Notes on Two Fully Homomorphic Encryption Schemes Without Bootstrapping

Yongge Wang

UNC Charlotte
Charlotte, NC 28223, USA
yonwang@uncc.edu

Abstract. Recently, IACR ePrint archive posted two fully homomorphic encryption schemes without bootstrapping. In this note, we show that these schemes are trivially insecure.

1 Introduction

Though it is a very challenging problem to design fully homomorphic encryption schemes without bootstrapping. We still see that quite a few researchers post candidate designs frequently. This note points out that the two schemes posted to IACR ePrint archive recently are trivially insecure: the scheme by Masahiro Yagisawa [4] on 2015-05-19 and the scheme by Dongxi Liu [3] on 2015-05-17.

2 Masahiro Yagisawa [4]'s Scheme

Octonion (see, e.g., Conway and Smith [2] or Baez [1]) is the largest of the four normed division algebra and is the only normed division algebra that is neither commutative nor associative. Each octonion number is a vector $\mathbf{a} = [a_0, \cdots, a_7] \in R^8$ where R is the real number. For each octonion number $\mathbf{a} = [a_0, \cdots, a_7]$, we define an associated 8×8 matrix

$$A_{\mathbf{a}} = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ -a_1 & a_0 & a_4 & a_7 & -a_2 & a_6 & -a_5 & -a_3 \\ -a_2 & -a_4 & a_0 & a_5 & a_1 & -a_3 & a_7 & -a_6 \\ -a_3 & -a_7 & -a_5 & a_0 & a_6 & a_2 & -a_4 & a_1 \\ -a_4 & a_2 & -a_1 & -a_6 & a_0 & a_7 & a_3 & -a_5 \\ -a_5 & -a_6 & a_3 & -a_2 & -a_7 & a_0 & a_1 & a_4 \\ -a_6 & a_5 & -a_7 & a_4 & -a_3 & -a_1 & a_0 & a_2 \\ -a_7 & a_3 & a_6 & -a_1 & a_5 & -a_4 & -a_2 & a_0 \end{pmatrix}$$

For two octonions $\mathbf{a} = [a_0, \cdots, a_7]$ and $\mathbf{b} = [b_0, \cdots, b_7]$, we can add them as $\mathbf{a} + \mathbf{b} = [a_0 + b_0, \cdots, a_7 + b_t]$ and multiply them as $\mathbf{ab} = \mathbf{b}A_{\mathbf{a}}$. The norm of an octonion $\mathbf{a} = [a_0, \cdots, a_7]$ is defined as $\|\mathbf{a}\| = \sqrt{a_0^2 + \cdots + a_7^2}$.

Using octonions over $GF(q)$, Yagisawa [4] introduced a fully homomorphic encryption scheme. Though Yagisawa [4] defined his fully homomorphic encryption scheme in terms of a sequence of private octonion numbers, the scheme could be simplified using matrix operations. Let $GF(q)$ be the underlying finite field that we will work with. Let $\mathbf{1} = [1, 0, 0, 0, 0, 0, 0, 0]$ and $\mathbf{z} \in GF(q)^8$ be a random octonion with $\|\mathbf{z}\| = 0$ and $z_0 \neq 0$. Then the protocol works as follows:

Key Setup. Choose a random invertible 8×8 matrix $K \in GF(q)^{8\times 8}$. K is the private key.

Encryption. For a message $m \in GF(q)$, choose a random $r \in GF(q)$ and compute the cipher text $C_m = \texttt{M.Enc}(K, m) = K^{-1} A_{m\mathbf{1}+r\mathbf{z}} K \in GF^{8\times 8}$ where $A_{m\mathbf{1}+r\mathbf{z}}$ is the associated matrix for $m\mathbf{1} + r\mathbf{z}$ when $m\mathbf{1} + r\mathbf{z}$ is considered as an octonion number.

