Deniable Fully Homomorphic Encryption from LWE

Shweta Agrawal, Shafi Goldwasser, Saleet Mossel
Crypto, 2021 (To appear)
Deniable Fully Homomorphic Encryption from LWE

Shweta Agrawal, Shafi Goldwasser, Saleet Mossel
Crypto, 2021 (To appear)

Most slides by Saleet Mossel
Deniable FHE

The notion of Deniable FHE
Deniable FHE

Bob, for whom did you vote?

\[ ct_0 = Enc(pk, b_0; r) \]

\[ ct_0, ct_1, \ldots, ct_n \]

\[ ct^* = Eval(\Sigma_{i=0}^n, ct_0, \ldots, ct_n) \]

\[ Dec(sk, ct^*) = \Sigma_{i=0}^n b_i \]
Deniable FHE

\[
ct_0 = \text{Enc}(pk, b_0; r) = \text{Enc}(pk, \bar{b}_0; r')
\]

\[
\{pk, \text{Enc}(pk, b_0; r), \bar{b}_0, r'\} \approx_c \{pk, \text{Enc}(pk, \bar{b}_0; r), \bar{b}_0, r\}
\]

"Fake" Distribution

"Honest" Distribution

Bob, for whom did you vote?

\[
ct_0 = \text{Enc}(pk, b_0; r) = \text{Enc}(pk, \bar{b}_0; r)
\]

\[
r' \leftarrow \text{Fake}(pk, b_0, r, \bar{b}_0)
\]

\[
\sum_{i=0}^{n} b_i
\]
Elections require Deniability & FHE

- Benefit of Deniable Encryption in Elections:
  - Honest Participation

- Benefit of Fully Homomorphic Encryption in Elections:
  - Homomorphically compute the voting result

- Any data-driven algorithm
- Store Encrypted Data
- Natural upgrade for all DE apps!
Deniable Encryption

• Introduced by Canetti, Dwork, Naor and Ostrovsky 1997
  • construction from trapdoor permutations, unique SVP
  • size of ct is the inverse of the detection probability

• Weak Deniable Encryption
  • can also lie about the encryption algorithm (Enc, Denc)
  • construction with compact ct and negligible deniability

• Lower bound (Efficiency vs. Deniability)
  • It seems inherent that the length of ct grows with the inverse of the detection probability in “separable” constructions.

• A significant step forward [SW14]
  • construction from iO and OWF
  • compact ct and negligible deniability

What does this mean given recent iO results?
Deniable Encryption

CDNO
• Based on TDP
• CT size inverse of detection prob

SW
• Based on iO
• CT size indpt of detection prob

1997
In full model, nothing else known!

2014
Our Results

• Notion of **Deniable FHE** (full and weak)

• Constructions based on **Learning With Errors**

• **Compact** \( ct \): size does not depend on detection probability!
  • Our construction is separable (so not inherent)
  • Total encryption **time** grows with the inverse of the detection probability!

• **Support large message space**
  • All prior work encode large messages **bit by bit**

• **Offline–Online Encryption**
  • Online time independent of the detection probability
Our Results: Deniable Encryption

CDNO, 1997
CT size inverse of detection prob

SW, 2014
CT size indpt of detection prob

This Work
• CT size independent of detection prob
• (Offline) encryption time inverse of detection prob
Via special properties in Fully Homomorphic Encryption!
Fully Homomorphic Encryption

Can be built using LWE (BV11, BGV12, GSW13⋯)

Expressive Functionality:
Supports arbitrary circuits

Compact ciphertext, independent of circuit size

Encryption and function evaluation commute!
Enc(f(x)) =* f(Enc(x))

*: roughly
Adding Deniability to the Mix

• A Deniable FHE scheme \((\text{Gen}, \text{Enc}, \text{Eval}, \text{Dec}, \text{Fake})\)

  • \((\text{Gen}, \text{Enc}, \text{Eval}, \text{Dec})\) is an FHE scheme

  • \((\text{Gen}, \text{Enc}, \text{Dec}, \text{Fake})\) is a Deniable Encryption scheme
Deniable FHE

A Deniable FHE scheme \((\text{Gen, Enc, Eval, Dec, Fake})\) syntax

- \(\text{Gen} \to (pk, sk)\)
- \(\text{Enc}(pk, m; r) = ct\)
- \(\text{Dec}(sk, ct) = b\)
- \(\text{Eval}(pk, f, ct_1, ..., ct_k) = ct^*\)
- \(\text{Fake}(pk, b, r, \overline{b}) \to r'\)
Deniable FHE

A Deniable FHE scheme \((Gen, Enc, Eval, Dec, Fake)\)

