

Watermarking Cryptographic Functionalities from Standard Lattice Assumptions

Sam Kim and David J. Wu

Stanford University

How to Watermark a Image?



How to Watermark a Image?



How (Not) to Remove a Watermark



Removing the watermark *destroys* the image

Watermarking (Cryptographic) Programs

[NSS99, BGIRSVY01, HMW07, YF11, Nis13, CHNVW16, BLW17]

```
void serveur1(portServ ports)
{
    int sockServ1, sockServ2, sockClient;
    struct sockaddr_in monAddr, addrClient, addrServ2;
    socklen_t lenAddrClient;


    if ((sockServ1 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("Erreur socket");
        exit(1);
    }
    if ((sockServ2 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("Erreur socket");
        exit(1);
    }
}
```

Watermarking (Cryptographic) Programs

[NSS99, BGIRSVY01, HMW07, YF11, Nis13, CHNVW16, BLW17]

```
void serveur1(portServ ports)
{
    int sockServ1, sockServ2, sockClient;
    struct sockaddr_in monAddr, addrClient, addrServ2;
    socklen_t lenAddrClient;

    if ((sockServ1 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("Erreur socket");
        exit(1);
    }
    if ((sockServ2 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("Erreur socket");
        exit(1);
    }
}
```



Embed a string within the program

Watermarking (Cryptographic) Programs

[NSS99, BGIRSVY01, HMW07, YF11, Nis13, CHNVW16, BLW17]

```
void serveur1(portServ ports)
{
    int sockServ1, sockServ2, sockClient;
    struct sockaddr_in monAddr, addrClient, addrServ2;
    socklen_t lenAddrClient;

    if ((sockServ1 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("Erreur socket");
        exit(1);
    }
    if ((sockServ2 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("Erreur socket");
        exit(1);
    }
}
```

 Eurocrypt 2017

Embed a string within the program



Watermarking (Cryptographic) Programs

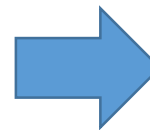
[NSS99, BGIRSVY01, HMW07, YF11, Nis13, CHNVW16, BLW17]

```
void serveur1(portServ ports)
{
    int sockServ1, sockServ2, sockClient;
    struct sockaddr_in monAddr, addrClient, addrServ2;
    socklen_t lenAddrClient;

    if ((sockServ1 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("Erreur socket");
        exit(1);
    }
    if ((sockServ2 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("Erreur socket");
        exit(1);
    }
}
```

 Eurocrypt 2017

Embed a string within the program



```
void serveur1(portServ ports)
{
    int sockServ1, sockServ2, sockClient;
    struct sockaddr_in monAddr, addrClient, addrServ2;
    socklen_t lenAddrClient;

    if ((sockServ1 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("Erreur socket");
        exit(1);
    }
    if ((sockServ2 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("Erreur socket");
        exit(1);
    }
}
```

If mark is removed, then program is destroyed

Watermarking (Cryptographic) Programs

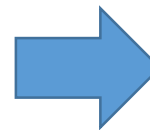
[NSS99, BGIRSVY01, HMW07, YF11, Nis13, CHNVW16, BLW17]

```
void serveur1(portServ ports)
{
  int sockServ1, sockServ2, sockClient;
  struct sockaddr_in monAddr, addrClient, addrServ2;
  socklen_t lenAddrClient;

  if ((sockServ1 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
    perror("Erreur socket");
    exit(1);
  }
  if ((sockServ2 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
    perror("Erreur socket");
    exit(1);
  }
}
```

 Eurocrypt 2017

Embed a string within the program



```
void serveur1(portServ ports)
{
  int sockServ1, sockServ2, sockClient;
  struct sockaddr_in monAddr, addrClient, addrServ2;
  socklen_t lenAddrClient;

  if ((sockServ1 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
    perror("Erreur socket");
    exit(1);
  }
  if ((sockServ2 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
    perror("Erreur socket");
    exit(1);
  }
}
```

If mark is removed, then program is destroyed

- Notion only achievable for functions that are not learnable

Watermarking (Cryptographic) Programs

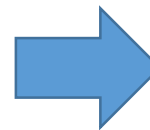
[NSS99, BGIRSVY01, HMW07, YF11, Nis13, CHNVW16, BLW17]

```
void serveur1(portServ ports)
{
    int sockServ1, sockServ2, sockClient;
    struct sockaddr_in monAddr, addrClient, addrServ2;
    socklen_t lenAddrClient;

    if ((sockServ1 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("Erreur socket");
        exit(1);
    }
    if ((sockServ2 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("Erreur socket");
        exit(1);
    }
}
```

