Obfuscation

Omer Paneth

What does it do?

```
#include <stdio.h>
void primes(int cap) {
    int i, j, composite;
    for(i = 2; i < cap; ++i) {
      composite = 0;
      for(j = 2; j * j <= i; ++j)
        composite += !(i % j);
      if(!composite)
        printf("%d\t", i);
    }
}
int main() {
    primes(100);
}
```

How about now?

```
#include <stdio.h>
 _(__,___,____,_____){___/__<=_____?_(__,___+_____,____,_____):
 !(___%__)?_(__,___+_____,___%__,_____):___%__==___/__&&!____?(
 printf("%d\t",___/__),_(__,___+_____,____,_____)):___%__>1&&___%_
 _<___/__?_(__,_____+___,____+!(___/__%(___%__),_____)):___<__*__
 ?_(__,___+_____,____,_____):0;}main(){_(100,0,0,1);}
```

Program Obfuscation

The **art** of writing “unintelligible” code,
while preserving functionality.

Obfuscation is Fun

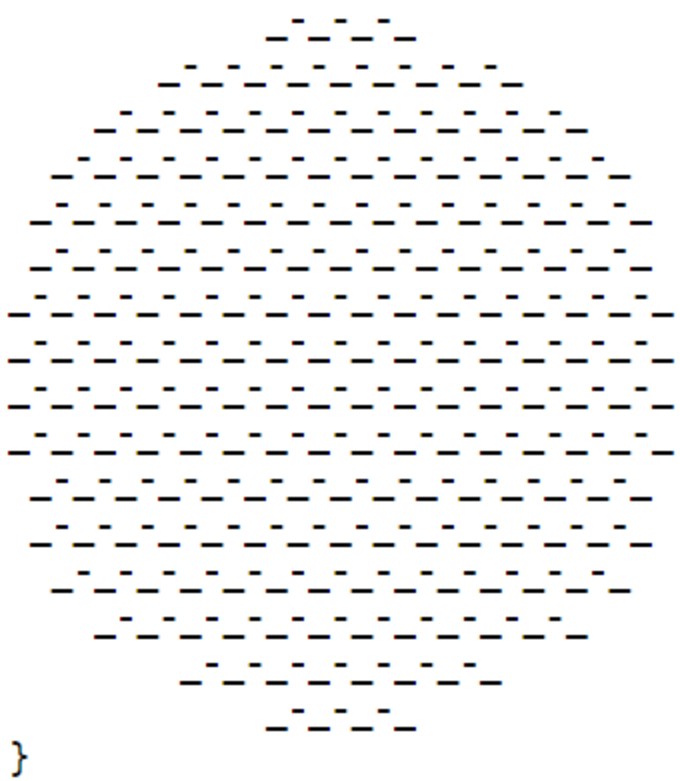
International Obfuscated C Code Contest (IOCCC)

First IOCCC 1984

```
int i;main(){for(;i["]<i;++i){--i;}"];read('-'-'-',i+++"hell\
o, world!\n",'/'/'/'));}read(j,i,p){write(j/p+p,i--j,i/i);}
```

IOCCC 88

```
#define _ -F<00||--F-OO--;
int F=00,OO=00;main(){F_OO();printf("%1.3f\n",4.*-F/OO/OO);}F_OO()
{
```



```
}
```

