Key idea: Let $\ell = |C|$ denote the number of input bits to $\mathcal{U}$ associated with the circuit C:

$$\text{mpk}: \begin{bmatrix} pk_1^{(0)} & \cdots & pk_\ell^{(0)} \\ pk_1^{(1)} & & pk_\ell^{(1)} \end{bmatrix}$$

pair of public keys / secret keys associated with each input wire (one for label 0 and one for label 1)

$$\text{msk}: \begin{bmatrix} sk_1^{(0)} & \cdots & sk_\ell^{(0)} \\ sk_1^{(1)} & & sk_\ell^{(1)} \end{bmatrix}$$

$sk_C : (sk_1^{(C_1)}, \dots, sk_\ell^{(C_\ell)})$ secret key for C consists of secret keys corresponding to the bits of C

Encrypt (mpk, x) : 1. Garble circuit $\mathcal{U} \to \tilde{\mathcal{U}}$, labels for bits of $x \in \{0,1\}^n$ and bits of $C \in \{0,1\}^\ell$

$$\begin{Bmatrix} L_1^{(0)}, \dots, L_n^{(0)} \\ L_1^{(1)}, \dots, L_n^{(1)} \end{Bmatrix} \qquad \begin{Bmatrix} L_{n+1}^{(0)}, \dots, L_{n+\ell}^{(0)} \\ L_{n+1}^{(1)}, \dots, L_{n+\ell}^{(1)} \end{Bmatrix}$$

2. Encrypt wire labels for input wires under mpk:

$$\begin{matrix} ct_1^{(0)} \leftarrow \text{Encrypt}(pk_1^{(0)}, L_{n+1}^{(0)}) & \cdots & ct_\ell^{(0)} \leftarrow \text{Encrypt}(pk_\ell^{(0)}, L_{n+\ell}^{(0)}) \\ ct_1^{(1)} \leftarrow \text{Encrypt}(pk_1^{(1)}, L_{n+1}^{(1)}) & & ct_\ell^{(1)} \leftarrow \text{Encrypt}(pk_\ell^{(1)}, L_{n+\ell}^{(1)}) \end{matrix}$$

3. Output $ct = \left(\tilde{\mathcal{U}}, \{L_i^{(x_i)}\}_{i \in [n]}, \{ct_i^{(b)}\}_{i \in [\ell], b \in \{0,1\}}\right)$

Decrypt (sk_C, ct_x) : 1. Using $sk_i^{(C_i)}$, decrypt $ct_i^{(C_i)}$ to obtain $L_{n+i}^{(C_i)}$.

2. Evaluate garbled circuit $\tilde{\mathcal{U}}$ with labels $L_i^{(x_i)}$, $L_{n+i}^{(C_i)}$.

Correctness: Follows by garbled circuit correctness and PKE correctness. Namely, evaluator has garbled circuit $\tilde{\mathcal{U}}$ and labels for C and x. Evaluation yields $C(x)$.

Security (Sketch). Let $C : \{0,1\}^n \to \{0,1\}^m$ be the circuit the adversary requests (e.g., in a selective security setting). Consider the challenge ciphertext together with the key for C:

$$\left(\tilde{\mathcal{U}}, \{L_i^{(x_i)}\}_{i \in [n]}, \{ct_i^{(C_i)}\}_{i \in [\ell]}, \{ct_i^{(1-C_i)}\}_{i \in [\ell]}, \{sk_i^{(C_i)}\}_{i \in [\ell]}\right)$$

$$\stackrel{c}{\approx} \text{ (PKE security)}$$

$$\left(\tilde{\mathcal{U}}, \{L_i^{(x_i)}\}_{i \in [n]}, \{ct_i^{(C_i)}\}_{i \in [\ell]}, \{\text{Encrypt}(pk_i^{(b)}, 0^{|L_i^{(1-C_i)}|})\}_{i \in [\ell]}, \{sk_i^{(C_i)}\}_{i \in [\ell]}\right)$$

$$\stackrel{c}{\approx} \text{ (garbling security)}$$

$$\left(S(1^\lambda, C, C(x)), \{\text{Encrypt}(pk_i^{(b)}, 0^{|L_i^{(1-C_i)}|})\}_{i \in [\ell]}, \{sk_i^{(C_i)}\}_{i \in [\ell]}\right)$$

simulator for garbled circuit (and outputs encryption of wire labels)

Key observation: Ciphertexts can be simulated given only $(C, C(x))$. In FE security game (the indistinguishability-based one), adversary must choose x_0 and x_1 such that $C(x_0) = C(x_1)$.

