CRYPTANALYSIS AND IMPROVEMENT OF AKLEYLEK ET AL.'S CRYPTOSYSTEM

R. RASTAGHI

Abstract. Akleylek et al. [S. Akleylek, L. Emmungil and U. Nuriyev, A modified algorithm for peer-to-peer security, *journal of Appl. Comput. Math.*, vol. 6(2), pp.258-264, 2007.], introduced a modified public-key encryption scheme with steganographic approach for security in peer-to-peer (P2P) networks. In this cryptosystem, Akleylek et al. attempt to increase security of the P2P networks by mixing ElGamal cryptosystem with knapsack problem. In this paper, we present a ciphertext-only attack against their system to recover message. In addition, we show that for their scheme *completeness* property is not holds, and therefore, the receiver cannot *uniquely* decrypts messages. Furthermore, we also show that this system is not chosen-ciphertext secure, thus the proposed scheme is vulnerable to man-in-the-middle-attack, one of the most pernicious attacks against P2P networks. Therefore, this scheme is not suitable to implement in the P2P networks.
We modify this cryptosystem in order to increase its security and efficiency. Our construction is the efficient CCA2-secure variant of the Akleylek et al.'s encryption scheme in the standard model, the *de facto* security notion for public-key encryption schemes.

1. Introduction

The use of computer network is raised day by day. This increment causes the number of nodes to increase. By increasing the client, the server becomes busy and insufficient although the bandwidths are high enough. Moreover, since the variety of requests is increased, servers may not have data the user needs. We can overcome these obstacles by using peer-to-peer (P2P) network. The P2P networks have become popular as a new paradigm for information exchange and are being used in many applications such as file sharing, distributed computing, video conference, VoIP, radio and TV broadcasting. The P2P networks did not have centralized servers; some powerful nodes act as server. The fourth generation supports streams over P2P networks and each node can talk with another. In these networks, since server has been decentralized and each node can directly communicate with other nodes, management and security become a most important problem. There are several ways to make P2P networks secure. Cryptography plays the most important role in each way. Cryptography is the art of keeping the data secure from eavesdropping and other malicious activities. Therefore, cryptographic algorithms are very essential in the P2P systems since they can uniquely protect message for

2000 *Mathematics Subject Classification.* Primary: 68P25, 94A60, Secondary: 11T71.
Key words and phrases. Cryptography, Cryptanalysis, Ciphertext-only attack, ElGamal cryptosystem, Knapsack problem, CCA2 security, Standard model.

an individual, and verify its integrity.
Due to peer-relying nature of the P2P networks, they are susceptible to many general attacks. Man-in-the-middle attack is one the most pernicious attacks against P2P networks. The man-in-the-middle attack is an indirect intrusion, where the attacker inserts its node undetected between two nodes. It is typically used for eavesdropping a public-key encrypted conversation to retrieve, modify or cut the information being sent by adopting some strategies and tricks. Therefore, the public-key encryption (PKE) scheme must resist against this type of powerful attack. Security against adaptive chosen-ciphertext attack (i.e., CCA2 security) [17] is the strong security notion for a PKE scheme. This notion is known to suffice for many applications of encryption in the presence of active attackers — a man-in-the-middle adversary — including: secure P2P transmission, secure communication, auctions, voting schemes, and many others. In this scenario, the adversary has seen *challenge ciphertext* before having access to the decryption oracle. The adversary is not allowed to ask the decryption of the challenge ciphertext, but can obtain the decryption of any relevant ciphertext *even modified ones based on the challenge ciphertext.* A cryptosystem is CCA2-secure if the cryptanalyst fails to obtain any partial information about the plaintext relevant to the challenge ciphertext. The most cryptographic protocols cannot prevent chosen-ciphertext attacks mounted by a man-in-the-middle adversary who has full control of the communication channel between the sender and the receiver. Indeed, design *efficient* CCA2-secure encryption scheme is a challenging problem in cryptography.

In [2], Akleylek et al. introduced a modified algorithm with steganographic approach for security in the P2P networks. In this cryptosystem, Akleylek et al. attempt to increase security of the P2P system by mixing ElGamal cryptosystem [8] with knapsack problem. The knapsack problem is a decision problem which is NP-complete [11, 12, 13]. That is to say, this problem cannot be easily solved even using quantum computers. They use the ElGamal encryption scheme to disguise private knapsack (easy knapsack) in order to produce public key (hard knapsack). In this paper, we show that this combination leaks the security and makes the cryptosystem vulnerable to ciphertext-only attack. Any encryption scheme vulnerable to this type of attacks is considered to be completely insecure. In addition, we show that in most cases *completeness* property does not holds for their system. Therefore, the receiver cannot *uniquely* decrypts ciphertexts. Besides, their construction is deterministic and so each message has one primage. Therefore, an attacker can simply distinguish between decryptions of the two different messages. Hereupon, this encryption scheme does not satisfies indistinguishability (a.k.a semantic security) against chosen ciphertext attack[1]. Hence, in the network an attacker can apply these attacks and simply can recover plaintext from any challenge ciphertext. Thereupon, this scheme is not suitable for using in a P2P network. We propose a modification to this scheme in order to increase security, efficiency and usability for using in the P2P networks. Our construction is a CCA2-secure PKE scheme in the standard model, the *de facto* security notion for PKE schemes. The main novelty is that scheme's *consistency* check can be directly implemented by the system without having access to some external gap-oracle as in [3, 4] or using other extrinsic rejection techniques [6].

[1] Randomized encryption algorithm is a necessary condition for CCA2 security. Although randomness is necessary, it is not sufficient (see subsection 2.4).

