Security Analysis of Public Key Encryptions Based on Conjugacy Search Problem

Akihiro Yamamura

Security Analysis of Public Key Encryptions Based on Conjugacy Search Problem

Akihiro Yamamura

Department of Mathematical Science and Electrical-Electric-Computer Engineering,
Akita University,
1-1, Tegata Gakuen-machi, Akita 010-8502 Japan

Abstract. We report a fatal flaw of CSP-ElG scheme, one of public key encryptions based on conjugacy search problem proposed in INSCRYPT 2010. It does not satisfy the security property claimed as it is. We also discuss imperfections of security proofs of the other proposals: CSP-hElG and CSP-CS schemes. Following the technique given by Gennaro et al. to smooth a distribution of DH transform outputs, we introduce a computational assumption related to monoid actions and fix the CSP-ElG scheme using a universal hash function and the leftover hash lemma.

Keywords: Conjugacy Search Problem, DDH Assumption, Monoid Action, Universal Hash Functions, Leftover Hash Lemma

1 Introduction

Three generic designs, CSP-ElG, CSP-hElG and CSP-CS schemes, to construct a public key encryption using a conjugacy search problem are proposed by L.Wang, L.Wang, Z.Cao, E.Okamoto and J.Shao in Inscrypt 2010 [9]. The proposed schemes can be instantiated using any conjugacy search problem (CSP for short), which is used in [6, 8]. Each scheme is claimed to have desired provable security provided that a computational assumption related to CSP holds. Their motivation of these proposals is to invent new schemes based on principles other than the ones suffering from attacks using quantum algorithms such as Shor's factoring or Glover's database search algorithm because most of the widely used schemes like RSA and ElGamal would confront such attacks if a quantum computer could be realized.

In this paper, we report fatal flaws in these schemes, in particular, we analyze the CSP-ElG scheme in detail. We also briefly discuss the security of the CSP-hElG and CSP-CS schemes and give circumstantial evidences that these schemes do not enjoy the claimed security property. In addition, we shall fix the CSP-ElG using the Gennaro, Krawczyk and Rabin's technique to smooth the distribution of outputs of DH transform over non-DDH group [5] using the leftover hash lemma [2, 7]. For this purpose we show that both ElGamal and CSP-ElG are an instantiation of a generic scheme based on monoid actions. Then we prove such a generic scheme is indistinguishable against chosen plaintext attacks in the standard model under a reasonable computational assumption, that is, t-MA-DDH assumption.

2 Flaws of Schemes Based on Conjugacy Search Problem

Let M be a (not necessarily commutative) monoid. Recall that a *monoid* is an algebraic system with an associative multiplication and the identity element "1". We denote the set of invertible elements x of M by $G(M)$, that is, $G(M) = \{x \mid \exists\, y \in M \text{ such that } xy = yx = 1\}$. The inverse of $x \in G(M)$ is the element y such that $xy = yx = 1$. Note that the inverse of x is uniquely determined. The *conjugacy search problem* is to find an element $g \in G(M)$ such that $f = gdg^{-1}$ for given $d, f \in M$ provided that such an element g exists.

Suppose that $d \in M$ and $g \in G(M)$ and the order of g is n. If the order of g is infinite, then n is specified to be a large enough. The *CSP-CDH problem* is to compute $g^{a+b}dg^{-(a+b)}$ for given $d \in M$, $g \in G(M)$, g^adg^{-a} and g^bdg^{-b}, where a and b are ramdomly and uniformly chosen from $\{1, \ldots, n\}$. We say that the *CSP-CDH assumption* holds for M if there is no efficient algorithm to answer correctly to a CSP-CDH problem instance.

The*CSP-DDH problem* is a decisional variant of the CSP-CDH problem, that is, it is to decide whether or not $f = g^{a+b}dg^{-(a+b)}$ for given $d \in M$, $g \in G(M)$, g^adg^{-a}, g^bdg^{-b} and $f = g^cdg^{-c}$, where a and b and are randomly chosen from $\{1, \ldots, n\}$ and either c is randomly chosen from $\{1, \ldots, n\}$ or $c = a + b$ with probability $\frac{1}{2}$. We say that the *CSP-DDH assumption* holds for M if there is no efficient algorithm to answer correctly to a CSP-DDH problem instance with probability non-negligibly larger than $\frac{1}{2}$.