IOCCC 98

```
#include                                  <math.h>
#include                                 <sys/time.h>
#include                                 <X11/Xlib.h>
#include                                <X11/keysym.h>
                                        double L ,o ,P
                                       ,_=dt,T,Z,D=1,d,
                                       s[999],E,h= 8,I,
                                       J,K,w[999],M,m,O
                                      ,n[999],j=33e-3,i=
                                      1E3,r,t, u,v ,W,S=
                                      74.5,l=221,X=7.26,
                                      a,B,A=32.2,c, F,H;
                                      int N,q, C, y,p,U;
                                     Window z; char f[52]
                                   ; GC k; main(){ Display*e=
 XOpenDisplay( 0); z=RootWindow(e,0); for (XSetForeground(e,k=XCreateGC (e,z,0,0),BlackPixel(e,0))
; scanf("%lf%lf%lf",y +n,w+y, y+s)+1; y ++); XSelectInput(e,z= XCreateSimpleWindow(e,z,0,0,400,400,
0,0,WhitePixel(e,0) ),KeyPressMask); for(XMapWindow(e,z); ; T=sin(O)){ struct timeval G={ 0,dt*1e6}
; K= cos(j); N=1e4; M+= H*_; Z=D*K; F+=_*P; r=E*K; W=cos( O); m=K*W; H=K*T; O+=D*_*F/ K+d/K*E*_; B=
sin(j); a=B*T*D-E*W; XClearWindow(e,z); t=T*E+ D*B*W; j+=d*_*D-_*F*E; P=W*E*B-T*D; for (o+=(I=D*W+E
*T*B,E*d/K *B+v+B/K*F*D)*_; p<y; ){ T=p[s]+i; E=c-p[w]; D=n[p]-L; K=D*m-B*T-H*E; if(p [n]+w[ p]+p[s
]== 0|K <fabs(W=T*r-I*E +D*P) |fabs(D=t *D+Z *T-a *E)> K)N=1e4; else{ q=W/K *4E2+2e2; C= 2E2+4e2/ K
 *D; N-1E4&& XDrawLine(e ,z,k,N ,U,q,C); N=q; U=C; } ++p; } L+=_* (X*t +P*M+m*l); T=X*X+ l*l+M *M;
  XDrawString(e,z,k ,20,380,f,17); D=v/l*15; i+=(B *l-M*r -X*Z)*_; for(; XPending(e); u *=CS!=N){
                                    XEvent z; XNextEvent(e ,&z);
                                        ++*((N=XLookupKeysym
                                          (&z.xkey,0))-IT?
                                          N-LT? UP-N?& E:&
                                          J:& u: &h); --*(
                                          DN -N? N-DT ?N==
                                          RT?&u: & W:&h:&J
                                           ); } m=15*F/l;
                                           c+=(I=M/ l,l*H
                                           +I*M+a*X)*_; H
                                           =A*r+v*X-F*l+(
                                           E=.1+X*4.9/l,t
                                           =T*m/32-I*T/24
                                            )/S; K=F*M+(
                                            h* 1e4/l-(T+
                                            E*5*T*E)/3e2
                                            )/S-X*d-B*A;
                                            a=2.63 /l*d;
                                            X+=( d*l-T/S
                                             *(.19*E +a
                                             *.64+J/1e3
                                             )-M* v +A*
                                             Z)*_; l +=
                                             K *_; W=d;
                                             sprintf(f,
                                             "%5d  %3d"
                                             "%7d",p =l
                                            /1.7,(C=9E3+
                                  O*57.3)%0550,(int)i); d+=T*(.45-14/l*
                                 X-a*130-J* .14)*_/125e2+F*_*v; P=(T*(47
                                 *I-m* 52+E*94 *D-t*.38+u*.21*E) /1e2+W*
                                 179*v)/2312; select(p=0,0,0,0,&G); v-=(
                                  W*F-T*(.63*m-I*.086+m*E*19-D*25-.11*u
                                   )/107e2)*_; D=cos(o); E=sin(o); } }
```