1.1. **Related works.** In 1998, Cai and Cusick [5] proposed an efficient lattice-based public-key cryptosystem with much less data expansion by mixing the Ajtai-Dwork cryptosystem [1] with an additive knapsack. Recently, their cryptosystem was broken by Pan and Deng [16]. They presented an iterative method to recover the message encrypted by the Cai-Cusick cryptosystem under a ciphertext-only scenario. They also present two chosen-ciphertext attacks to get a similar private key which acts as the real private key. In another work, with several known attacks in mind, very recently Pan et al. [15] introduced a new lattice-based PKE scheme mixed with additive knapsack problem which has reasonable key size and quick encryption and decryption. Unfortunately, their scheme was broken by Xu et al. [19]. They proposed two feasible attacks on the cryptosystem of Pan et al.; the first one is a broadcast attack assuming a single encrypted message directed towards for several recipients with different public keys, the message can be recovered by solving a system of nonlinear equations via linearization technique. The second one is a multiple transmission attack in which a single message is encrypted under the same public key for several times using different random vectors. In this situation, the message can be easier to recover. Very recently, Rasatghi [18] introduced an efficient PKE scheme which is robust against man-in-the-middle adversaries for the P2P networks. His scheme uses RSA cryptosystem in combination of the additive knapsack problem. Since RSA encryption scheme is deterministic and therefore does not satisfies CCA2 security requirements, the encryption algorithm uses a new padding scheme for encoding input messages in order to secure mixed scheme against chosen-ciphertext attack.

Organization. The rest of this paper is organized as follows: In the next section, we give some mathematical background and definitions. Akleylek et al.'s cryptosystem will be presented in section 3. Section 4 presents our cryptanalysis and in section 5, we modify this cryptosystem to achieve desired security i.e., CCA2-security and efficiency. Some conclusion is given in section 6.

2. Preliminaries

2.1. **Notation.** We will use standard notation. If x is a string, then $|x|$ denotes its length. If $k \in \mathbb{N}$, then $\{0, 1\}^k$ denote the set of k-bit strings, 1^k denote a string of k ones and $\{0, 1\}^*$ denote the set of bit strings of finite length. $y \leftarrow x$ denotes the assignment to y of the value x. For a set S, $s \leftarrow S$ denote the assignment to s of a uniformly random element of S. For a deterministic algorithm $\mathcal{A}$, we write $x \leftarrow \mathcal{A}^{\mathcal{O}}(y, z)$ to mean that x is assigned the output of running $\mathcal{A}$ on inputs y and z, with access to oracle $\mathcal{O}$. We denote by $\Pr[E]$ the probability that the event E occurs.

2.2. **Mathematical background.**

Definition 2.1 (Subset sum problem [2]). Given a set of positive integers $(a_1, \ldots, a_n)$ and a positive integer s. Whether there is a subset of the a_is such that their sums

[2] Additive knapsack problem.

equal to s. That is equivalent to determine whether there are variables $(x_1, \dots, x_n)$ such that

$$s = \sum_{i=1}^{n} a_i x_i, \quad x_i \in \{0,1\}, \quad 1 \leq i \leq n.$$

The subset sum $(0-1$ knapsack) is a decision problem which is NP-complete. The computational version of the subset sum problem is NP-hard [13]

Definition 2.2 (**Super-increasing sequence**)**.** The sequence $(a_1, \dots, a_n)$ of positive integers is a super increasing sequence if $a_i > \sum_{j=1}^{i-1} a_j$ for all $i \geq 2$.

There is an efficient greedy algorithm to solve the subset sum problem if the b_is are a super-increasing sequence: Just subtract the largest possible value from s and repeat. The following algorithm efficiently solves the subset sum problem for super-increasing sequences in the polynomial time.

Algorithm 1 Solving a super-increasing subset sum problem.

Input: Super-increasing sequence $(a_1, \dots, a_n)$ and an integer s which is the sum of a subset of the a_i.
Output: $(x_1, \dots, x_n)$ where $x_i \in \{0,1\}$, such that $s = \sum_{i=1}^{n} a_i x_i$.

(1) $i \leftarrow n$

(2) While $i \geq 1$ do the following:

  (a) If $s \geq a_i$, then $x_i \leftarrow 1$ and $s \leftarrow s - a_i$. Otherwise $x_i \leftarrow 0$.

  (b) $i \leftarrow i - 1$.

(3) Return $(x_1, \dots, x_n)$.

Definition 2.3 (**Subset product problem**[3])**.** A set of positive integers $(a_1, \dots, a_n)$ and a positive integer d are given. Whether there is a subset of the a_i's such that their product equals to d. That is equivalent to determine whether there are variables $(x_1, \dots, x_n)$ such that

$$d = \prod_{i=1}^{n} a_i^{x_i}, \quad x_i \in \{0,1\}, \quad 1 \leq i \leq n.$$

The multiplicative knapsack (subset product) problem is a decision problem which is NP-complete [11, 12]. As observed in [10, 11, 12, 14], if the a_is are relatively prime, then this problem can be solved in polynomial time by factoring d. Their result can be summarized in the following lemma.

Lemma 2.4. *If $(a_1, a_2, \dots, a_n)$ are relatively prime, then we can solve subset product problem in the polynomial time.*

[3] Multiplicative knapsack problem.

Proof. Since the a_is are relatively prime and $x_i \in \{0,1\}$, so we have

$$x_i = \begin{cases} 1 & \text{if} \quad \mathsf{gcd}(d, a_i) = a_i \\ 0 & \text{if} \quad \mathsf{gcd}(d, a_i) = 1 \end{cases}, \quad 1 \leq i \leq n$$

Hence,

$$x_i = \begin{cases} 1 & \text{if} \quad a_i \mid d \\ 0 & \text{if} \quad a_i \nmid d \end{cases}, \quad 1 \leq i \leq n$$

where gcd means the greatest common divisor. □

Definition 2.5 (**Discrete logarithm problem (DLP)**)**.** Given a prime p, a generator g of $\mathbb{Z}_p^*$, and an element $y \in \mathbb{Z}_p^*$. Find integer $x, 0 \leq x \leq p-2$, such that

$$g^x = y \mod p.$$

is called the discrete logarithm problem.

Fact 2.6. *Suppose that g is a generator of $\mathbb{Z}_p^*$. Then $b = g^i \mod p$ is also a generator of $\mathbb{Z}_p^*$ if and only if* $\gcd(i, p-1)) = 1$.

Definition 2.7. A safe prime p is a prime of the form $p = 2q + 1$ where q is also prime.

