Key-Recovery Attack on a Public-Key Encryption Related to Planted Clique

Caicai Chen[*] Chris Jones[†]

February 28, 2024

Abstract

Hudoba [Hud16] proposed a public key encryption (PKE) scheme and conjectured its security to be based on the Planted Clique problem. In this note, we show that this scheme is not secure. We do so by devising an efficient algorithm for the even neighbor independent set problem proposed by [Hud16]. This leaves open the possibility of building PKE based on Planted Clique.

1 Introduction

The Planted Clique problem is a well-known average-case problem that is suspected to exhibit a statistical/computational gap. In the Planted Clique problem, we are given an n-vertex graph G drawn from one of the following two random models, and the task is to determine which:

1. G is an Erdős-Rényi graph $G \sim \mathcal{G}(n, 1/2)$
2. G is an Erdős-Rényi graph $G \sim \mathcal{G}(n, 1/2)$, then a random subset of k vertices is chosen, and the subset is replaced by a clique. k is a parameter of the problem

On the one hand, with high probability, the size of the largest clique in $\mathcal{G}(n, 1/2)$ is $(2 + o(1)) \log_2 n$, and therefore the two cases are statistically distinguishable once $k \geq (2 + \varepsilon) \log_2 n$. On the other hand, no algorithm is known to detect the presence of the planted clique until $k = \Omega(\sqrt{n})$ [AKS98]. The range of k between $2 \log n$ and $\Omega(\sqrt{n})$ is a conjectured hard regime for the problem, in which the planted clique cannot be identified by any polynomial time algorithm.

Can we use the Planted Clique problem for cryptography? The general idea is to use the planted clique as a secret key for a scheme based on the graph G. Thinking in this direction, Hudoba [Hud16] proposed a public key encryption (PKE) scheme and conjectured

[*]Department of Computing Sciences, Bocconi University. caicai.chen@unibocconi.it
[†]Department of Computing Sciences and BIDSA, Bocconi University. chris.jones@unibocconi.it

its security to be based on Planted Clique. In this note, we observe that the suggested scheme is not secure, and we implement a key recovery attack. This leaves open the possibility of building PKE based on Planted Clique.

On the success side, the Planted Clique problem was suggested as a candidate one-way function by Juels and Peinado [JP00], but that is not known to imply a public key encryption scheme.

Note that the Planted Clique problem for any k can be solved by a quasipolynomial algorithm i.e. time $n^{O(\log n)}$ which brute-force checks for the existence of a clique of size $(2+\varepsilon)\log_2 n$. This is much faster than the exponential or subexponential time hardness typically desired for cryptographic uses. That being said, the Densest k-Subgraph problem is a noisy variant of Planted Clique for which a subexponential time brute-force algorithm is predicted to be optimal [BBH18]. Planted Clique is a simpler starting point for approaching these problems.

Acknowledgements. We are grateful to Alon Rosen for the discussions on this problem. The authors are partly supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreements No. 101019547 and No. 834861).

2 Proposed PKE and Attack

2.1 Key Generation

Let n denote the graph size, let p denote edge probability, let k denote planted clique size, and let p_{add} denote the probability of adding.

The key generation steps ([Hud16, Algorithm 1 in Section 3.1]) are as follows:

1. Choose a random $G \leftarrow \mathcal{G}(n,p)$ graph.
2. Choose a random k sized subset from the nodes of the graph containing the nth vertex. Denote it with $S \subset [n]$.
3. Remove all edges between nodes contained in S: replace E by $E \setminus \{(u,v) \mid u,v \in S\}$ (plant an independent set at the positions corresponding to S).
4. Iterate through $\{u \in V \setminus S : |\Pi_G(u) \cap S| \equiv 1 \pmod 2\}$ in random order
    (a) with p_{add} probability add (u,v) for $v \leftarrow S \setminus \Pi_G(u)$ to E,
    (b) else remove (u,v) for $v \leftarrow \Pi_G(u) \cap S$ from E.

