Public-Key Watermarking Schemes for Pseudorandom Functions

Rupeng Yang
HKU \Rightarrow UOW

Zuoxia Yu
HKU

Man Ho Au
HKU

Willy Susilo
UOW
Watermarking: A Cryptographic Program
Watermarking A Cryptographic Program
Watermarking A Cryptographic Program

Mark

Alice

Extract

mark key

extraction key

Key Generation

Alice

a program
Watermarking A Cryptographic Program

Correctness Requirement: Functionality Preserving
Watermarking A Cryptographic Program

Correctness Requirement: Extraction Correctness
Watermarking A Cryptographic Program

Mark

Alice

Remove

(Remove or Modify the Mark)

\( \approx \)

Computationally Difficult!!

Security Requirement: Unremovability
Watermarking a Cryptographic Program

It is *impossible* to watermark a learnable functionality.
It is *impossible* to watermark a learnable functionality.
Security Definitions of Watermarkable PRF

\[(MK, EK) \leftarrow \text{KeyGen} \]
\[K \leftarrow \mathcal{H} \]
\[C^* \leftarrow \text{Mark}(MK, K, m) \]
\[\tilde{C} \]

The adversary wins if:
1. \(C^* \approx \tilde{C}\)
2. \(\text{Extract}(EK, \tilde{C}) \neq m\)
(MK, EK) ← KeyGen
K ← \mathcal{K}

C^* ← \text{Mark}(MK, K, m)

The adversary wins if:
1. C^* ≈ \tilde{C}
2. Extract(EK, \tilde{C}) \neq m

Secret-Key Security
\[(MK, EK) \leftarrow \text{KeyGen} \]
\[K \leftarrow \mathcal{K} \]

\[C^\ast \leftarrow \text{Mark}(MK, K, m)\]
\[\overline{C} \]

The adversary wins if:
1. \(C^\ast \approx \overline{C}\)
2. \(\text{Extract}(EK, \overline{C}) \neq m\)

---

\[(MK, EK) \leftarrow \text{KeyGen} \]
\[K \leftarrow \mathcal{K} \]

\[C^\ast \leftarrow \text{Mark}(MK, K, m)\]
\[\overline{C} \]

The adversary wins if:
1. \(C^\ast \approx \overline{C}\)
2. \(\text{Extract}(EK, \overline{C}) \neq m\)

---

**Secret-Key Security**

**Public-Marking Security**
The adversary wins if:
1. $C^* \approx \tilde{C}$
2. $\text{Extract}(EK, \tilde{C}) \neq m$

Secret-Key Security

The adversary wins if:
1. $C^* \approx \tilde{C}$
2. $\text{Extract}(EK, \tilde{C}) \neq m$

Public-Marking Security

The adversary wins if:
1. $C^* \approx \tilde{C}$
2. $\text{Extract}(EK, \tilde{C}) \neq m$

Public-Extraction Security
\[(MK, EK) \leftarrow \text{KeyGen}\]
\[K \leftarrow \mathcal{K}\]

\[C^* \leftarrow \text{Mark}(MK, K, m)\]
\[\bar{C}\]

The adversary wins if:
1. \(C^* \approx \bar{C}\)
2. \(\text{Extract}(EK, \bar{C}) \neq m\)

**Secret-Key Security**

\[(MK, EK) \leftarrow \text{KeyGen}\]
\[K \leftarrow \mathcal{K}\]

\[\bar{C}\]

The adversary wins if:
1. \(C^* \approx \bar{C}\)
2. \(\text{Extract}(EK, \bar{C}) \neq m\)

**Public-Marking Security**

\[(MK, EK) \leftarrow \text{KeyGen}\]
\[K \leftarrow \mathcal{K}\]

\[m\]
\[C^* \leftarrow \text{Mark}(MK, K, m)\]
\[\bar{C}\]

The adversary wins if:
1. \(C^* \approx \bar{C}\)
2. \(\text{Extract}(EK, \bar{C}) \neq m\)

**Public-Extraction Security**

\[(MK, EK) \leftarrow \text{KeyGen}\]
\[K \leftarrow \mathcal{K}\]

\[EK\]
\[m\]
\[C^* \leftarrow \text{Mark}(MK, K, m)\]
\[\bar{C}\]