IOCCC 04

```
#define G(n) int n(int t, int q, int d) #define X(p,t,s) (p>=t&&p<(t+s)&&(p-
(t)&1023)<(s&1023)) #define U(m) *((signed char *)(m)) #define F if(!--q){ #define I(s)
(int)main-(int)s #define P(s,c,k) for(h=0; h>>14==0;
h+=129)Y(16*c+h/1024+Y(V+36))&128>>(h&7)?U(s+(h&15367))=k:k G (B) { Z; F D = E (Y (V),
C = E (Y (V), Y (t + 4) + 3, 4, 0), 2, 0); Y (t + 12) = Y (t + 20) = i; Y (t + 24) = 1; Y (t + 28) = t; Y (t
+ 16) = 442890; Y (t + 28) = d = E (Y (V), s = D * 8 + 1664, 1, 0); for (p = 0; j < s; j++, p++) U (d
+ j) = i == D | j < p ? p--, 0 : (n = U (C + 512 + i++)) < ' ' ? p |= n * 56 - 497, 0 : n; } n = Y (Y (t +
4)) & 1; F U (Y (t + 28) + 1536) |= 62 & -n; M U (d + D) = X (D, Y (t + 12) + 26628, 412162) ? X
(D, Y (t + 12) + 27653, 410112) ? 31 : 0 : U (d + D); for (; j < 12800; j += 8) P (d + 27653 + Y (t
+ 12) + ' ' * (j & ~511) + j % 512, U (Y (t + 28) + j / 8 + 64 * Y (t + 20)), 0); } F if (n) { D = Y (t +
28); if (d - 10) U (++Y (t + 24) + D + 1535) = d; else { for (i = D; i < D + 1600; i++) U (i) = U (i +
64); Y (t + 24) = 1; E (Y (V), i - 127, 3, 0); } } else Y (t + 20) += ((d >> 4) ^ (d >> 5)) - 3; } } G (_);
G (o); G (main) { Z, k = K; if (!t) { Y (V) = V + 208 - (I (_)); L (209, 223) L (168, 0) L (212, 244)
_((int) &s, 3, 0); for (; 1;) R n = Y (V - 12); if (C & ' ') { k++; k %= 3; if (k < 2) { Y (j) -= p; Y (j) +=
p += U (&D) * (1 - k * 1025); if (k) goto y; } else { for (C = V - 20; !i && D & 1 && n && (X (p, Y
(n + 12), Y (n + 16)) ? j = n + 12, Y (C + 8) = Y (n + 8), Y (n + 8) = Y (V - 12), Y (V - 12) = n, 0 : n);
C = n, n = Y (n + 8)); i = D & 1; j &= -i; } } else if (128 & ~D) { E (Y (n), n, 3, U (V + D % 64 + 131)
^ 32); n = Y (V - 12); y:C = 1 << 24; M U (C + D) = 125; o (n, 0, C); P (C + p - 8196, 88, 0); M U
(Y (0x11028) + D) = U (C + D); } } } for (D = 720; D > -3888; D--) putchar (D > 0 ? "
)!\320\234\360\256\370\256 0\230F .,mnbvcxz ;lkjhgfdsa \n][poiuytrewq =-0987654321
\357\262 \337\337 \357\272 \337\337 ( )\"\343\312F\320!/ !\230 26!/\16 K>!/\16\332
\4\16\251\0160\355&\2271\20\2300\355`x{0\355\347\2560 \237qpa%\231o!\230
\337\337\337 , )\"K\240 \343\316qrpxzy\0 sRDh\16\313\212u\343\314qrzy !0( " [D] ^ 32 :
Y (I (D))); return 0; } G (o) { Z; if (t) { C = Y (t + 12); j = Y (t + 16); o (Y (t + 8), 0, d); M U (d + D)
= X (D, C, j) ? X (D, C + 1025, j - 2050) ? X (D, C + 2050, j - 3075) ? X (D, C + 2050, j - 4100) ? X
(D, C + 4100, ((j & 1023) + 18424)) ? 176 : 24 : 20 : 28 : 0 : U (d + D); for (n = Y (t + 4); U (i +
n); i++) P (d + Y (t + 12) + 5126 + i * 8, U (n + i), 31); E (Y (t), t, 2, d); } } G (_) { Z = Y (V + 24); F
Y (V - 16) += t; D = Y (V - 16) - t; } F for (i = 124; i < 135; i++) D = D << 3 | Y (t + i) & 7; } if (q >
0) { for (; n = U (D + i); i++) if (n - U (t + i)) { D += _(D, 2, 0) + 1023 & ~511; i = ~0; } F if (Y (D)) {
n = _(164, 1, 0); Y (n + 8) = Y (V - 12); Y (V - 12) = n; Y (n + 4) = i = n + 64; for (; j < 96; j++) Y (i
+ j) = Y (t + j); i = D + 512; j = i + Y (i + 32); for (; Y (j + 12) != Y (i + 24); j += 40); E (Y (n) = Y (j +
16) + i, n, 1, 0); } } } return D; }
```

The author, Gavin Barraclough: “This is a 32-bit multitasking operating system for x86 computers, with GUI and filesystem, support for loading and executing user applications in elf binary format, with ps2 mouse and keyboard drivers, and vesa graphics. And a command shell. And an application - a simple text-file viewer.”

What is it good for?

- Protecting “sensitive” code (e.g. login).
- Protecting proprietary algorithms.
- “Authenticated” client applications.
- Digital Rights Management (DRM)
- Polymorphic code.