2.3. **Definitions.**

Definition 2.8 (**Public-key encryption scheme**)**.** A PKE scheme is a triple of probabilistic polynomial time (PPT) algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ such that:

- Gen is a probabilistic polynomial-time key generation algorithm which takes a security parameter 1^n as input and outputs a public key pk and a secret key sk. We write $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$. The public key specifies the message space $\mathcal{M}$ and the ciphertext space $\mathcal{C}$.
- Enc is a (possibly) probabilistic polynomial-time encryption algorithm which takes as input a public key pk, a $m \in \mathcal{M}$ and random coins r, and outputs a ciphertext $C \in \mathcal{C}$. We write $C \leftarrow \mathsf{Enc}(pk, m; r)$ to indicate explicitly that the random coins r is used and $C \leftarrow \mathsf{Enc}(pk, m)$ if fresh random coins are used.
- Dec is a deterministic polynomial-time decryption algorithm which takes as input a secret-key sk and a ciphertext $C \in \mathcal{C}$, and outputs either a message $m \in \mathcal{M}$ or an error symbol $\perp$. We write $m \leftarrow \mathsf{Dec}(C,\ sk)$.
- Completeness. For any pair of public and secret keys generated by Gen and any message $m \in \mathcal{M}$ it holds that $\mathsf{Dec}(sk,\ \mathsf{Enc}(pk, m; r)) = m$ with overwhelming probability over the randomness used by Gen and the random coins r used by Enc.

Definition 2.9 (**Ciphertext-only attack**)**.** A ciphertext-only attack is a scenario by which the adversary (or cryptanalyst) tries to deduce the decryption key by only observing the ciphertexts or decrypt a *challenge ciphertext.*

Attacker knowledge: some $y_1 = \mathsf{Enc}(x_1, pk)$, $y_2 = \mathsf{Enc}(x_2, pk)$, ...

Attacker goal: obtain $x_1, x_2, \ldots$ or the secret-key sk.

Any encryption scheme vulnerable to this type of attacks is considered to be completely insecure.

Definition 2.10 (CCA2-security). A PKE scheme is secure against adaptive chosen-ciphertext attacks (i.e. IND-CCA2) if the advantage of any two-stage PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ in the following experiment is negligible in the security parameter k:

$$\begin{aligned}
&\mathbf{Exp}^{cca2}_{\mathrm{PKE},\mathcal{A}}(k): \\
&\quad (pk, sk) \leftarrow \mathsf{Gen}(1^k) \\
&\quad (m_0, m_1, \mathsf{state}) \leftarrow \mathcal{A}_1^{\mathsf{Dec}(sk,\cdot)}(pk) \quad \text{s.t.} \quad |m_0| = |m_1| \\
&\quad b \leftarrow \{0,1\} \\
&\quad C^* \leftarrow \mathsf{Enc}(pk, m_b) \\
&\quad b' \leftarrow \mathcal{A}_2^{\mathsf{Dec}(sk,\cdot)}(C^*, \mathsf{state}) \\
&\quad \text{if} \quad b = b^{'} \ \text{ return } 1, \ \text{else return } 0.
\end{aligned}$$

The attacker may query a decryption oracle with a ciphertext C at any point during its execution, with the exception that $\mathcal{A}_2$ is not allowed to query $\mathrm{Dec}(sk, \cdot)$ with *challenge ciphertext* C^*. The decryption oracle returns $b' \leftarrow \mathcal{A}_2^{\mathrm{Dec}(sk,\cdot)}(C^*, \mathsf{state})$. The attacker wins the game if $b = b'$ and the probability of this event is defined as $\Pr[\mathrm{Exp}^{\,cca2}_{\,\mathrm{PKE},\,\mathcal{A}}(k)]$. We define the advantage of $\mathcal{A}$ in the experiment as

$$\mathsf{Adv}^{\mathrm{cca2}}_{\mathrm{PKE},\,\mathcal{A}}(k) = \left| \Pr[\mathrm{Exp}^{\,cca2}_{\,\mathrm{PKE},\,\mathcal{A}}(k) = 1] - \frac{1}{2} \right|.$$

2.4. **ElGamal Cryptosystem.** The ElGamal cryptosystem [8] is a PKE scheme based on discrete logarithm problem (DLP) in $(\mathbb{Z}_p^*, \cdot)$. Let p ba a large prime such that the DLP is infeasible in $(Z_p^*, \cdot)$, and let $g \in \mathbb{Z}_p^*$ be a primitive element. Each user selects a random integer x, $1 \leq x \leq p-2$, and computes $y = g^x \mod p$. (p, g, y) is the public key and x is the secret key.
For encrypts a message, the sender randomly chooses integer $r, 1 \leq r \leq p-2$ and computes $C_1 = g^r$, $C_2 = my^r$ and send $C = (C_1, C_2)$ to the receiver. To recover message m from ciphertext C, the receiver using private key x computes $m = C_2(C_1^x)^{-1} \mod p$.

Altough the ElGamal scheme is randomized, but it not CCA2-secure. An attacker can pick a random number r' and generate the ciphertext $C_1' = g^{r+r'}$, $C_2' = my^{r+r'} = mg^{x(r+r')}$, as the values g and y are known from the public key. The attacker can then query for the decryption of this modified ciphertext and receive the message m as answer.

3. Akleylek et al. Cryptosystem

In this section, we present Akleylek et al. cryptosystem [2]. They wish to increase security of proposed cryptosystem by mixing the ElGamal cryptosystem with multiplicative knapsack problem.

(1) **Key generation**

(a) We choose a super-increasing sequence $A = (a_1, \ldots, a_n)$, such that $a_i > \sum_{i=1}^{j-1} a_i$, $2 \leq j \leq n$, and all a_i's are integer.

(b) The keys of the ElGamal cryptosystem (y, g, p, x) are calculated, where $y = g^x$.

(c) For calculating public knapsack $B = (b_0, \ldots, b_n)$, randomly select an integer k, $1 \leq k \leq p-2$ and compute:

$$\begin{aligned} &y = g^x \mod p, \quad s_i = g^k \mod p, \\ &u_i = y^k.a_i \mod p, \quad b_i = (s_i, u_i) \quad \text{for} \quad 1 \leq i \leq n. \end{aligned}$$

Finally, $B = (b_1, \ldots, b_n) = ((s_1, u_1), \ldots, (s_n, u_n))$ is the public key and $(y, g, p, x, (a_1, \ldots, a_n))$ is the secret key.

(2) **Encryption**
To encrypt n bit binary message $m = (m_1, \ldots, m_n)$, we compute

$$C = (C_1, C_2) = \prod_{i=1}^{n} (s_i, u_i)^{m_i}, \tag{3.1}$$

and send ciphertext C to the receiver.