 Eurocrypt 2017

Embed a string within the program



```
void serveur1(portServ ports)
{
    int sockServ1, sockServ2, sockClient;
    struct sockaddr_in monAddr, addrClient, addrServ2;
    socklen_t lenAddrClient;

    if ((sockServ1 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("Erreur socket");
        exit(1);
    }
    if ((sockServ2 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("Erreur socket");
        exit(1);
    }
}
```

If mark is removed, then program is destroyed

- Notion only achievable for functions that are not learnable
- Focus has been on cryptographic functions

Watermarking (Cryptographic) Programs

Existing constructions (that is robust against arbitrary removal strategies) all rely on indistinguishability obfuscation

[CHNVW16, BLW17]

Watermarking (Cryptographic) Programs

Existing constructions (that is robust against arbitrary removal strategies) all rely on indistinguishability obfuscation

[CHNVW16, BLW17]

Conceptually seems like an obfuscation-like primitive (embed a string within a program that an adversary cannot remove)

Main Result

Under standard lattice assumptions, there exists a (secretly-verifiable) watermarkable family of PRFs.

Main Result

Under standard lattice assumptions, there exists a (secretly-verifiable) watermarkable family of PRFs.

private puncturable PRFs
[BLW17, BKM17, CC17]

private programmable
PRFs [BLW17]

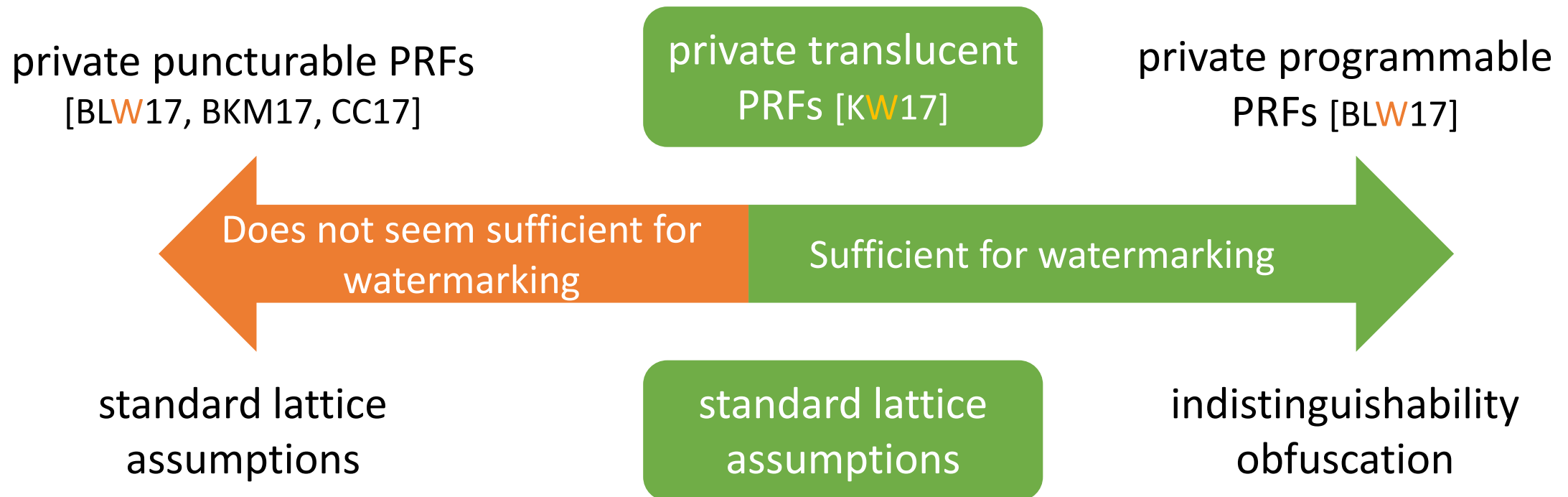


standard lattice
assumptions

indistinguishability
obfuscation

Main Result

Under standard lattice assumptions, there exists a (secretly-verifiable) watermarkable family of PRFs.



Main Result

Under standard lattice assumptions, there exists a (secretly-verifiable) watermarkable family of PRFs.

private puncturable PRFs
[BLW17, BKM17, CC17]

private translucent
PRFs [KW17]

private programmable
PRFs [BLW17]



Thank you!

<http://eprint.iacr.org/2017/380>