2.1 CPS-ElG scheme

The CSP-ElG scheme is defined as follows. Let $K = \{g^adg^{-a} \mid 1 \leq a \leq \mathrm{Ord}(g)\}$, where $\mathrm{Ord}(g)$ stands for the order of the element g. Let P be the message space $\{0,1\}^k$, C the ciphertext space $K \times P$. Suppose $H : K \to P$ is a cryptographic hash function. Alice picks a $(1 \leq a \leq \mathrm{Ord}(g))$ and publicizes g^adg^{-a}. Bob picks b $(1 \leq b \leq \mathrm{Ord}(g))$ and encrypts a message $m \in P$ by

$$c = (g^bdg^{-b}, m \oplus H(g^b(g^adg^{-a})g^{-b})).$$

Receiving the ciphertext $c = (c_1, c_2)$, Alice decrypts it by $m = c_2 \oplus H(g^ac_1g^{-a})$.

Theorem 1 of [9] claims that the CSP-ElG scheme is indistinguishable against chosen plaintext attacks in the standard model under the CSP-DDH assumption. We note that if the monoid is instantiated by a braid group then the CSP-ElG scheme is exactly identical with the public key encryption proposed by Ko et al. [8], in which the function H is assumed to be an ideal hash function. The authors did not clearly describe what they mean by an "ideal hash function" and no precise security analysis of the scheme is given in [8].

We must not assume an random oracle in the standard model, and so H is not allowed to be a random oracle. We shall see that if H is a random oracle, the scheme is indistinguishable against chosen plaintext attacks and so a random oracle is vital in the CSP-ElG scheme and this disproves Theorem 1 of [9].

CSP-ElG is not indistinguishable in the standard model. We choose two messages m_1 and m_2 from P. One of them is chosen by coin toss and it is encrypted as c then we are asked to decide whether c is a ciphertext of m_1 or m_2. First, we define a cryptographic hash function H to be

$$H(m) = \text{SHA–1}(m)|0. \tag{1}$$

The value of H is the concatenation of the value of SHA-1(m) and a bit 0. Then H is a cryptographic hash function of hash size 161 bits and satisfies collision resistance, pre-image and second pre-image resistance, while it is not a random oracle because the last bit is always 0 and so the hash value is not random.

Let $P = \{0,1\}^{161}$. Take m_1 as any message with the last bit is 1, and m_2 as any message with the last bit is 0. Then the ciphertext of m_1 is given by

$$c = (g^b d g^{-b}, m_1 \oplus H(g^b(g^a d g^{-a})g^{-b})).$$

The last bit of the second entry is always 1 since the last bit of $H(g^b(g^a dg^{-a})g^{-b})$ is 0. The ciphertext of m_2 is

$$c = (g^b d g^{-b}, m_2 \oplus H(g^b(g^a d g^{-a})g^{-b})).$$

Similarly the last bit of the second entry is always 0 since the last bit of $H(g^b(g^a dg^{-a})g^{-b})$ is 0. Therefore an attacker can always distinguish the ciphertexts of m_1 and m_2 with probability 1. This shows that the CSP-ElG scheme is not indistinguishable in the standard model and disproves Theorem 1 of [9].

Error in Security Proof of CSP-ElG Scheme. We analyze the proof of Theorem 1 given in Appendix B of [9], where the authors assume an efficient adversary $\mathcal{A}$ that can distinguish ciphertexts of two distinct messages m_0 and m_1 and then construct an algorithm $\mathcal{B}$ for the CSP-DDH problem, which contradicts the CSP-DDH assumption. We may assume without loss of generality that the last bit of m_0 and m_1 are 0 and 1, respectively.

Given a CSP-DDH instance $Z = (d, g, g^a dg^{-a}, g^b dg^{-b}, g^c dg^{-c})$, $\mathcal{B}$ chooses randomly v and sets $g^{a+v} dg^{-(a+v)} = g^v(g^a dg^{-a})g^{-v}$ as a public key. $\mathcal{A}$ chooses two distinct messages m_0 and m_1. Receiving m_0 and m_1, $\mathcal{B}$ chooses randomly w and $\beta \in \{0,1\}$ and computes a ciphertext c^*_β by

$$c^*_\beta = (g^{b+w} dg^{-(b+w)}, m_\beta \oplus H(g^{c+v+w} dg^{-(c+v+w)})).$$

If $c = a + b$, then we have $c + v + w = (a + v) + (b + w)$. In this case c^*_β is a legitimate ciphertext of m_β since

$$c^*_\beta = (g^{b+w} dg^{-(b+w)}, m_\beta \oplus H(g^{b+w}(g^{a+v} dg^{-(a+v)})g^{-(b+w)})),$$

and so, $\mathcal{A}$ can answer correctly β with probability non-negligibly larger than $\frac{1}{2}$.

On the other hand, the authors of [9] claim that if c is chosen randomly, the distribution c^*_0 is identical to that of c^*_1 over all possible random choices of

v and w. In this case c^*_β is an illegitimate ciphertext and so $\mathcal{A}$ cannot behave differently depending on $\beta = 0$ or 1 to the input c^*_β. Repeating these tests, $\mathcal{B}$ can decide whether Z is a CSP-DDH instance or not because behavior of $\mathcal{A}$ is different according to whether $c = a + b$ or not. This contradicts to the CSP-DDH assumption. Then the authors conclude that the CSP-ElG scheme is indistinguishable in the standard model under the CSP-DDH assumption.