(3) **Decryption**
To decrypt the ciphertext C, the receiver firstly calculates

$$d = C_2.(C_1^x)^{-1} \mod p = \frac{\prod_{i=1}^{n} u_i^{m_i}}{\prod_{i=1}^{n} (s_i^x)^{m_i}} \mod p = \prod_{i=1}^{n} a_i^{m_i} \mod p. \tag{3.2}$$

After calculating d, we must obtain plaintext $m = (m_1, \ldots, m_n)$ from $d = a_1^{m_1} a_2^{m_2} \ldots a_n^{m_n}$. Note that $u_i = y^k.a_i \mod p = g^{xk}.a_i \mod p = (s_i)^x.a_i \mod p$.

Remark 3.1. *We stress that for the decryption algorithm works, we need to choose prime p such that $p \geq \prod_{i=1}^{n} a_i$, which does not remark on the Akleylek et al.'s original paper. We illustrate this with an example in the next subsection.*

3.1. **On the Completeness of the Akleylek et al.'s Cryptosystem.** The Akleylek et al.'s cryptosystem has some ambiguity. Completeness property for a PKE scheme (Definition 2.3) guarantees that for any message $m \in \mathcal{M}$ it holds that $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m$. In the Akleylek et al.'s cryptosystem, after apply secret key we have $d = C_2.(C_1^x)^{-1} \mod p = \prod_{i=1}^{n} a_i^{m_i} \mod p$[4], where $a_i, 1 \leq i \leq n$ is a supper-increasing sequence. If the Hamming weight of the input message is small, for small a_is we can efficiently retrieve the input messages but for large Hamming weight, d is the product of the large subset of the $a_1, \ldots, a_n$ and therefore it maybe impossible for the receiver to efficiently recovers m_is from d. The main drawback is that the small a_is maybe the divisors of the larger a_is and therefore a ciphertext maybe does not decrypted *uniquely* and has several decryptions. A moment's reflection reveals that if we want any ciphertext decrypts uniquely, the a_is must be *pairwise primes*. Therefore, super-increasing assumption on the a_is

[4]As we mentioned in Remark 3.1, we suppose that $p \geq \prod_{i=1}^{n} a_i$ and therefore $d = \prod_{i=1}^{n} a_i^{m_i} \mod p = \prod_{i=1}^{n} a_i^{m_i}$ and we have no problem for decrypting the input messages. See Example 3.3 for more details.

is not sufficient for completeness of the PKE scheme and their system does not satisfies completeness property. We illustrate our claims with a small example.

Example 3.2. Suppose $(a_1, a_2, a_3, a_4, a_5) = (2, 3, 6, 12, 24)$ be a super-increasing sequence. Let $p = 2579$ and $g = 2$, where g is a generator of $\mathbb{Z}^*_{2579}$. If we randomly choose $k = 348$ and $x = 1500$, then we have:

$$\begin{aligned} y &= g^x \mod p = 2^{1500} \mod 2579 = 862, \\ s_i &= g^k \mod p = 2^{348} \mod 2579 = 104, \\ u_1 &= y^k \times a_1 \mod p = 862^{348} \times 2 \mod 2579 = 2165, \\ u_2 &= 1958, \quad u_3 = 1337, \quad u_4 = 95, \quad u_5 = 190. \end{aligned}$$

Suppose $(m_1, \ldots, m_5) = (0, 1, 0, 0, 1)$ be an input message. For encrypts message m, we compute $C_1 = s_2 \times s_5 = 104^2 = 10816$ and $C_2 = u_2 \times u_5 = 1958 \times 190 = 372020$. For decrypt ciphertext $C = (10816, 372020)$, receiver computes $d = 372020 \times (10816^{1500} \mod 2579)^{-1} \mod 2579 = 372020 \times 2483 \mod 2579 = 72$. Based on the super-increasing sequence $(2, 3, 6, 12, 24)$, we have: $72 = \underbrace{3}_{a_2} \times \underbrace{24}_{a_5} = \underbrace{6}_{a_3} \times \underbrace{12}_{a_4}$ and therefor the input message $(0, 1, 0, 0, 1)$ has two decryptions: itself and $(0, 0, 1, 1, 0)$. Therefor, completeness does not holds for the Akleylek et al.'s encryption scheme.

As we mentioned in Remark 3.1, if $p < \prod_{i=1}^{n} a_i$, then the decryption algorithm does not works properly. In the previous example, since $p > a_2 \times a_5$, we have no problem for decryption of the input message.

Example 3.3. Now, consider input message $m = (0, 1, 1, 1, 1)$. For encrypt message m, one computes $C_1 = s_2 \times s_3 \times s_4 \times s_5 = 104^4 = 116985856$ and $C_2 = u_2 \times u_3 \times u_4 \times u_5 = 47252120300$. For decrypt ciphertext $(116985856, 27107795330)$, receiver computes $d = 27107795330 \times (116985856^{1500} \mod 2579)^{-1} \mod 2579 = 47252120300 \times 1479 \mod 2579 = 26 \neq \underbrace{3}_{a_2} \times \underbrace{6}_{a_3} \times \underbrace{12}_{a_4} \times \underbrace{24}_{a_5}$. It is because $p = 2579 < a_2 \times a_3 \times a_4 \times a_5 = 5184$.

Therefore in such cases, we cannot efficiently retrieve the input messages from the corresponding ciphertexts.

4. Cryptanalysis of the Akleylek et al. Cryptosystem

In this section, we propose our ciphertext-only attack against Akleylek et al.'s cryptosystem to recover message. We also show since encryption algorithm of the system is deterministic, therefore cryptosystem is not chosen-ciphertext secure. As we previously mentioned, randomness is the necessary property for CCA2 security, but it is not sufficient.

4.1. **Ciphertext-only attack.** In this subsection, we show that the Akleylek et al.'s cryptosystem is vulnerable to ciphertext-only attack. In other words, we can obtain message from challenge ciphertext.