Drawback: Ciphertexts are large! In fact, as large as the function that is applied to it.

Ideally: ciphertext size is independent (or sublinear) in size of functions.

- Succinct FE: running time of encryption is independent of the size of supported functions (or: only depends on depth)
- Compact FE: running time of encryption also independent of output length of supported functions

Single-key compact FE (for NC^1 circuits) $\Longrightarrow$ indistinguishability obfuscation

Open question: Compact FE from lattices.

Constructing succinct FE turns out to be easier. We will rely on ABE + FHE + garbled circuits.

complement of function f

We will need to tweak the ABE scheme to encrypt two messages μ_0 and μ_1: (for convenience)

if $f(x) = 0$, decryption outputs μ_0 $\longrightarrow$ can be built from vanilla ABE by encrypting μ_0 under f and μ_1 under $\overline{f}$

if $f(x) = 1$, decryption outputs μ_1

(recall convention: $f(x)=0$ means decryption succeeds)

Goal: Ciphertext size should be sublinear in size of circuit being evaluated

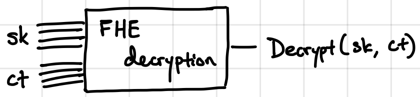
Idea: Use FHE to encrypt the input x.

Given FHE encryption of x $\longrightarrow$ FHE encryption of $f(x)$

Need a way to decrypt $f(x)$.

How to give out a "constrained" decryption algorithm. Should only allow decrypting an encryption of $f(x)$, but not any other ciphertext.

Give out a garbled circuit that implements FHE decryption.



Ciphertext includes wire labels of the bits of the FHE secret key.

Decrypter needs a way to obtain labels for ciphertext (should only obtain labels for encryption of $f(x)$).

Encrypt wire labels of ciphertext using the ABE scheme where the attribute is the FHE encryption of x:

$$\text{ABE.Encrypt}(\text{mpk}, x, L_i^{(0)}, L_i^{(1)})$$

where $L_i^{(0)}, L_i^{(1)}$ are the labels for the wires associated with the i^{th} bit of the ciphertext

The ciphertext is then

- Garbled circuit for FHE decryption circuit
- Wire labels for FHE decryption key
- ABE encryptions of wire labels for FHE ciphertext [we use an independent ABE scheme to encrypt labels for each bit]

To construct a decryption key for a function f

Let ct be an FHE ciphertext

Let f_i be the function that takes ct and outputs $[\text{FHE.Eval}(f, ct)]_i$ — i^{th} bit of ciphertext output by $\text{FHE.Eval}(f_i, ct)$.

Let ℓ be the length of an FHE ciphertext. Issue ABE secret keys for $\underline{f_1, \dots, f_\ell}$.

$\hookrightarrow$ recall that these keys are associated with ℓ independent ABE schemes

To decrypt, use $sk_{f_1}, \dots, sk_{f_\ell}$ to recover wire labels for

$$ct_{f(x)} \leftarrow \text{FHE.Eval}(f, ct_{\tilde{x}})$$

Evaluate the garbled circuit to learn

$$f(x) \leftarrow \text{FHE.Decrypt}(sk, ct_{f(x)})$$

Succinctness: Size of ciphertext: Garbled circuit for FHE decryption: $\text{poly}(\lambda, d)$ where d is the depth of the computation

Wire labels for FHE secret key: $|sk| \cdot \text{poly}(\lambda) = \text{poly}(\lambda, d)$

ABE encryption of wire labels for FHE ciphertext: $\ell \cdot \text{poly}(\lambda, d) = \text{poly}(\lambda, d)$

Overall size scales with depth of circuit rather than size.

Key idea: Basic scheme from PKE but instead of evaluating f using the garbled circuit, we instead evaluate the FHE decryption function, which has complexity smaller than f

Security: ABE security: labels not associated with $ct_{f(x)}$ hidden by semantic security

removes dependence on FHE secret key

Garbling security: Can simulate garbled circuit + labels given only FHE.Decrypt$(\cdot, \cdot)$ and $f(x)$

FHE security: Replace encryption of x_0 with x_1

Still only secure in the single-key setting (since garbled circuit is not reusable)