1. Correctness
2. CPA-Security
3. Deniability
4. Compactness
Correctness versus Deniability

Correctness:

For every $f$ and $m_1, \ldots, m_k$:

$$\Pr[\text{Dec}(sk, \text{Eval}(pk, f, ct_1, \ldots, ct_k)) = f(m_1, \ldots, m_k)] = 1 - \text{negl}$$

where $ct_i \leftarrow \text{Enc}(pk, m_i)$ and $(pk, sk) \leftarrow \text{Gen}$

Cannot simultaneously satisfy perfect correctness and deniability
δ(\lambda) - Deniability

We consider (inverse) polynomial deniability

For every bit \( b \), and PPT adversary \( A \)

\[
|\Pr[A(pk, Enc(pk, b; r), b, r)] - \Pr[A(pk, Enc(pk, \overline{b}; r), b, r')]| \leq \delta(\lambda)
\]

"Honest" Distribution

"Fake" Distribution

where \((pk, sk) \leftarrow Gen\), \(r \leftarrow \{0, 1\}^\ell\), and \(r' \leftarrow Fake(pk, \overline{b}, r, b)\)
Evaluation & Deniability Compactness

a) For every $f$ and $m_1, \ldots, m_k$:

$$|\text{Eval}(pk, f, ct_1, \ldots, ct_k)| \leq \text{poly}$$

where $ct_i \leftarrow \text{Enc}(pk, m_i)$ and $(pk, sk) \leftarrow \text{Gen}$

Independent of $k$ and the complexity of $f$

b) For every $m$:

$$|\text{Enc}(pk, m)| \leq \text{poly}$$

where $(pk, sk) \leftarrow \text{Gen}$, regardless of the encryption running time

Independent of the detection probability
Deniable FHE
Our Construction of Deniable FHE
FHE from LWE: A Very Brief Recap

• All* known FHE schemes add noise in CT for security.
• Homomorphic evaluation of CTs (eval(f, ct_1\ldots ct_n)) cause noise to grow
• Kills correctness after noise grows too much
• Limits number of homomorphic operations

How to keep going: Gentry’s bootstrapping [Gen09]!
The Magic of Bootstrapping

• Assume that an FHE is powerful enough to support evaluation of its own decryption circuit $\text{Dec}$.

• By correctness of decryption, $\text{Dec}(ct_x, sk) = x$

\[
\text{Dec}\left(\begin{array}{c} x \\ sk \end{array}\right) = x
\]

• Define circuit $\text{Dec}_{ct}(sk) = \text{Dec}(sk, ct)$

• By correctness of homomorphically evaluation, $\text{Eval}(F, ct_x) = ct(F(x))$

\[
\text{Eval}\left(\text{Dec}_{ct}, \begin{array}{c} sk \end{array}\right) = \text{Dec}_{ct}(sk) = x
\]
The Magic of Bootstrapping

• Originally introduced to reduce noise in evaluated ciphertext

• Homomorphic evaluation of decryption
  • removes large old noise
  • adds small new noise (size small since decryption shallow)

This work: Oblivious Sampling of FHE ciphertexts!
The Magic of Bootstrapping

• Assume that decryption always outputs 0 or 1
  • even if input ct is not well formed

• Then, bootstrapping always outputs proper encryption of 0 or 1!

\[
\text{Eval} \left( \text{Dec}_{ct}, \text{sk} \right) = \text{Dec}_{ct}(\text{sk}) = x
\]

Even if input “ct” is a random element in ciphertext space!
The Magic of Bootstrapping

• Assume that decryption outputs 0 w.o.p. for random input

• Then, bootstrapping outputs encryption of 0 w.o.p for random input

\[
\text{Eval } \left( \text{Dec}_{\text{rand}}, \text{sk} \right) = \text{Dec}_{\text{rand}}(\text{sk}) = 0
\]

Given \( \text{enc}(\text{sk}) \), run \( \text{dec} \) homomorphically on random to generate encryption of 0 w.o.p!
But, wait a minute…

- Given encryption of $1$, decryption outputs $1$ w.o.p.
- Encryption of $1$ is indistinguishable from random!

$$\text{Eval} \left( \text{Dec}_{ct1}, \text{sk} \right) = \text{Dec}_{ct1}(\text{sk}) = 1$$

- Can pretend as if $ct1 = \text{enc}(1)$ is a random string

Pretend bootstrapping outputs $\text{enc}(0)$ but actually $\text{enc}(1)$!
Can provide randomness $R$ so it looks like $\text{Bootstrap}(R) = \text{enc}(0)$ but actually $\text{enc}(1)$

OK... but why is this useful?
Leveraging our trick (binary msg space)