The adversary wins if:
1. \(C^* \approx \bar{C}\)
2. \(\text{Extract}(EK, \bar{C}) \neq m\)

**Public-Key Security**
The adversary wins if:
1. \( C^* \approx \tilde{C} \)
2. \( \text{Extract}(EK, \tilde{C}) \neq m \)

**Secret-Key Security**

\[(MK, EK) \leftarrow \text{KeyGen} \]
\[K \leftarrow \mathcal{K} \]

\[C^* \leftarrow \text{Mark}(MK, K, m) \]

\[\tilde{C} \]

\[\text{The adversary wins if:} \]
\[1. \text{ } C^* \approx \tilde{C} \]
\[2. \text{ } \text{Extract}(EK, \tilde{C}) \neq m \]

\[\text{Public-Marking Security} \]

\[(MK, EK) \leftarrow \text{KeyGen} \]
\[K \leftarrow \mathcal{K} \]

\[E K \]

\[\text{The adversary wins if:} \]
\[1. \text{ } C^* \approx \tilde{C} \]
\[2. \text{ } \text{Extract}(EK, \tilde{C}) \neq m \]

\[\text{Public-Extraction Security} \]

\[(MK, EK) \leftarrow \text{KeyGen} \]
\[K \leftarrow \mathcal{K} \]

\[\text{The adversary wins if:} \]
\[1. \text{ } C^* \approx \tilde{C} \]
\[2. \text{ } \text{Extract}(EK, \tilde{C}) \neq m \]

\[\text{Public-Key Security} \]

\[\text{👍 No authority (holding secret mark key and/or extraction key) is needed.} \]
\[(MK, EK) \leftarrow \text{KeyGen} \]
\[K \leftarrow \mathcal{H} \]
\[C^* \leftarrow \text{Mark}(MK, K, m) \]
\[\tilde{C} \]

The adversary wins if:
1. \(C^* \approx \tilde{C}\)
2. \(\text{Extract}(EK, \tilde{C}) \neq m\)

Secret-Key Security: [BLW17,KW17]

\[(MK, EK) \leftarrow \text{KeyGen} \]
\[K \leftarrow \mathcal{H} \]
\[M^* \leftarrow \text{Mark}(MK, K, m) \]
\[\tilde{C} \]

The adversary wins if:
1. \(M^* \approx \tilde{C}\)
2. \(\text{Extract}(EK, \tilde{C}) \neq m\)

Public-Marking Security: [QWZ18,KW19,YAYX20]

\[(MK, EK) \leftarrow \text{KeyGen} \]
\[K \leftarrow \mathcal{H} \]
\[C^* \leftarrow \text{Mark}(MK, K, m) \]
\[\tilde{C} \]

The adversary wins if:
1. \(C^* \approx \tilde{C}\)
2. \(\text{Extract}(EK, \tilde{C}) \neq m\)

Public-Extraction Security: [CHN+16,YAL+19]

\[(MK, EK) \leftarrow \text{KeyGen} \]
\[K \leftarrow \mathcal{H} \]
\[M^* \leftarrow \text{Mark}(MK, K, m) \]
\[\tilde{C} \]

The adversary wins if:
1. \(M^* \approx \tilde{C}\)
2. \(\text{Extract}(EK, \tilde{C}) \neq m\)

Public-Key Security: [??]?.

Public-Extraction Security: [CHN+16,YAL+19]
\((MK, EK) \leftarrow \text{KeyGen}\)
\(K \leftarrow \mathcal{K}\)

\(C^* \leftarrow \text{Mark}(MK, K, m)\)
\(C^* \leftarrow \text{Mark}(MK, K, m)\)

\(\bar{C}\)
\(\bar{C}\)

The adversary wins if:
1. \(C^* \approx \bar{C}\)
2. Extract\((EK, \bar{C}) \neq m\)

Secret-Key Security: [BLW17, KW17]

\((MK, EK) \leftarrow \text{KeyGen}\)
\(K \leftarrow \mathcal{K}\)

\(EK\)
\(m\)

\(C^* \leftarrow \text{Mark}(MK, K, m)\)
\(C^* \leftarrow \text{Mark}(MK, K, m)\)