The outcome of step 4 is that every vertex $u \in V$ has an even number of neighbors in S.

2.2 Encryption and Decryption

Following is the encryption method proposed in [Hud16, Section 2.1] originates from [ABW10]:

> **Public key**: G in Section 2.1.
>
> **Private key**: S in Section 2.1.
>
> **Encryption**: Let A be the adjacency matrix of G. Choose a random vector $x \leftarrow \{0,1\}^n$ and a random noise vector $e_i \leftarrow \text{Bernoulli}(\varepsilon), i \in [n]$. Let $b = Ax + e \pmod 2$.
>
> - To encrypt 0, send the vector b.
> - To encrypt 1, send the vector b with its last bit flipped.
>
> **Decryption**: To decrypt $y \in \{0,1\}^n$, output $\sum_{i \in S} y_i \pmod 2$.

2.3 Attack based on computing null space of A

The even neighbor independent set problem is defined as follows:

Definition 2.1 (Even neighbor independent set problem, [Hud16, Definition 6.1])**.**

> ***Input:*** *graph G, positive integer k.*
>
> ***Output:*** *A subset of k nodes $S \subseteq V(G)$ such that*
>
> *(a) S is an independent set, i.e. there are no edges inside S, and*
>
> *(b) Each node outside S has an even number of neighbors in S.*

We show how to solve this problem in a random graph.

Theorem 2.1. *There is an algorithm solving the even neighbor independent set problem that runs in time $n^{O(1)}$ with high probability for the input $G \sim \mathcal{G}(n, 1/2)$.*

Let A be the adjacency matrix of G. The approach is to observe that the indicator vector $\mathbb{1}_S$ satisfies

$$A\mathbb{1}_S \equiv \vec{0} \mod 2$$

The equality holds on S because of the independent set property, and it holds on $V \setminus S$ because of the even neighbor property. Therefore, $\mathbb{1}_S$ lies in the null space of A over $\mathbb{F}_2$.

First, we compute a basis for the null space of A over $\mathbb{F}_2$ using Gaussian elimination. Then, we enumerate all vectors in the null space to search for $\mathbb{1}_S$. The following lemmas bound the runtime for implementing this algorithm.

Lemma 2.1. *For $G \sim \mathcal{G}(n, \frac{1}{2})$ and its adjacency matrix A,* $\dim(\text{null}(A)) = O(\log n)$ *whp.*

Proof. We consider picking the rows of A one at a time. If a row lies in the span of the previous rows, then this will increase the dimension of null(A) by one. The probability of row i lying in the span of rows 1 through $i-1$ is at most $2^{(i-1)-n}$ since the previous rows span a space of dimension at most $i-1$. So we have the upper bound,

$$\dim(\text{null}(A)) \leq \sum_{i=1}^{n} \text{Bernoulli}(2^{i-1-n}).$$

Let $X_i = \text{Bernoulli}(2^{i-1-n})$. We prove $\Pr[\sum_{i=1}^{n} X_i \geq O(\log n)] \leq \frac{1}{n}$ using the Chernoff bound technique i.e. we upper bound the moment generating function. For any $t \geq 0$,

$$\begin{aligned}
&\mathbb{E}[e^{t\sum_{i=1}^{n} X_i}] \\
&= \prod_{i=1}^{n} \mathbb{E}[e^{tX_i}] \\
&= \prod_{i=1}^{n} \left(2^{i-1-n} \cdot e^t + 1 - 2^{i-1-n}\right) \\
&= \prod_{i=1}^{n} (1 + 2^{-i}(e^t - 1)) \\
&= 1 + \left(\sum_{i=1}^{n} 2^{-i}\right)(e^t-1) + \left(\sum_{\substack{i,j=1 \\ i<j}}^{n} 2^{-i}2^{-j}\right)(e^t-1)^2 + \left(\sum_{\substack{i,j,k=1 \\ i<j<k}}^{n} 2^{-i}2^{-j}2^{-k}\right)(e^t-1)^3 + \cdots \\
&\leq 1 + \left(\sum_{i=1}^{\infty} 2^{-i}\right)(e^t-1) + \left(\sum_{i=1}^{\infty} 2^{-i}\right)^2 \frac{(e^t-1)^2}{2!} + \left(\sum_{i=1}^{\infty} 2^{-i}\right)^3 \frac{(e^t-1)^3}{3!} + \cdots \\
&= 1 + (e^t-1) + \frac{(e^t-1)^2}{2!} + \frac{(e^t-1)^3}{3!} + \cdots = e^{e^t-1}.
\end{aligned}$$