Decryption. For a received ciphertext C_m, compute $A_{m\mathbf{1}+r\mathbf{z}} = \texttt{M.Dec}(K, C_m) = KC_m K^{-1}$. The plaintext message m can then be recovered by finding an octonion $\mathbf{u}$ such that $\|\mathbf{u}\| = 0$ and $\mathbf{1}A_{m\mathbf{1}+r\mathbf{z}} = m\mathbf{1} + \mathbf{u}$.

Ciphertext addition. The addition of two ciphertexts C_{m_0} and C_{m_1} is defined as the component wise addition $C_{m_0+m_1} = C_{m_0} + C_{m_1}$. That is, this is just the regular matrix addition.

Ciphertext multiplication. The multiplication of two ciphertexts C_{m_0} and C_{m_1} is defined as the regular matrix multiplication

$$C_{m_0 \times m_1} = C_{m_1} C_{m_0} = KA_{m_1} K^{-1} KA_{m_0} K^{-1} = KA_{m_1} A_{m_0} K^{-1}.$$

It is straightforward to observe that for the above scheme, the message 0 is encrypted to a matrix C_m such that $\|\mathbf{1}C_m\| = 0$. In other words, we can easily distinguish the ciphertext of m and $-m$ since $\|\mathbf{1}(C_m + C_{-m})\| = 0$.

2.1 Yagisawa's [4] original encryption scheme

It should be noted that our matrix operation based encryption scheme is equivalent to Yagisawa's [4] original encryption scheme when messages are chosen from $GF(q)$. If the scheme is considered a homomorphic scheme over octonion $GF(q)^8$, then our scheme is not equivalent to to original scheme. In the following, we briefly describe the original scheme in [4].

Key Setup. Let $\mathbf{x}$ be a variable representing octonions. Choose random invertible octonions $\mathbf{k}_0, \cdots, \mathbf{k}_{t-1} \in GF(q)^8$. The private key is $\texttt{key} = \{\mathbf{k}_0, \cdots, \mathbf{k}_{t-1}\}$.

Encryption. For a message $\mathbf{m} \in GF(q)^8$, the cipher text $C_{\mathbf{m}}(\mathbf{x}) = \texttt{O.Enc}(\texttt{key}, \mathbf{m}) = \mathbf{k}_0(\cdots(\mathbf{k}_{t-1}(\mathbf{m}(\mathbf{k}_{t-1}^{-1}(\cdots(\mathbf{k}_0^{-1}\mathbf{x})\cdots)$.

Decryption. Let $g_0(\mathbf{x}) = \mathbf{k}_{t-1}^{-1}(\cdots(\mathbf{k}_0^{-1}\mathbf{x})\cdots)$ and $g_1(\mathbf{x}) = \mathbf{k}_1(\cdots(\mathbf{k}_{t-1}\mathbf{x})\cdots)$. For a received ciphertext $C_m(\mathbf{x})$, compute $\mathbf{m} = \texttt{O.Dec}(\texttt{key},(C_m(\mathbf{x})) = g_0(C_m(g_1(\mathbf{1})))$.

Ciphertext addition. The addition of two ciphertexts $C_{\mathbf{m}_0}(\mathbf{x})$ and $C_{\mathbf{m}_1}(\mathbf{x})$ is defined as the component wise addition $C_{\mathbf{m}_0+\mathbf{m}_1}(\mathbf{x}) = C_{\mathbf{m}_0}(\mathbf{x}) + C_{\mathbf{m}_1}(\mathbf{x})$. That is, this is just the octonion addition.

Ciphertext multiplication. The multiplication of two ciphertexts $C_{\mathbf{m}_0}(\mathbf{x})$ and $C_{\mathbf{m}_1}(\mathbf{x})$ is defined as $C_{\mathbf{m}_1\mathbf{m}_0} = C_{\mathbf{m}_1}(C_{\mathbf{m}_0}(\mathbf{x}))$.