There is a pitfall in their discussion. One of the building blocks in the CSP-ElG scheme is a cryptographic hash function H. Since Theorem 1 of [9] is based in the standard model, H may not be a random oracle. Let us instantiate the CSP-ElG scheme by the hash function H defined in (1) which is intensionally designed to have correlation among output bits. The authors claim that if c is chosen randomly, the distribution c^*_0 is identical to that of c^*_1 over all possible random choices of v and w. This claim is incorrect as we explain next. Whatever c is chosen, the last bit of the second entry of c^*_0 and c^*_1 are 0 and 1, respectively, by the definitions of m_1, m_2 and H. It follows that the distribution c^*_0 is completely different from that of c^*_1. As a matter of fact, we can construct the adversary $\mathcal{A}$ as follows. $\mathcal{A}$ chooses two messages m_1 of all 0 and m_2 of all 1. Given c^*_β, $\mathcal{A}$ outputs the last bit of the second entry of c^*_β. Then the algorithm $\mathcal{A}$ always correctly distinguishes the cipher texts of m_0 and m_1.

Our argument clarifies that the security of the CSP-ElG greatly depends on the randomness of hash values; hash values make the distribution of c^*_0 indistinguishable fromthat of c^*_1. If H is a random oracle, the distributions c^*_0 and c^*_1 are indistinguishable. Thus, the proof of Theorem 1 of [9] is correct "in the random oracle model."

The CSP-ElG is different from ElGamal and its security requires a random oracle. The bits extracted from the underlying CSP problem do not necessarily have sufficient randomness and do not match plaintexts in size. The length of the element $g^a(g^b d g^{-b})g^{-a}$ is not equal to the length of a plaintext. The random oracle solves these two issues.

2.2 CSP-hElG and CSP-CS schemes

The CSP-hElG and CSP-CS schemes are the other proposals. Theorem 4 of [9] claims the CSP-CS enjoys IND-CCA in the standard model using a target collision hash function H and a secure symmetric cipher Π under the CSP-DDH assumption. The CSP-CS scheme is a CSP-based variant of a Cramer-Shoup like encryption in [4]. The authors give no proof but claim that the proof of Theorem 4 in [9] is similar to the one of Theorem 13 of [4]. Surprisingly, Theorem 13 does not exist in [4]. We strongly believe Theorem 4 in [9] is incorrect. As a circumstantial evidence, we remark that the hashed DDH assumption, which is a computational assumption of hash values of Diffie-Hellman transforms, is required for the variant of Cramer-Shoup encryption given in [4], whereas no similar assumption is required for the CSP-CS scheme. On the other hand, Theorem 2 of [9] claims that the CSP-hElG scheme enjoys IND-CCA in the random oracle model. No proof is given but the authors claim that a proof is similar to that of Theorem 2 of [1] in which the proof is omitted. In the ElGamal scheme,

a DH output is multiplied with a plaintext, on the other hand, the CSP-DH output must be filtered by a hash function and so we also suspect the security of the CSP-hElG scheme even though we have not completely analyzed.

3 Gennaro, Krawczyk and Rabin's Method

We recall Gennaro, Krawczyk and Rabin's method to obtain a uniform distribution over the set $\{0,1\}^s$ bit strings of length s from DH transform over non-DDH groups to fix the CSP-ElG scheme. The ElGamal encryption is indistinguishable against chosen plaintext attacks provided a generator g is chosen adequately and the base group enjoys the DDH assumption. However, g may be chosen inadequately and its order may be insufficient in length in real-life systems. For example, SSH and IPSec standards instantiate groups in which the DDH assumption does not necessarily hold. Even in such a case, the ElGamal scheme still enjoys provable security under the so-called t-DDH assumption introduced in [5].

We recall necessary terminology. Let $\mathcal{X}$ and $\mathcal{Y}$ be random variables with support contained in $\{0,1\}^n$. The *statistical distance* between $\mathcal{X}$ and $\mathcal{Y}$ is

$$\mathrm{dist}(\mathcal{X},\mathcal{Y}) = \frac{1}{2}\sum_{x\in\{0,1\}^n} |\mathrm{Prob}(\mathcal{X}=x) - \mathrm{Prob}(\mathcal{Y}=x)|.$$