Common Techniques

- Obfuscating source code:
  - Variable renaming
  - Changing the program's flow
  - Higher level semantic changes
- Obfuscating object code:
  - Adding and reordering operations
  - Encryption of unused modules
  - Anti-debugging

The Obfuscation Race

- Common methods based on “security by obscurity”.
- Doomed to be broken eventually.
- Automatic obfuscation tools.
- Automatic “de-obfuscators”

“[Secure obfuscation] is unlikely. The computer ultimately has to decipher and follow a software program’s true instructions. Each new obfuscation technique has to abide by this requirement and, thus, will be reverse engineered.”

- Chris Wysopal

Good Obfuscation, Bad Code

Can we have “unbreakable obfuscation”?

Obfuscation meets Modern Crypto:

- Definitions [Hada 00, Barak et. al. 01]
- Constructions [Garg et. al. 13]
- Applications [Diffie–Hellman 76,…]

Definition:

[Barak et. al 01]

An PPT algorithm O is an obfuscator for a class of programs Π if:

- Functionality:

  For every $P \in \Pi$, $O(P) \to \tilde{P}$

  such that P and $\tilde{P}$ have the same functionality.

- Security (Virtual-black-box)

  For every $P \in \Pi$, $O(P)$ gives no more information than a **black-box** that computes P.

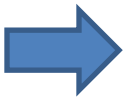
Obfuscating “Learnable” Functions

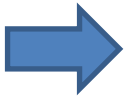
Construct an obfuscator for the following family of programs:

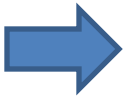
$$\Pi = \{P_c\}_{c \in \mathbb{N}}, \qquad P_c(x) = c \cdot x$$