\(\bar{C}\)
\(\bar{C}\)

The adversary wins if:
1. \(C^* \approx \bar{C}\)
2. Extract\((EK, \bar{C}) \neq m\)

Public-Extraction Security: [CHN+16, YAL+19]

\((MK, EK) \leftarrow \text{KeyGen}\)
\(K \leftarrow \mathcal{K}\)

\(MK\)
\(m\)

\(C^* \leftarrow \text{Mark}(MK, K, m)\)
\(C^* \leftarrow \text{Mark}(MK, K, m)\)

\(\bar{C}\)
\(\bar{C}\)

The adversary wins if:
1. \(C^* \approx \bar{C}\)
2. Extract\((EK, \bar{C}) \neq m\)

Public-Key Security: This Work
Constructing Public-Key Watermarkable PRF

A Watermarkable PRF with Public-Extraction Security:

\[ \text{Mark}'(mk, k, m) \rightarrow \begin{array}{c} m \\ C \\ C(x) \end{array} \rightarrow \text{Extract}'(ek, C) \rightarrow m \]
Constructing Public-Key Watermarkable PRF

A Watermarkable PRF with Public-Extraction Security:

\[
\begin{align*}
K &= (k, mk, ek) \\
F_K(x) &= F'_k(x)
\end{align*}
\]
Constructing Public-Key Watermarkable PRF

A Watermarkable PRF with Public-Extraction Security:

\[
\text{Mark}'(mk, k, m) \rightarrow \text{Extract}'(ek, C) \rightarrow m
\]

\[
K = (k, mk, ek)
\]

\[
F_K(x) = F'_k(x)
\]
Constructing Public-Key Watermarkable PRF

A Watermarkable PRF with Public-Extraction Security:

Constructing Public-Key Watermarkable PRF

A Watermarkable PRF with Public-Extraction Security:

\[\text{Mark}'(mk, k, m)\]

\[\text{Extract}'(ek, C)\]

\[k\]

\[F'_k(x)\]

\[\text{Mark}(K, m) = \text{Mark}'(mk, k, m)\]

\[F_K(x) = F'_k(x)\]

\[\text{Extract}(C) = \text{Extract}'(ek, C)\]

\[m\]

\[\text{How to Get } ek\text{ in the extraction algorithm?}\]
Constructing Public-Key Watermarkable PRF

A Watermarkable PRF with Public-Extraction Security:

\[ \text{Mark}'(mk, k, m) \quad \text{Extract}'(ek, C) \]

\[ K = (k, mk, ek) \quad \text{Mark}(K, m) = \text{Mark}'(mk, k, m) \]

\[ F_K(x) = F'_k(x) \quad \text{Extract}(C) = \text{Extract}'(ek, C) \]

How to Get \( ek \) in the extraction algorithm? Send \( ek \) to the extraction algorithm.
Constructing Public-Key Watermarkable PRF

\[ K = (k, mk, ek) \]
\[ F_k(x) = (F'_k(x), \text{Enc}(pk, ek)) \]

**(pk, sk)** is a key pair of a PKE scheme and is included in the public parameter of the watermarking scheme.
Constructing Public-Key Watermarkable PRF

\(\mathcal{K} = (k, mk, ek)\)

\(F_k(x) = (F'_k(x), \text{Enc}(pk, ek))\)

\(\text{Mark}(K, m) : \)

\(C_1 = \text{Mark'}(mk, k, m)\)

\(C(x) := (C_1(x), \text{Enc}(pk, ek))\)

\((\mathcal{P}, s)\) is a key pair of a PKE scheme and is included in the public parameter of the watermarking scheme.
Constructing Public-Key Watermarkable PRF

\[ K = (k, mk, ek) \]
\[ F_k(x) = (F'_k(x), \text{Enc}(pk, ek)) \]

Mark\((K, m)\):
\[ C_1 = \text{Mark}'(mk, k, m) \]
\[ C(x) := (C_1(x), \text{Enc}(pk, ek)) \]

Extract\((C)\):
\[ \text{Get } ek \text{ from } C(x) \text{ on uniform } x \]
\[ m' = \text{Extract}'(ek, C_1) \]