Suppose $C = (C_1, C_2)$ be any challenge ciphertext which encrypted with this cryptosystem and we wish to find the corresponding message. From equation 3.1, we have $C = (C_1, C_2) = \prod_{i=1}^{n}(s_i, u_i)^{m_i} = (s_1, u_1)^{m_1}(s_2, u_2)^{m_2} \dots (s_n, u_n)^{m_n}$. We note that the components $s_i = g^k \mod p$ of the public key are constant respect to i and we have

$$C_1 = \prod_{i=1}^{n} s_i^{m_i} = {s_i}^{m_1} \times \dots \times {s_i}^{m_n} = \underbrace{s_i \times \dots \times s_i}_{h-times} = {s_i}^{h}, \tag{4.1}$$

where $h = \sum_{i=1}^{n} m_i$ is the Hamming weight (the number of $m_i = 1$) of the input message $m = (m_1, \dots, m_n)$. From equation 4.1, we can compute the Hamming weight h of the message $m = (m_1, \dots, m_n)$, as the values s_i and C_1 are known. Thus, we know the *number* of the m_is, where $m_i = 1$. From equation 3.1, we have

$$C_2 = \prod_{i=1}^{n} u_i^{m_i} = {u_1}^{m_1} \times \dots \times {u_n}^{m_n},$$

and therefore from C_2, we know the number of the u_is where product of them equal to C_2, but we do not know which of them. For obtaining these u_is, we need to find a h-tuple subset of the $(u_1, \dots, u_n)$ from public key $B = ((*, u_1), \dots, (*, u_n))$ such that product of them equals to C_2. We denote this subset by S. One can chooses h elements of $(u_1, \dots, u_n)$ in $\binom{n}{h}$ ways. Therefore, we need at most $\binom{n}{h}$ operations to find such subset. After obtaining these u_is, we can obtain original message from the following equation

$$m_i = \begin{cases} 1 & \text{if} \quad u_i \in S \\ 0 & \text{if} \quad u_i \notin S \end{cases}, \qquad 1 \le i \le n.$$

PROBABILITY OF SUCCESS: For small n, we can efficiently compute $\binom{n}{h}$. For sufficiently large fixed integer n, we provide an upper bound for $\binom{n}{h}$.

Lemma 4.1. *Suppose that $h = \lambda n$ is an integer in the range $[0, n]$. Then*

$$\binom{n}{\lambda n} \le 2^{nH(\lambda)},$$

where $H(\lambda) = -\lambda \lg \lambda - (1 - \lambda) \lg(1 - \lambda)$ is the binary entropy function and lg *is the binary logarithm.*

Proof. The statement is trivial if $\lambda = 0$ or $\lambda = 1$, so assume that $0 < \lambda < 1$. To prove the upper bound, by the binomial theorem we have

$$\binom{n}{\lambda n} \lambda^{\lambda n}(1-\lambda)^{(1-\lambda)n} \le \sum_{k=0}^{n} \binom{n}{k} \lambda^{k}(1-\lambda)^{(n-k)} \le (\lambda + (1-\lambda))^n = 1.$$

Hence,

$$\binom{n}{\lambda n} \le \lambda^{-\lambda n}(1-\lambda)^{-(1-\lambda)n} = 2^{-\lambda n \lg \lambda} 2^{-(1-\lambda)n \lg(1-\lambda)} = 2^{nH(\lambda)}.$$

□

We show that the number of binary strings of length n with Hamming weight $h = \lambda n$ is bounded by $2^{nH(h/n)}$. Thus, the running time of the proposed attack is $\mathcal{O}(2^{nH(h/n)})$, and depends on the value of h. For small and large h i.e., for small and large λ, $H(\lambda)$ is small and we can efficiently compute $\binom{n}{h}$ for all n. Therefore, if the Hamming weight of the input message is either small or large, we can efficiently break the cryptosystem for all value of n. $H(\cdot)$ takes the maximum its value on $\lambda = 1/2$, where $H(1/2) = 1$. Thus, $\binom{n}{h}$ takes the maximum its value if $h = n/2$ and the running time of the attack is $\mathcal{O}(2^n)$. Therefore, if n chosen enough large and the input message has Hamming weight close to $n/2$, then the proposed ciphertext-only attack seem does not works. But on the other hand, as we stated in subsection 3.1, for large n, h, *completeness* is not holds for the encryption scheme. From equation 3.2, we have $d = \prod_{i=1}^{n} a_i^{m_i}$. From Lemma 2.4 and [10, 11, 12, 14], when the a_is are relatively prime, we can *efficiently* calculate m_is from d. In the Akleylek et al.'s cryptosystem, since the a_is are super-increasing sequence and are not relatively prime, so small a_is are the divisors of the larger a_is. Thus, as we showed in the example 3.2, we cannot uniquely obtain $m_1, \ldots, m_n$ from equation 3.2. Namely, the problem remains NP-complete and we cannot solve it, especially when h, n is large, i.e., d is the product of the large subset of the $(a_1, \ldots, a_n)$.

As a result, for enough large n we have three cases:

(a) Input messages with *small* hamming weight. In theses cases, we can efficiently compute $\binom{n}{h}$ and therefore we can apply proposed ciphertext-only attack in polynomial time.

(b) Input messages with *medium* hamming weight, i.e., h is close to $n/2$. In theses cases, $\binom{n}{h}$ takes the maximum its value and if n chosen enough large, we cannot efficiently compute it. In such cases, the system has ambiguity and completeness does not holds. Therefore, encryption scheme is not usable.

(c) Input messages with *large* hamming weight. In theses cases, we can efficiently compute $\binom{n}{h}$, however, such as previous case, completeness does not holds.

4.2. **Chosen ciphertext security.** As we previously stated in the introduction section, the Akleylek et al.'s PKE scheme is deterministic and therefore does not satisfies CCA2 security conditions. Following definition 2.10, in the CCA2 security experiment, the challenger runs the key generation algorithm and gives the public key pk to the adversary. The adversary chooses two messages m_0, m_1 with $|m_0| = |m_1|$ and gives it to the challenger. The challenger chooses $b \in \{0, 1\}$ at random and encrypts m_b, obtaining the challenge ciphertext $C^* = \mathsf{Enc}_{pk}(m_b)$ and gives it to the adversary. Since the encryption algorithm is deterministic, thus each message has one preimage. Therefore, CCA2 adversary simply can compute encryption of m_0 with public key pk, namely $C = \mathsf{Enc}_{pk}(m_0)$, and then compare it with the challenge ciphertext. If they are equal then $m_b = m_0$, otherwise $m_b = m_1$.