The obfuscator: $O(P_c) \rightarrow P_c$

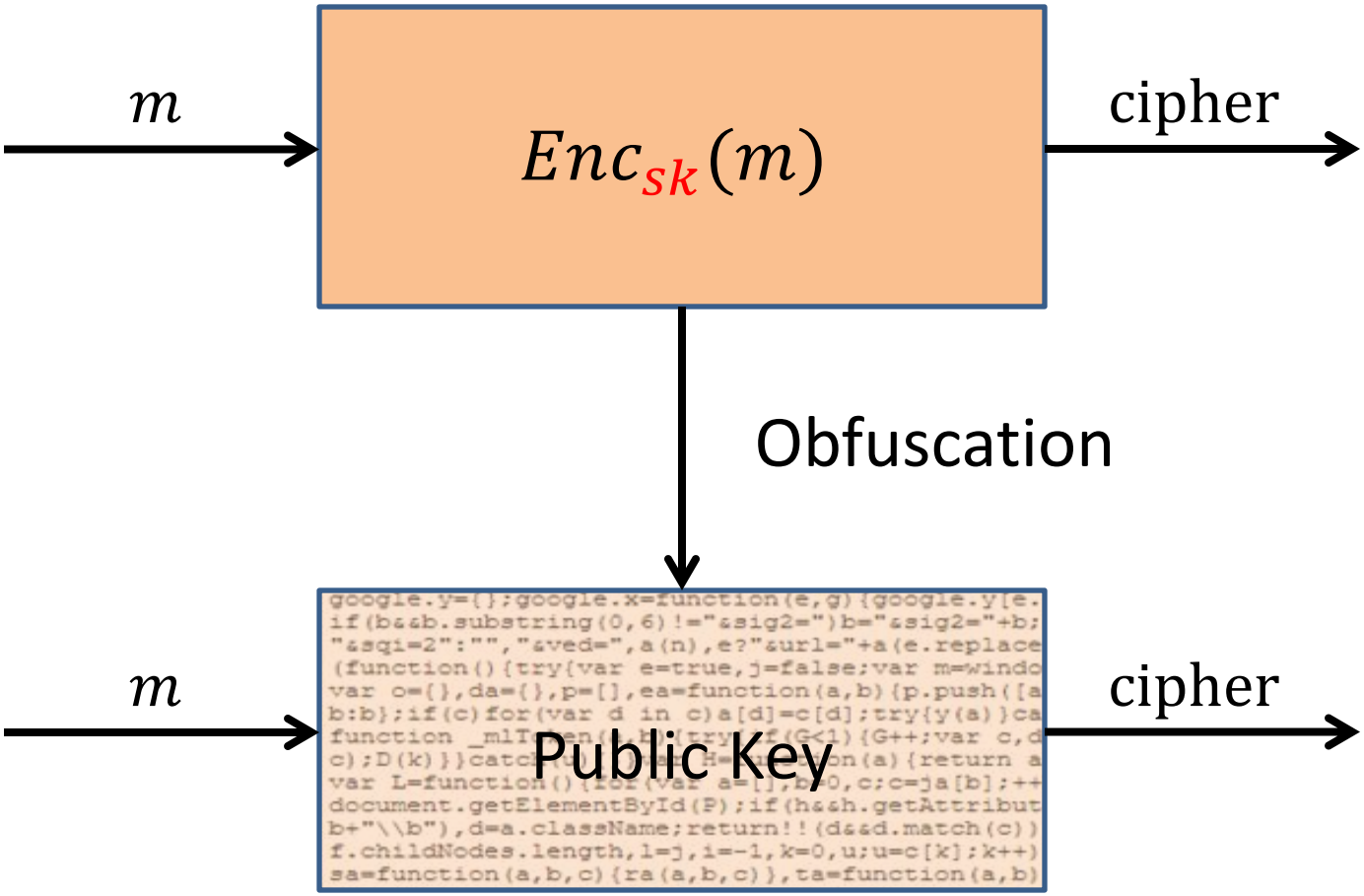
Crypto Applications

Private-key encryption  Public-key encryption

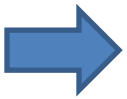
Message authentication codes (MACs)  Digital Signatures

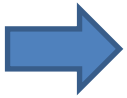
Pseudo-random functions  “Random oracles”

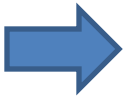
Private-Key to Public-Key



Crypto Applications

Private-key encryption  Public-key encryption

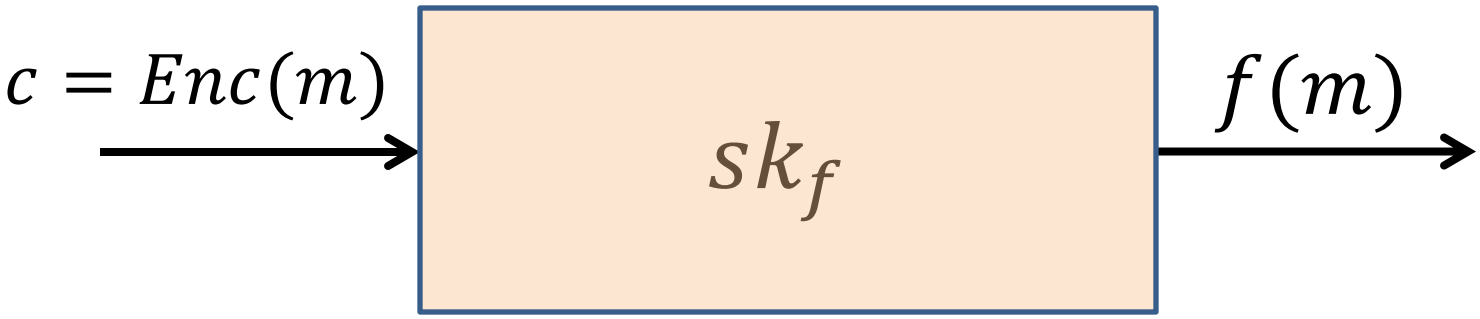
Message authentication codes (MACs)  Digital Signatures

Pseudo-random functions  “Random oracles”