2.2 Differences of the two encryption scheme

If Yagisawa's scheme is only used to encrypt messages in $GF(q)$ (instead of octonion messages in $GF(q)^8$) by mapping a message $m \in GF(q)$ to $m\mathbf{1} + r\mathbf{z} \in GF(q)^8$, then it is straightforward to check that our matrix operation based scheme is equivalent to Yagisawa's original scheme. However, if Yagisawa's scheme is used to encrypt messages octonion messages in $GF(q)^8$, then these two schemes are not equivalent. Masahiro Yagisawa constructed the following counter example.

Let $\mathbf{m}_0, \mathbf{m}_1, \mathbf{m}_2$ be invertible octonions such that $\mathbf{m}_0(\mathbf{m}_1\mathbf{m}_2) \neq (\mathbf{m}_0\mathbf{m}_1)\mathbf{m}_2$ and $\mathbf{m}_0^{-1}, \mathbf{m}_1^{-1}, \mathbf{m}_2^{-1}$ are inverses of $\mathbf{m}_0, \mathbf{m}_1, \mathbf{m}_2$ respectively. Then for Yagisawa's scheme, we have

$$
\begin{aligned}
c_0 &= \texttt{O.Enc}(\texttt{key},\mathbf{m}_0(\mathbf{m}_1\mathbf{m}_2)) = C_{\mathbf{m}_0}(C_{\mathbf{m}_1}(C_{\mathbf{m}_2}(\mathbf{x}))) \\
c_1 &= \texttt{O.Enc}(\texttt{key},\mathbf{m}_2^{-1}(\mathbf{m}_1^{-1}\mathbf{m}_0^{-1})) = C_{\mathbf{m}_2^{-1}}(C_{\mathbf{m}_1^{-1}}(C_{\mathbf{m}_0^{-1}}(\mathbf{x})))
\end{aligned}
$$

Since the inverse of $(\mathbf{m}_0\mathbf{m}_1)\mathbf{m}_2$ is $\mathbf{m}_2^{-1}(\mathbf{m}_1^{-1}\mathbf{m}_0^{-1})$, we have

$$\texttt{O.Dec}(\texttt{key},c_1)\texttt{O.Dec}(\texttt{key},c_0) \neq \mathbf{1}.$$

On the other hand, for our matrix operation based encryption, we have

$$
\begin{aligned}
c_0 &= \texttt{M.Enc}(K, (\mathbf{m}_0\mathbf{m}_1)\mathbf{m}_2) = K^{-1}A_{\mathbf{m}_2}A_{\mathbf{m}_1}A_{\mathbf{m}_0}K \\
c_1 &= \texttt{M.Enc}(K, \mathbf{m}_2^{-1}(\mathbf{m}_1^{-1}\mathbf{m}_0^{-1})) = K^{-1}A_{\mathbf{m}_0^{-1}}A_{\mathbf{m}_1^{-1}}A_{\mathbf{m}_2^{-1}}K
\end{aligned}
$$

Thus we have

$$\texttt{M.Dec}(K, c_1)\texttt{M.Dec}(K, c_0) = \mathbf{1}.$$

3 Dongxi Liu [3]'s Scheme

Liu [3] proposed a candidate fully homomorphic encryption scheme using linear algebra over $GF(q)$. Though the design in [3] is very complicated, we give a simple (equivalent) description of the protocol in [3]. From the simplified description, it is straightforward that the public evaluation keys leak all of the private key.

Let l, n be given numbers with $l \leq n-2$. It is recommended to use $n = 5$ and $l = 3$ in [3]. The protocol works as follows.

Key Setup.

- Choose random vectors $\mathbf{k} = [k_0, \cdots, k_n] \in GF(q)^{n+1}$ and $\Theta = [\theta_0, \cdots, \theta_{l-1}] \in GF(q)^l$.
- For each $m \in GF(q)$, let $\mathbf{c}_m = \texttt{ENC}(\mathbf{k}, m) = [c_0, \cdots, c_n] \in GF(q)^{n+1}$ such that $m = \mathbf{k} \cdot \mathbf{c}_m$ where $\cdot$ is the inner product of $\mathbf{k}$ and $\mathbf{c}_m$. That is, $\mathbf{k} \cdot \mathbf{c}_m = c_0k_0 + c_1k_1 + \cdots + c_nk_n$.
- Let $\Phi = [\texttt{ENC}(\mathbf{k}, \theta_0), \cdots, \texttt{ENC}(\mathbf{k}, \theta_{l-1}), \texttt{ENC}(\mathbf{k}, 1)]$.
- The private key is $\mathbf{k}$ and Θ.
- The public evaluation key is $\mathbf{pek} = \{\mathbf{p}_{i,j} = \texttt{ENC}(\mathbf{k}, k_ik_j) : 0 \leq i, j \leq n\}$