We summarize the results in the following table.

Table 2. Security and Efficiency Analysis of the Akleylek et al.'s Cryptosystem

Input Message	Proposed Attack	Efficiency	Security
With small hamming weight	Ciphertext-only attack	—	Not secure
With medium hamming weight	Ciphertext-only attack	? [1]	$\approx \mathcal{O}(2^n)$
With large hamming weight	Ciphertext-only attack	—	Not secure
Any input message	CCA attack	—	Not secure

1. Completeness does not holds.

5. Modified Cryptosystem

In this section, we propose our modified encryption scheme based on the Akleylek et al.'s construction.

- **Key generation.** On security parameter n, key generator algorithm $\mathsf{Gen}(1^n)$:
  - (a) Randomly chooses n primes p_i and *safe prime* $p = 2q + 1$ such that $p > \prod_{i=1}^{n} p_i$. It is clear that $|p| \gg n$.
  - (b) Randomly chooses integers x, k such that $1 < x, k < p-2$ and $\gcd(k, p-1) = 1$. Computes

$$\begin{aligned} y &= g^x \mod p, \\ s_i &= g^k \mod p, \\ u_i &= y^k . p_i \mod p, \end{aligned}$$

and $b_i = (s_i, u_i)$ for $1 \le i \le n$. Outputs $(n, p, (b_1, \ldots, b_n))$ as the public key and $(y, g, x, k, (p_1, \ldots, p_n))$ as the private key.

Remark 5.1. Note that since $\gcd(k, p-1) = 1$, from fact 2.6, $s_i = g^k \mod p$ also is a generator.

- **Encryption.** On inputs $m \in \mathbb{Z}_p^*$, pk, encryption algorithm Enc:
  - (a) Uniformly chooses n-bit integer $r = (r_1, \ldots, r_n) \in \{0,1\}^n$ with $r \neq 0, 1$ at random and computes $h = \sum_{i=1}^{n} r_i$.
  - (b) If r is even then $r' \leftarrow r + 1$, else $r' \leftarrow r$.
  - (c) Computes

$$C_1 = (C_1', C_1'') = \prod_{i=1}^{n} (s_i, u_i)^{r_i} \mod p \quad \text{and} \quad C_2 = (m + h)^{r'} \mod p, \tag{5.1}$$

and outputs (C_1, C_2).

It is obviously clear that the modified scheme is chosen-plaintext secure. Each message has 2^n corresponding ciphertext, and therefore, the probability of *distinguish* between two message is 2^{-n} which is negligible.

- **Decryption.** In the decryption phase, firstly we recover randomness r' was used for encrypts message m from C_1. Then r' used to recover message m from C_2. It is clear that for correctly recover message m, we must recover exact the same randomness r from C_1. To recover message m from (C_1, C_2), decryption algorithm Dec performs as follows:

(a) Computes

$$\hat{d} = C_1''.(C_1'^{x})^{-1} \mod p = \frac{\prod_{i=1}^{n} u_i^{\hat{r}_i}}{\prod_{i=1}^{n} (s_i^x)^{\hat{r}_i}} \mod p = \prod_{i=1}^{n} p_i^{\hat{r}_i} \mod p.$$

(b) Since $p > \prod_{i=1}^{n} p_i$ and $\hat{r}_i \in \{0,1\}$, hence $\prod_{i=1}^{n} p_i^{\hat{r}_i} \mod p = \prod_{i=1}^{n} p_i^{\hat{r}_i}$ and so we have

$$\hat{d} = \prod_{i=1}^{n} p_i^{\hat{r}_i}.$$

Since $\hat{r}_i \in \{0,1\}$, then $\hat{d}$ is the product of some distinct primes p_i. By Lemma 2.4, we conclude that

$$\hat{r}_i = \begin{cases} 1 & \text{if} \quad p_i \mid d \\ 0 & \text{if} \quad p_i \nmid d \end{cases}, \quad 1 \le i \le n.$$

(c) With retrieved randomness $\hat{r} = (\hat{r}_1, \dots, \hat{r}_n)$ and secret key $(y, k, (p_1, \dots, p_n))$, computes $\hat{h} = \sum_{i=1}^{n} \hat{r}_i$ and checks wether

$$C_1'' \stackrel{?}{=} y^{k\hat{h}} \prod_{i=1}^{n} p_i^{\hat{r}_i} \mod p \tag{5.2}$$

holds (*consistency* checking) and rejects the ciphertext if not. If it holds then $r \leftarrow \hat{r}$ and $h \leftarrow \hat{h}$. Note that $C_1'' = \prod_{i=1}^{n} u_i^{r_i} \mod p = y^{kh} \prod_{i=1}^{n} p_i^{r_i} \mod p$.

(d) If r is even then $r' \leftarrow r + 1$, else $r' \leftarrow r$.

(e) Finds integer $w, 1 \le w \le p-2$ such that $w \cdot r' = 1 \mod p-1$.[5]

(f) Computes $\hat{m} = ((C_2)^w \mod p) - h$.

(g) Checks wether

$$C_2 \stackrel{?}{=} (\hat{m} + h)^{r'} \mod p \tag{5.3}$$

holds (consistency checking) and rejects the ciphertext if not. If it holds then outputs $m = \hat{m}$.

5.1. **Security analysis.**

5.1.1. Provable Security. The basic idea of provable security theory [9] is to reduce the security of a PKE scheme under some attack model to a mathematically hard problem i.e., integer factorization, discrete logarithm problems and NP-complete problem such as knapsack problem. Provable security has been widely accepted as a standard method for analyzing the security of cryptosystems. Such as original Akleylek et al.'s scheme and previous knapsack-based PKE schemes [5, 12, 14, 15], we fail to obtain any security proof. In this subsection we nonetheless recall certain security-related facts for the clarity of this paper.

Proposition 5.2. *If the discrete logarithm problem (DLP) can be computed very efficiently, then the proposed system is not secure.*

[5] Since $|r'| = n \le |p|$, thus $r' < p$. r' is odd and $p - 1 = 2q$ is even and has two divisor $(2, q)$, therefore, $\gcd(r', p-1) = 1$ and r' has multiplicative inverse modulo $p - 1$.