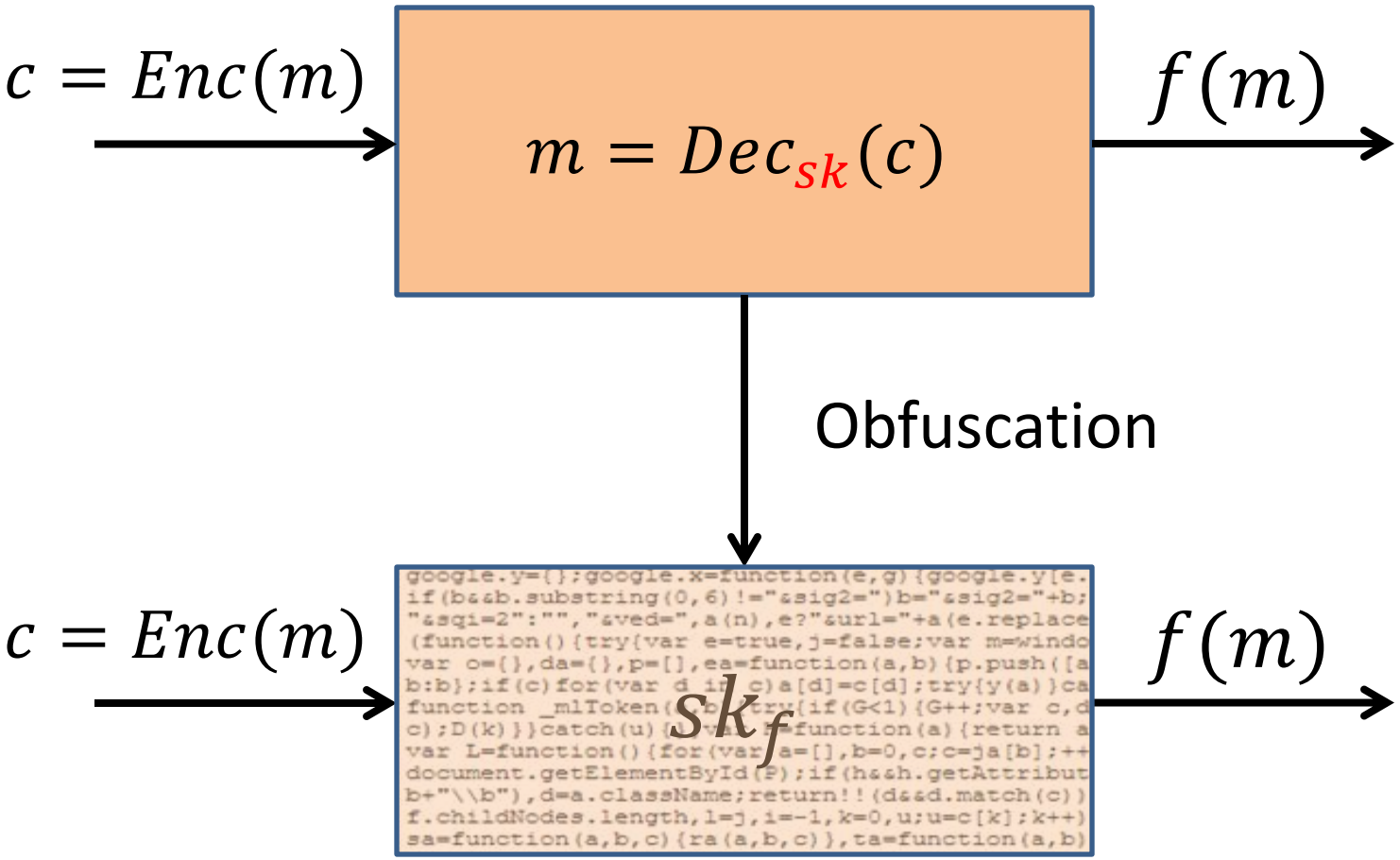
Functional Encryption

Give out data and control how it is being used

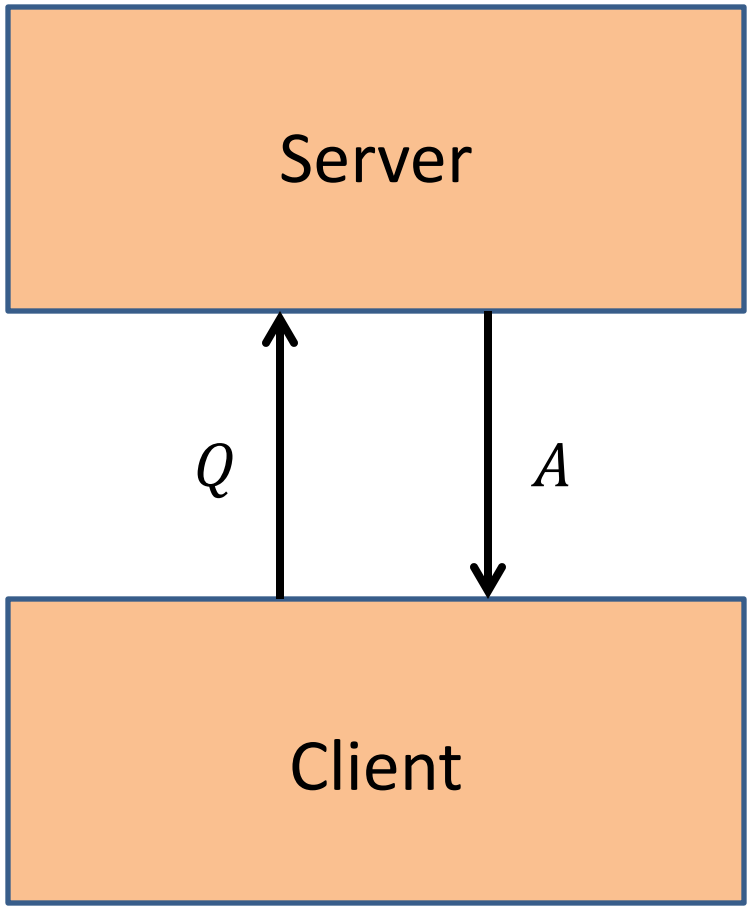
- DRM \ Watermarking
- Privacy
- Delegating storage\computation