\((pk, sk)\) is a key pair of a PKE scheme and is included in the public parameter of the watermarking scheme.
Constructing Public-Key Watermarkable PRF

\( K = (k, mk, ek) \)

\( F_k(x) = (F'_k(x), \text{Enc}(pk, ek)) \)

\( \text{Mark}(K, m) : \)

\( C_1 = \text{Mark}'(mk, k, m) \)

\( C(x) := (C_1(x), \text{Enc}(pk, ek)) \)

\( \text{Extract}(C) : \)

Get \( ek \) from \( C(x) \) on uniform \( x \)

\( m' = \text{Extract}'(ek, C_1) \)

Distinguishable from random string given \( sk \) !

\( (pk, sk) \) is a key pair of a PKE scheme and is included in the public parameter of the watermarking scheme.
Constructing Public-Key Watermarkable PRF

\[ K = (k, mk, ek) \]

\[ F_k(x) = (F'_k(x), \text{Enc}(pk, ek)) \]

Mark\( (K, m) : \)
\[ C_1 = \text{Mark}'(mk, k, m) \]
\[ C(x) := (C_1(x), \text{Enc}(pk, ek)) \]

Extract\( (C) : \)
\[ \text{Get } ek \text{ from } C(x) \text{ on uniform } x \]
\[ m' = \text{Extract}'(ek, C_1) \]

Distinguishable from random string given \( sk \)!
Use robust unobfuscable PRF instead of PKE!
Constructing Public-Key Watermarkable PRF

\( K = (k, mk, ek) \)

\( F_k(x) = (F'_k(x), \text{Enc}(pk, ek)) \)

Distinguishable from random string given \( sk \)!
Use robust unobfuscatable PRF instead of PKE!

A PRF \( UF_{k_s}(\cdot) \)

A PRF key \( k_s \) is associated with a secret \( s \).

Pseudorandomness: \( UF_{k_s}(\cdot) \) is pseudorandom given oracle access to it.

Learnability: It is easy to get the secret \( s \) given a circuit \( C(\cdot) \equiv UF_{k_s}(\cdot) \).
Constructing Public-Key Watermarkable PRF

\[ K = (k, mk, ek) \]

\[ F_k(x) = (F'_k(x), \text{Enc}(pk, ek)) \]

\[ \text{Mark}(K, m) : \]
\[ C_1 = \text{Mark}'(mk, k, m) \]
\[ C(x) := (C_1(x), \text{Enc}(pk, ek)) \]

\[ \text{Extract}(C) : \]
\[ \text{Get } ek \text{ from } C(x) \text{ on uniform } x \]
\[ m' = \text{Extract}'(ek, C_1) \]

Distinguishable from random string given \( sk \) !
Use robust unobfuscatable PRF instead of PKE!

A PRF \( UF_{k_s}(\cdot) \)

A PRF key \( k_s \) is associated with a secret \( s \).

Pseudorandomness: \( UF_{k_s}(\cdot) \) is pseudorandom given oracle access to it.

Robust Learnability: It is easy to get the secret \( s \) given a circuit \( C(\cdot) \approx UF_{k_s}(\cdot) \).
Constructing Public-Key Watermarkable PRF

A PRF $UF_{k_s}(\cdot)$

A PRF key $k_s$ is associated with a secret $s$.

Pseudorandomness: $UF_{k_s}(\cdot)$ is pseudorandom given oracle access to it.

Robust Learnability: It is easy to get the secret $s$ given a circuit $C(\cdot) \approx UF_{k_s}(\cdot)$. 

$K = (k, mk, ek, k_{ek})$

$F_k(x) = (F'_k(x), UF_{k_{ek}}(x))$
Constructing Public-Key Watermarkable PRF

A PRF $\text{UF}_{k_s}(\cdot)$

A PRF key $k_s$ is associated with a secret $s$.

Pseudorandomness: $\text{UF}_{k_s}(\cdot)$ is pseudorandom given oracle access to it.

Robust Learnability: It is easy to get the secret $s$ given a circuit $C(\cdot) \approx \text{UF}_{k_s}(\cdot)$. 
Constructing Public-Key Watermarkable PRF

A PRF $\mathcal{UF}_{k_s}(\cdot)$

A PRF key $k_s$ is associated with a secret $s$.

