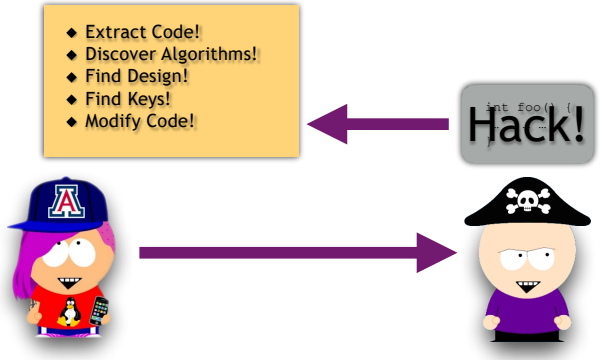


Tamper to Cause Harm

Exercise

Ca' Foscari University of Venice

- *Why do we obfuscate?*
- *Why do we tamperproof?*



Discuss with your friends!!!

Exercise

Ca' Foscari University of Venice

- *Can obfuscation be used to tamperproof a program?*

Discuss with your friends!!!

Exercise

Ca' Foscari University of Venice

- *Should you both obfuscate and tamperproof a program?*

If so, why?

Discuss with your friends!!!