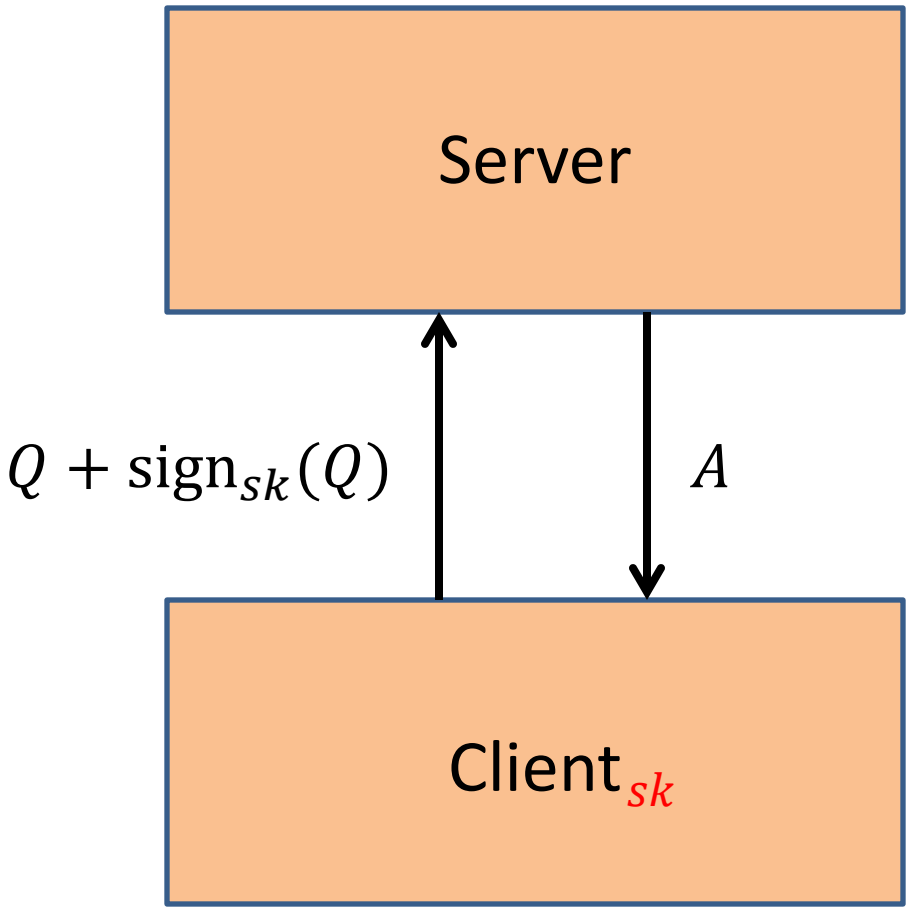
Functional Encryption



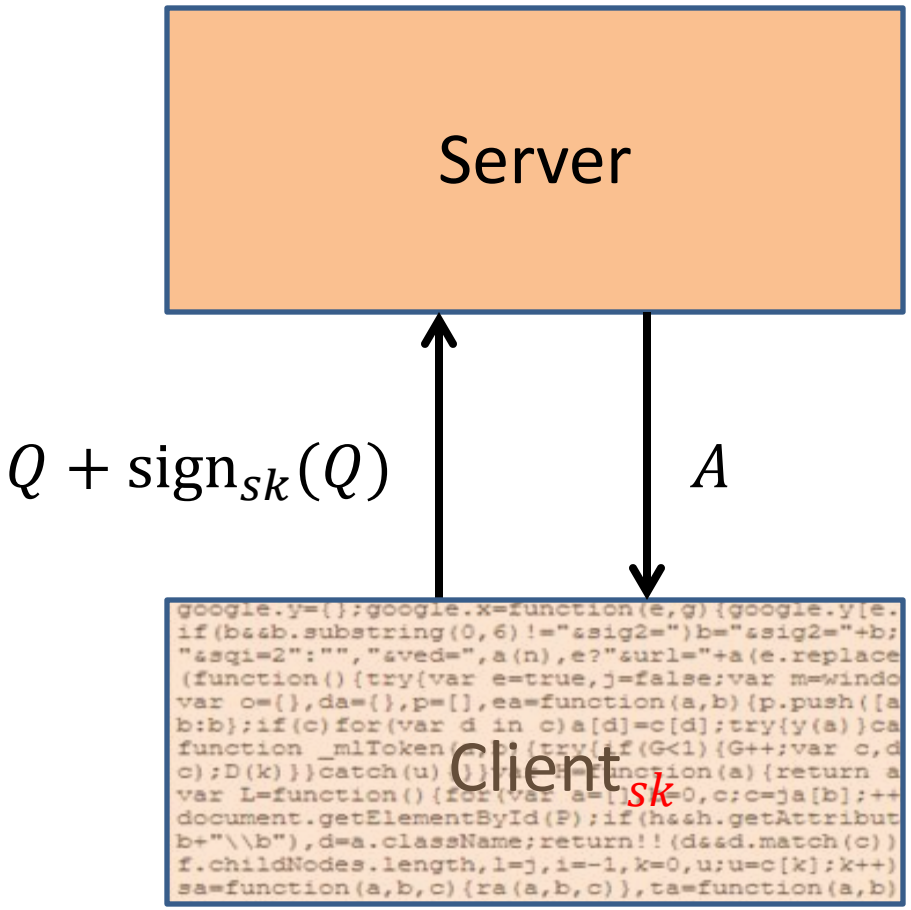
Authenticated Clients



Authenticated Clients



Authenticated Clients



Back to the Security Definition

$O(P)$ gives no more information than a **black-box** that computes P

For every PPT adversary A

There exists a PPT simulator S

Such that for every $P \in \Pi$:

$$A\big(O(P)\big) \rightarrow \ \approx_c \ \leftarrow S^P$$

Cannot Always be Achieved

$$A\big(O(P)\big) \rightarrow O(P) \qquad ? \quad \leftarrow S^P$$

2nd attempt: For every PPT adversary A

There exists a PPT simulator S

Such that for every $P \in \Pi$

and every predicate g:

$$\Pr_o\big[A\big(O(P)\big) = g(P)\big] \approx \Pr\left[S^P = g(P)\right]$$

Still Cannot Always be Achieved

$$\Pi = \{P_{a,b}\}_{a,b\in\{0,1\}^n}$$

$$P_{a,b}(x) = \begin{cases} b & \text{if } x = a \\ a,b & \text{if } x(a) = b \end{cases}$$