Pseudorandomness: $\mathcal{UF}_{k_s}(\cdot)$ is pseudorandom given oracle access to it.

Robust Learnability: It is easy to get the secret $s$ given a circuit $C(\cdot) \approx \mathcal{UF}_{k_s}(\cdot)$.
Instantiating Public-Key Watermarkable PRF
Instantiating Public-Key Watermarkable PRF

Indistinguishability Obfuscation

[CHN+16]

Watermarkable PRF with Public Extraction

One-Way Function

Roust Unobfuscatable PRF

Fully Homomorphic Encryption

Public-Key Watermarkable PRF
Hinting Watermarkable PRF:

A hint associated with the PRF key can be used in the extraction algorithm.
Instantiating Public-Key Watermarkable PRF

- Indistinguishability Obfuscation
- [CHN+16]
- Watermarkable PRF with Public Extraction
- Public-Key Hinting Watermarkable PRF
- One-Way Function
- Fully Homomorphic Encryption
- Roust Unobfuscatable PRF
- Public-Key Watermarkable PRF
Constructing Public-Key Hinting Watermarkable PRF from \textbf{Puncturable PRF}

Correctness: if $x \neq x^*$, $F_k(x) = F_{k_{x^*}}(x)$
Correctness: if $x \neq x^*$, $F_k(x) = F_{k_{x^*}}(x)$

Pseudorandomness: $F_k(x^*)$ is hidden given $k_{x^*}$
Constructing Public-Key Hinting Watermarkable PRF from Puncturable PRF

\[ K = (k, x^*) \]
\[ \text{hint} = (x^*, z^*) \]
\[ F_k(x) = F'_k(x) \]

\( k \) is a PRF key of \( F' \).
\( x^* \) is a random input of \( F' \).
\( y^* = F'_k(x^*) \)
\( z^* = g(y^*) \)

\( F' \) is a puncturable PRF and \( g \) is an injective one-way function.
Constructing Public-Key Hinting Watermarkable PRF from Puncturable PRF

\[ K = (k, x^*) \]
\[ \text{hint} = (x^*, z^*) \]
\[ F_k(x) = F'_k(x) \]

Mark(\(K\)):
\[ C(x) := F'_{k^*}(x) \]

Marked
\[ C \]
\[ C(x) \]

\(k\) is a PRF key of \(F'\).
\(x^*\) is a random input of \(F'\).
\(y^* = F'_k(x^*)\)
\(z^* = g(y^*)\)

\(F'\) is a puncturable PRF and \(g\) is an injective one-way function.
Constructing Public-Key Hinting Watermarkable PRF from Puncturable PRF

\[ K = (k, x^*) \]
\[ \text{hint} = (x^*, z^*) \]
\[ F_k(x) = F'_k(x) \]

\[ \text{Mark}(K): \]
\[ C(x) := F'_{k^*}(x) \]

\[ \text{Marked} \]
\[ C \]
\[ C(x) \]

\[ \text{Extract}(\text{hint}, C): \]
Output marked iff \( g(C(x^*)) \neq z^* \)

\[ F' \text{ is a puncturable PRF and } g \text{ is an injective one-way function.} \]
Instantiating Public-Key Watermarkable PRF

[CHN+16]

Indistinguishability Obfuscation

Watermarkable PRF with Public Extraction

[SW14]

One-Way Function

Puncturable PRF

Fully Homomorphic Encryption

Public-Key Hinting Watermarkable PRF

Roust Unobfuscatable PRF

Public-Key Watermarkable PRF
Instantiating Public-Key Watermarkable PRF

- Indistinguishability Obfuscation
  - [CHN+16]
  - Watermarkable PRF with Public Extraction

- Lattice
  - [GVW12, …]
  - Bounded Functional Encryption
  - Public-Key Hinting Watermarkable PRF

- One-Way Function
  - [SW14]
  - Puncturable PRF
  - Round Unobfuscatable PRF

- Fully Homomorphic Encryption

- Public-Key Watermarkable PRF
Conclusion

Indistinguishability Obfuscation
[CHN+16]

Lattice
[GVW12, …]