• Let $B(x) = Eval(pk, Dec_x, ct_{sk})$ the bootstrapping procedure
  • recall $Dec_x(sk) = Dec(sk, x)$

• Denote homomorphic addition (mod 2) as
  \[ Eval(pk, +, ct_a, ct_b) = ct_a \oplus ct_b \]
  \[ B(R_i) \oplus \cdots \oplus B(R_n) = Enc(\text{Parity} \ (x_1, \ldots, x_n)) \]
Construction

Gen:
1. \((pk, sk) \leftarrow Gen\)
2. \(ct_{sk} \leftarrow Enc(pk, sk)\)
3. Output \(pk = (pk, ct_{sk}), sk = sk\)
Construction

\[ rand = (x_1, \ldots, x_n, \{R_i\}_{x_i=0}, \{r_i\}_{x_i=1}) \]

**Enc(pk, b):**

1. Sample \( x_1, \ldots, x_n \leftarrow \{0,1\} \) s.t. \( \sum_i x_i = b \pmod{2} \)
2. For \( x_i = 0 \), sample \( R_i \leftarrow \mathcal{R}^\ell \)
3. For \( x_i = 1 \), sample \( r_i \leftarrow \{0,1\}^\ell' \) and set \( R_i = Enc(pk, 1; r_i) \)
4. Compute \( ct = B(R_1) \oplus \cdots \oplus B(R_n) \)
5. Output \( ct \)

\( B(\mathcal{R}^\ell) \) is a valid encryption of 0 w.h.p
Construction

\( \text{Fake}(pk, b, rand, \overline{b}) : \)

1. If \( b = \overline{b} \), output \( \text{rand} \)
2. Sample \( k \leftarrow [n] \) s.t. \( x_k = 1 \)
3. Set \( x'_k = 0 \) and \( R'_k = \text{Enc}(pk, 1; r_k) \)
4. For \( i \neq k \), set \( R'_i = R_i \) and \( r'_i = r_i \)
5. Output \( \text{rand}' = (x'_1, ..., x'_n, \{R'_i\}_{x'_i=0}, \{r'_i\}_{x'_i=1}) \)

By pretending one ciphertext \( \text{enc}(1) \) is random, parity flipped!

Statistical distance from honest dist is \( 1/\text{poly}(n) \)
Construction

$Eval(pk, f, ct_1, ..., ct_k)$:
1. Interpret $ct_i$ as special FHE ciphertext $ct_i$
2. Output $Eval(pk, f, ct_1, ..., ct_k)$

$Dec(dsk, ct)$:
1. Interpret $ct$ as special FHE ciphertext $ct$
2. Output $Dec(sk, ct)$

As before!
Special FHE
Definition and Instantiation
Special FHE

Circular security

Pseudo-random CT

Det. eval and dec

$B(\mathcal{R}^\ell)$ is a valid encryption of 0 w.h.p

- Can be removed
- Usually OK
- Can be weakened whp to wnnp
Weaker Special FHE

1. Pseudorandom Ciphertext
2. Deterministic evaluation and decryption
3. Decryption always outputs a valid message

\[ \Pr[\text{Dec}(sk, R) = 0] = \frac{1}{\text{poly}} \]

where \( R \leftarrow \mathcal{R}^{\ell} \) and \((pk, sk) \leftarrow \text{Gen}\)

[BGV14] FHE satisfies all properties!
Online-Offline Encryption

Bulk of the computation is independent of the message, and may be performed in an offline pre-processing phase.

Enc(dpk, b):
1. Select \( x_1, ..., x_n \leftarrow \{0,1\} \) s.t. \( \sum_i x_i = b (mod \ 2) \)
2. For \( x_i = 0 \), select \( R_i \leftarrow \mathcal{R}^\ell \)
3. For \( x_i = 1 \), select \( r_i \leftarrow \{0,1\}^{\ell'} \) and set \( R_i = Enc(pk, 1; r_i) \)
4. Output \( dct = B(R_1) \oplus \cdots \oplus B(R_n) \)

\( n-1 \) computations of \( B(R_i) \) can be done offline: choose \( R_n \) depending on \( b \) and compute \( B(R_n) \) online.
Main Takeaway:
Evaluation compactness in FHE implies deniability compactness in DE!
Going Forward

• Compact CT \( \Rightarrow \) compact encryption runtime?
  • Analogy to FE [LPST16,GKPVZ13]

• Technical barrier: unidirectional cheating

• Need: Invertible oblivious sampling with bias
  • SW construction may be viewed through this lens

• From LWE: can have oblivious sampling with bias (this work) or oblivious sampling with inversion but not both (so far).

Thank You

Images Credit: Hans Hoffman