Proof. First note that even the DLP is computable, we cannot compute x, k from $s_i = g^k \mod p$, $y = g^x \mod p$ and $u_i = y^k \cdot p_i \mod p$, since $(y, g, x, k, (p_1, \dots, p_n))$ is secret.
In the modified cryptosystem, we have

$$C_1' = \prod_{i=1}^{n} s_i^{r_i} \mod p = s_i^{\sum_{i=1}^{n} r_i} \mod p = s_i^h \mod p,$$

where $h = \sum_{i=1}^{n} r_i$ and $s_i = g^k \mod p$ is a generator of $\mathbb{Z}_p^*$. If the DLP is computable, then we can determine Hamming weight h from $C_1' = S_i^h \mod p$. According to the discussion in subsection 4.1, then the modified scheme is vulnerable to ciphertext-only attack if n, h are small or h is large. In such cases, the adversary can retrieve randomness r from $C_1 = (C_1', C_1'')$ and then recover m from $C_2 = (m+h)^{r'} \mod p$. Even if the DLP is computable, then the proposed scheme is not completely breaks. The ciphertext-only attack will works for small (n, h) and large h. It cannot not break system for large n with medium Hamming weight. □

Proposition 5.3. *If a certain special knapsack-type problem can be solved very efficiently, then the proposed system is not secure.*

Proof. Given $p, u_1, \dots, u_n$ and a ciphertext $C_1 = (C_1', C_1'')$, we want to find a subset T of $\{1, \dots, n\}$ such that

$$\prod_{i \in T} u_i \mod p = C_1''. \tag{5.4}$$

Then we can immediately recover randomness r from C_1'' and then compute message m from $C_2 = (m+h)^{r'} \mod p$. Finding such a subset T is a kind of knapsack problem. □

Note congruence 5.4 is a disguised version of the easy knapsack-type problem of finding a subset T of $\{1, \dots, n\}$ such that

$$\prod_{i \in T} p_i \mod p = C_1''.(C_1'^x)^{-1} \mod p,$$

which we solve by computing $\gcd((C_1''.(C_1'^x)^{-1} \mod p), p_i)$ for $i = 1, 2, \dots$.

Birthday Attack. If prime p is chosen too small, then from inequality $p > \prod_{i=0}^{n} p_i$, it follows that n is small. Hence p must be sufficiently large to prevent birthday-search through two lists A and B of $2^{n/2}$ elements to find a couple of sets such that:

$$\prod_{i \in A} u_i = (\prod_{i \in B} u_i)^{-1}.C_1'' \mod p.$$

Therefore n must be chosen such that the adversary's running time is significantly smaller than $2^{n/2}$ steps.

5.1.2. CCA2 Security. In this subsection, we show that the modified scheme satisfies CCA2 security. As we showed in subsection 2.4, the ElGamal system is not CCA2-secure. It is because values g, y are public. Unlike the ElGamal system, in the modified system values $(y, g, x, k, (p_1, \dots, p_n))$ are secret and we cannot perform any modification to the (C_1', C_1'') in order to retrieve randomness r. Even if we can perform any modifications to the challenge ciphertext, then the maliciously-formed ciphertexts will be rejected in the scheme's consistency checking step in (5.2). If we can retrieve randomness r, then we can simply recover message m.

Theorem 5.4. *If the mixed ElGamal-Knapsack encryption scheme is secure, then the modified PKE scheme satisfies CCA2 security in the standard model.*

Proof. In the proof of security, we exploit the fact that for a well-formed ciphertext, we can recover the message if we know the *randomness* r that was used to create the ciphertext.
In the CCA2 experiment (Definition 2.10), the challenger runs the key generation algorithm and gives the public key pk to the adversary.
Challenge Ciphertext. The adversary chooses two messages m_0, m_1 with $|m_0| = |m_1|$ and gives it to the challenger. The challenger chooses $b \in \{0,1\}$ at random, randomness r^* and encrypts m_b, obtaining the challenge ciphertext $C^* = (C_1^*, C_2^*)$, where $C_1^* = \prod_{i=1}^{n}(s_i, u_i)^{r_i^*} \mod p$ and $C_2^* = (m_b + h^*)^{r'^*} \mod p$ and gives it to the adversary, where h^* is the Hamming weight of the randomness r^*. We denote by r^* the corresponding intermediate quantity chosen by the challenger.

The challenger has to simulate the decryption oracle. The CCA2 adversary submits a request $C = (C_1, C_2)$ to the challenger, and it outputs decryption of the queried ciphertext to the adversary. He attempts to guess the challenge bit b based on the output of the challenger. In the CCA2 experiment, the adversary is not allowed to ask the decryption of the challenge ciphertext, but can obtain the decryption of any modified ones based on the challenge ciphertext.
To investigate CCA security experiment, we consider two potential cases chosen by the adversary for querying from the challenger. We also show that any modification to the challenge ciphertext does not reveal any useful information about the challenge message m_b.
Case 1: $C_1 = C_1^*$ and $C_2 \neq C_2^*$. In this case, the adversary chooses C_2 at random and queries on ciphertext (C_1^*, C_2). The challenger takes as input (C_1^*, C_2) and computes $r = \mathsf{Dec}_{pk}(C_1^*) = r^*$, $h = h^*$ and $r' = r'^*$. It also computes $\hat{m} = ((C_2)^{\mathsf{Inv}(r'^*)} \mod p) - h^* \neq ((C_2^*)^{\mathsf{Inv}(r'^*)} \mod p) - h^* = m_b$, where $\mathsf{Inv}(r') = (r')^{-1} \mod p-1$ is the multiplicative inverse of r. Since $C_2^* = (m_b + h^*)^{r'^*} \mod p \neq (\hat{m} + h^*)^{r'^*} \mod p$, thus the simulator rejects the ciphertext in (5.3). Therefore, the system does not reveal any information about the challenge message m_b, and so, advantage of the adversary to guess the challenge bit b in this case is zero.
In this case, the adversary cannot perform any modification to C_2 based on C_2^* in order to retrieve m_b, since he does not know the internal random component r^* was chosen by the challenger for encrypts m_b.