Bounded Functional Encryption

One-Way Function
[SW14]

Puncturable PRF

Public-Key Hinting

Watermarkable PRF

with Public Extraction

Roust Unobfuscatable PRF

Fully Homomorphic Encryption

Public-Key

Watermarkable PRF
## Conclusion

<table>
<thead>
<tr>
<th>Assumption</th>
<th>Message-Embedding</th>
<th>$\varepsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lattice</td>
<td>✗</td>
<td>negl</td>
</tr>
<tr>
<td>Lattice + FHE</td>
<td>✗</td>
<td>$\frac{1}{6}$</td>
</tr>
<tr>
<td>Lattice</td>
<td>✓</td>
<td>$\frac{1}{\exp}$</td>
</tr>
<tr>
<td>iO</td>
<td>✓</td>
<td>negl</td>
</tr>
<tr>
<td>iO+FHE</td>
<td>✓</td>
<td>$\frac{1}{6}$</td>
</tr>
</tbody>
</table>
## Conclusion

<table>
<thead>
<tr>
<th>Assumption</th>
<th>Message-Embedding</th>
<th>$\varepsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lattice</td>
<td>$\times$</td>
<td>negl</td>
</tr>
<tr>
<td>Lattice + FHE</td>
<td>$\times$</td>
<td>$1/6$</td>
</tr>
<tr>
<td>Lattice</td>
<td>$\checkmark$</td>
<td>$1/exp$</td>
</tr>
<tr>
<td>iO</td>
<td>$\checkmark$</td>
<td>negl</td>
</tr>
<tr>
<td>iO+FHE</td>
<td>$\checkmark$</td>
<td>$1/6$</td>
</tr>
</tbody>
</table>

$(MK, EK) \leftarrow \text{KeyGen}$

$K \leftarrow \mathcal{K}$

$C^* \leftarrow \text{Mark}(MK, K, m)$

$C^{*}$

$\tilde{C}$

The adversary wins if:
1. $C^* \approx \tilde{C}$
2. $\text{Extract}(EK, \tilde{C}) \neq m$

$$\varepsilon = \frac{|\{x \in \mathcal{X} : C^*(x) \neq \tilde{C}(x)\}|}{|\mathcal{X}|}$$
## Open Problems

<table>
<thead>
<tr>
<th>Assumption</th>
<th>Message-Embedding</th>
<th>$\varepsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lattice</td>
<td>$\times$</td>
<td>$\text{negl}$</td>
</tr>
<tr>
<td>Lattice + FHE</td>
<td>$\times$</td>
<td>$1/6$</td>
</tr>
<tr>
<td>Lattice</td>
<td>$\checkmark$</td>
<td>$1/\exp$</td>
</tr>
<tr>
<td>iO</td>
<td>$\checkmark$</td>
<td>$\text{negl}$</td>
</tr>
<tr>
<td>iO+FHE</td>
<td>$\checkmark$</td>
<td>$1/6$</td>
</tr>
</tbody>
</table>

**Construct Public-Key Watermarkable PRFs with**

- message embedding and $\varepsilon \geq \text{negl}$ from lattice.
- constant $\varepsilon$ without using FHE.
- optimal $\varepsilon$ ($\varepsilon \approx 1/2$)
## Open Problems

<table>
<thead>
<tr>
<th>Assumption</th>
<th>Message-Embedding</th>
<th>$\varepsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lattice</td>
<td>$\times$</td>
<td>negl</td>
</tr>
<tr>
<td>Lattice + FHE</td>
<td>$\times$</td>
<td>$1/6$</td>
</tr>
<tr>
<td>Lattice</td>
<td>$\checkmark$</td>
<td>$1/exp$</td>
</tr>
<tr>
<td>iO</td>
<td>$\checkmark$</td>
<td>negl</td>
</tr>
<tr>
<td>iO + FHE</td>
<td>$\checkmark$</td>
<td>$1/6$</td>
</tr>
</tbody>
</table>

Construct Public-Key Watermarkable PRFs with

- message embedding and $\varepsilon \geq negl$ from lattice.
- constant $\varepsilon$ without using FHE.
- optimal $\varepsilon$ ($\varepsilon \approx 1/2$)

Thanks for your Attention!