$$A\left(\tilde{P} \leftarrow O\left(P_{a,b}\right)\right) \rightarrow \tilde{P}\left(\tilde{P}\right) = a,b \quad ? \quad \leftarrow S^{P_{a,b}}$$

Obfuscating Point Functions

$$\Pi = \{P_a\}_{a \in \{0,1\}^n}$$

$$P_a(x) = \begin{cases} 1 & \text{if } x = a \\ 0 & \text{if } x \neq a \end{cases}$$

$O(P_a) \rightarrow \tilde{P}_{r,h}$ where: $r \leftarrow \$$, $\mathrm{h} \leftarrow \mathrm{HASH}(r, a)$

$$\tilde{P}_{r,h}(x) = \begin{cases} 1 & \text{if } \mathrm{HASH}(r, x) = \mathrm{HASH}(r, a) \\ 0 & \text{if } \mathrm{HASH}(r, x) \neq \mathrm{HASH}(r, a) \end{cases}$$

Constructions

$$P_{a,b}(x) = \begin{cases} b & \text{if } x = a \\ a, b & \text{if } x(a) = b \end{cases}$$

Impossible

All other functions?

Possible!

[Garg Gentry Halevi Raykova Sahai Waters 13]

$$P_a(x) = \begin{cases} 1 & \text{if } x = a \\ 0 & \text{if } x \neq a \end{cases}$$

Possible
(given a suitable hash function)

?