Case 2: $C_1 \neq C_1^*$ and $C_2 = C_2^*$. In this case, the adversary chooses C_1 at random and queries on ciphertext (C_1, C_2^*). The challenger takes as input (C_1, C_2^*) and computes $r = \mathsf{Dec}_{pk}(C_1)$. Since encryption algorithm of C_1 is deterministic, therefore any randomness r has *one* preimage. Thus if $C_1 \neq C_1^*$, then $r = \mathsf{Dec}_{pk}(C_1) \neq \mathsf{Dec}_{pk}(C_1^*) = r^*$. In the worst case, we assume r and r^* have the same Hamming weight, namely $h = h^*$. So, we have $\hat{m} = ((C_2^*)^{\mathsf{Inv}(r')} \mod p) - h^* \neq ((C_2^*)^{\mathsf{Inv}(r'^*)} \mod p) - h^* = m_b$. Hence, the simulator rejects the

ciphertext in (5.3), since $C_2^* = (m_b + h^*)^{r'^*} \mod p \neq (\hat{m} + h^*)^{r'} \mod p$. Therefore, the encryption scheme does not reveal any information about the challenge message m_b and so, advantage of the adversary to guess the challenge bit b in this case is zero.
As shown in [7, 18], in the knapsack-based PKE schemes, CCA2 adversary cannot efficiently produces legitimate ciphertext based on C_1^*. As shown in [18], the probability of succeed adversary for retrieve r with *one* bit differ from r'^* is $1/2n$ which is smaller than $1/2$ (note in general, the probability of guessing b is $1/2$; $b = 0$ or $b = 1$). We stress that even if the adversary can computes r with probability greater than $1/2$, then since the retrieved randomness r is not equal to r^* (differ from one bit), therefore $\hat{m} = ((C_2^*)^{\mathsf{Inv}(r')} \mod p) - h^*$ is not equal to m_b, where we assume r and r^* have the same Hamming weight. So, as we state above, the simulator will rejects the ciphertext in (5.3). □

6. Conclusion

In this paper, we consider a knapsack-based PKE scheme mixed with the ElGamal cryptosystem. This cryptosystem uses the ElGamal system in the key generation stage to disguise the secure knapsack (super-increasing sequence) in order to produce the public knapsack. It uses subset product (multiplicative knapsack) problem as encryption function which is NP-complete problem. We showed that this combination leaks the security and makes the cryptosystem vulnerable to ciphertext-only attack. In addition, since encryption algorithm for the mentioned scheme is deterministic, therefore it does not satisfy CCA2 security requirements. Thus, the resulting encryption scheme is also vulnerable to man-in-the-middle attack, and therefore, the scheme is not suitable to implement in a P2P network. Besides, as we showed, completeness property does not holds for the system in the general.
We modified this cryptosystem to improve its security and efficiency. The modified scheme is CCA2-secure and the proposed ciphertext-only attack is not applicable. Completeness holds for all cases and each ciphertext decrypts uniquely.

Acknowledgment

We thank the anonymous referees for insightful comments.

References

[1] M. Ajtai and C. Dwork. A public-key cryptosystem with worstcase/average-case equivalence. In *STOC'97*, ACM Press, (1997), pp. 284-293.
[2] S. Akleylek, L. Emmungil and U. Nuriyev, A modified algorithm for peer-to-peer security, *journal of Appl. Comput. Math.*, vol.6, no.2, (2007), pp. 258-264.
[3] D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM Journal on Computing*, vol. 36(5), (2006), pp. 915-942.
[4] X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. In *ACM CCS'05*, ACM Press, (2005), pp. 320-329.
[5] J. Y. Cai and T. W. Cusick. A lattice-based public-key cryptosystem. In *SAC'98*, LNCS, vol. 1556, (1999), pp. 219-233.
[6] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, vol. 33(1), (2003), pp. 167-226.
[7] T.W. Cusick. A Comparison of RSA and the Naccache-Stern Public-Key Cryptosystem. In *Security Protocols*, LNCS, vol. 1189, (1997), pp 111-116.

[8] T. ElGamal, A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithm, *IEEE Trans. on Information Theory*, vol.31(4), (1985), pp. 469-472.

[9] N. Koblitz and A. Menezes. Another look at "provable security". *Journal of Cryptology*, vol. 20(1) (2007), pp.3-37.

[10] M. K. Lai. Knapsack Cryptosystems: The past and the future. *Available in*: `http://www.ics.uci.edu/~mingl/knapsack.html`.

[11] B.M. Moret. The Theory of Communication, Addison-Wesley, Reading 1998.

[12] R.C. Merkle and M.E. Hellamn, Hiding Information and Signatures in Trapdoor knapsacks, *IEEE Trans. on Information Theory*, vol.24 (1978), pp.525-530.

[13] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, Handbook of Applied Cryptography. CRC-Press 1996.

[14] D. Naccache and J. Stern, A new public-key cryptosystem, *In Advances in Cryptology, EUROCRYPT'97*, LNCS, vol. 1233, (1997), pp.27-36.

[15] Y. Pan, Y. Deng, Y. Jiang,Z. Tu, A New Lattice-Based Public-Key Cryptosystem Mixed with a Knapsack. In *CANS'11*, LNCS, vol. 7092, (2011), pp. 126137.

[16] Y. Pan and Y. Deng. A Ciphertext-Only Attack Against the Cai-Cusick Lattice-Based Public-Key Cryptosystem, *IEEE Trans. on Information Theory*, vol. 57(3), (2011), pp. 1780-1785.

[17] C. Rackoff and D. Simon. Noninteractive Zero-knowledge Proof of Knowledge and Chosen Ciphertext Attack. *In Advances in Cryptology, CRYPTO'91*, LNCS, vol. 576, 1991, pp. 433-444.

[18] R. Rastaghi. An Efficient Encryption Algorithm for P2P Networks Robust Against Man-in-the-Middle Adversary. *IJCSI, International Journal of Computer Science Issues*, vol.9(6), no.1, (2012), pp. 97-102.

[19] J. Xu, L. Hu, S. Sun, P. Wang. Cryptanalysis of a Lattice-Knapsack Mixed Public Key Cryptosystem, In *CANS'12*, LNCS, vol. 7712, (2012), pp. 32-42

Department of Electrical Engineering, Aeronautical University of Sciences & Technology, Tehran, Iran.

E-mail address: `r.rastaghi59@gmail.com`