Encryption. For a message $\mathbf{m} \in GF(q)$, choose random $r_0, \cdots, r_l \in GF(q)$ with $m = r_0 \oplus r_1 \oplus \cdots \oplus r_l$. The ciphertext of m is $\mathbf{c}_m = (r_0 \cdot \texttt{ENC}(\mathbf{k}, \theta_0)) \oplus \cdots \oplus (r_{l-1} \cdot \texttt{ENC}(\mathbf{k}, \theta_{l-1})) \oplus (r_l \cdot \texttt{ENC}(\mathbf{k}, 1))$.

Decryption. For a received ciphertext $\mathbf{c}_m$, compute $m = \mathbf{k} \cdot \mathbf{c}_m$.

Ciphertext addition. The addition of two ciphertexts $\mathbf{c}_{m_0}$ and $\mathbf{c}_{m_1}$ is defined as the component wise addition $\mathbf{c}_{m_0+m_1} = \mathbf{c}_{m_0} + \mathbf{c}_{m_1}$. That is, this is just the regular component wise vector addition.
Ciphertext multiplication. The multiplication of two ciphertexts $\mathbf{c}_{m_0} = [c_0, \cdots, c_n]$ and $\mathbf{c}_{m_1} = [c'_0, \cdots, c'_n]$ is defined as $\mathbf{c}_{m_0 m_1} = \sum_{i,j=0}^{n} c_i c_j \mathbf{p}_{i,j}$.

The correctness of the protocol could be easily verified (for details, it is referred to the original paper [3]. However, the protocol cannot be secure since the private key $\mathbf{k}$ could be trivially derived from the public evaluation key $\mathbf{pek}$. As an example, we can assume that $\mathbf{p}_{i,j} = [p_{i,j,0}, \cdots, p_{i,j,n}]$. Then we have the equations

$$\begin{array}{l} k_0 k_0 = p_{0,0,0} k_0 + \cdots + p_{0,0,n} k_n \\ \cdots \\ k_i k_j = p_{i,j,0} k_0 + \cdots + p_{i,j,n} k_n \\ \cdots \\ k_n k_n = p_{n,n,0} k_0 + \cdots + p_{nn,,n} k_n \end{array} \tag{1}$$

Using equation (1), one can easily obtain the private key $\mathbf{k}$ by constructing polynomial equations $f(k_i) = 0$ in one variable and then using the Euclidean algorithm to compute $\gcd(f(x), x^q - x)$ (or use Berlekamp's algorithm). For example, from the first equation, one can obtain an expression of k_n in terms of $k_0, \cdots, k_{n-1}$. By substituting this k_n into all remaining equations, one eliminates the occurrence of k_n from all remaining equations.

Acknowledgment

I would like to thank Masahiro Yagisawa for some discussion on the protocls in [4].

References

1. John Baez. The octonions. *Bulletin of the American Mathematical Society*, 39(2):145–205, 2002.
2. John H Conway and Derek A Smith. On quaternions and octonions. *AMC*, 10:12, 2003.
3. Dongxi Liu. Practical fully homomorphic encryption without noise reduction. Technical report, Cryptology ePrint Archive, Report 2015/468. http://eprint.iacr.org/2015/468, 2015.
4. Masahiro Yagisawa. Fully homomorphic encryption without bootstrapping. Technical report, Cryptology ePrint Archive, Report 2015/474. http://eprint.iacr.org/2015/474, 2015.