The final bound e^{e^t-1} is the moment generating function of a Poisson(1) random variable, so heuristically $\sum_{i=1}^{n} X_i \approx \text{Poisson}(1)$. Using Markov's inequality, for any $L \in \mathbb{R}$,

$$\Pr\left[\sum_{i=1}^{n} X_i \geq L\right] = \Pr\left[e^{t\sum_{i=1}^{n} X_i} \geq e^{tL}\right]$$

$$\leq \mathbb{E}\left[e^{t\sum_{i=1}^{n} X_i}\right] \cdot e^{-tL} \leq e^{e^t-1-tL}$$

Choosing $t = 1$ and $L = O(\log n)$, the tail probability is at most $1/n$. □

Lemma 2.2. *If the dimension of the null space is m, there exist 2^m potential $\mathbb{1}_C$ vectors.*

Consider a graph $G = (V, E)$ defined by the method detailed in Section 2.1. Breaking the scheme requires discovering $\mathbb{1}_S$ with its n-th entry being 1.

An attacker can potentially breach the scheme by executing the following steps:

1. Compute the null space of A, denoting it as null(A).
2. Identify $\mathbb{1}_C \in \text{null}(A)$ and $\mathbb{1}_C[n] = 1$ where $\sum_{i=1}^{n} \mathbb{1}_C[i] = k$. This will recover the indicator of the independent set, which is the secret key.

2.4 Experiments

The experimental framework was designed to operate on randomly generated graph instances. We executed it on different graph sizes, ranging from 1024 to 16384 nodes, each with a planted independent set of size $k = 2\log_2(n)$. For each graph size, the experiment was repeated across 1000 distinct instances. The results show that the dimension of the null space is essentially constant, aligning with Lemma 2.1. The maximum dimension of the null space across all runs was 6. The algorithm successfully recovered all planted independent sets.

We used Julia to implement the attacking algorithm in our experiments. We used the LightGraphs package to handle graph-related operations and the Nemo package for finite field operations.

n	1024	1450	2048	2898	4096	5794	8192	11586	16384
k	20	21	22	23	24	25	26	27	28
Mean	2.322	2.338	2.392	2.342	2.314	2.316	2.332	2.364	2.372
Max	6	6	6	6	6	6	6	6	6

Table 1: Mean and Max of the dimension of nullspace for planted independent set size $k = 2\log_2(n)$ for graph size n, each value is an average of 1000 instances.

References

[ABW10] Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 171–180, 2010. 3

[AKS98] Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a large hidden clique in a random graph. *Random Structures & Algorithms*, 13(3-4):457–466, 1998. 1

[BBH18] Matthew Brennan, Guy Bresler, and Wasim Huleihel. Reducibility and computational lower bounds for problems with planted sparse structure. In *Conference On Learning Theory*, pages 48–166. PMLR, 2018. 2

[Hud16] Péter Hudoba. “public key cryptography based on the clique and learning parity with noise problems for post-quantum cryptography”. In *Proceedings of the 11th*

Joint Conference on Mathematics and Computer Science, pages 102–112, 2016. 1, 2, 3

[JP00] Ari Juels and Marcus Peinado. Hiding cliques for cryptographic security. *Designs, Codes and Cryptography*, 20(3):269–280, 2000. 2