Now suppose $\mathcal{X}_n$ and $\mathcal{Y}_n$ are probability ensembles. Let $\mathcal{D} = \{D_n\}$ be a family of circuits. Then $\mathcal{X}_n$ and $\mathcal{Y}_n$ are called *computationally indistinguishable* (by non-uniform distinguishers) if for every polynomial-size distinguisher family $\mathcal{D}$, for every polynomial $P(\,\cdot\,)$ and for sufficiently large n we have

$$|\mathrm{Prob}_{x\in\mathcal{X}_n}(D_n(x)=1) - \mathrm{Prob}_{y\in\mathcal{Y}_n}(D_n(y)=1)| \le \frac{1}{P(n)}.$$

Let $\mathcal{X}_n$ be a probability ensemble over A_n. The *min-entropy* of $\mathcal{X}_n$ is defined to be

$$\mathsf{min\text{–}ent}(\mathcal{X}_n) = \mathsf{min}_{x\in A_n:\mathrm{Prob}_{x\in\mathcal{X}_n}(x)\neq 0}(-\log(\mathrm{Prob}_{x\in\mathcal{X}_n}(x))).$$

Let $\mathcal{G} = \{G_n\}$ be a family of cyclic groups. We say that $t(n)$-*DDH assumption* holds over $\mathcal{G}$ if for all n there exists a family of probability distributions $\mathcal{X}_n(x^a, x^b)$ such that

1. $\mathsf{min\text{-}ent}(\mathcal{X}_n(x^a,x^b)) \ge t(n)$
2. The probability ensemble

$$\mathcal{DH}_n = \{(x^a, x^b, x^{ab} \mid a,b \in_U \{1,\ldots,\mathrm{Ord}(G_n)\}\}$$

is computationally indistinguishable from the ensemble

$$\mathcal{R}_n^* = \{(x^a, x^b, C \mid a,b \in_U \{1,\ldots,\mathrm{Ord}(G_n)\} \,\mathrm{and}\, C \in_{\mathcal{X}_n(x^a,x^b)} G_n\},$$

where $\mathrm{Ord}(G_n)$ stands for the order of the group G_n.

The notation $x \in_{\mathcal{D}} A$ is to be read as x is chosen from A according to the distribution $\mathcal{D}$, and $x \in_U S$ means choosing x uniformly from the set S. The probability distributions $\mathcal{X}_n(x^a, x^b)$ may be different for each triple x, x^a, x^b. Intuitive meaning of the assumption is that a DH output x^{ab} has some degree of unpredictability.

We also recall a universal hash function introduced in [3]. Suppose $h : \{0,1\}^n \times \{0,1\}^{l(n)} \to \{0,1\}^{m(n)}$ is a function. For each fixed $Y \in \{0,1\}^{l(n)}$ we have a function $h_Y(\,\cdot\,) = h(\,\cdot\,, Y)$ that maps n bits to $m(n)$ bits. Then h is called a *(pairwise independent) universal hash function* if for all $x_1, x_2 \in \{0,1\}^n$ $(x_1 \neq x_2)$ and for all $a_1, a_2 \in \{0,1\}^{m(n)}$, we have

$$\mathrm{Prob}_{Y \in_U \{0,1\}^{l(n)}}(h_Y(x_1) = a_1 \text{ and } h_Y(x_2) = a_2) = \frac{1}{2^{2m(n)}}.$$

Leftover hash lemma is introduced and used to construct pseudorandom bit strings in [7]. It is also used to smooth distributions in [5]. See also [2] for a recent development of the leftover hash lemma.

Lemma 1 (Leftover hash lemma [7]). *Let $\mathcal{X}_n$ be a probability ensemble such that $\mathsf{min\text{–}ent}(\mathcal{X}_n) = m(n)$. Let $e(n)$ be a positive integer valued parameter. Let $h : \{0,1\}^n \times \{0,1\}^{l(n)} \to \{0,1\}^{m(n)-2e(n)}$ be a universal hash function. Let $X \in_{\mathcal{X}_n} \{0,1\}^n$, $Y \in_U \{0,1\}^{l(n)}$ and $Z \in_U \{0,1\}^{m(n)-2e(n)}$. Then we have*

$$\mathrm{dist}(\langle h_Y(X), Y\rangle, \langle Z, Y\rangle) \leq \frac{1}{e(n)+1},$$

where $\langle X, Y\rangle$ stands for the concatenation of X and Y.

Using the leftover hash lemma, Gennaro et al. [5] show that if $\mathcal{G} = \{G_n\}_n$ is a group family in which the $t(n)$-DDH assumption holds and $h : \{0,1\}^{|G_n|} \times \{0,1\}^{l(n)} \to \{0,1\}^{t'(n)}$ is a universal hash function, where $t'(n) = t(n) - \omega(\log n)$, then the induced distribution of $h(g_n^{ab}, Y)$ for $a, b \in_U \{1, 2, \ldots, \mathrm{Ord}(G_n)\}$ and $Y \in_U \{0,1\}^{l(n)}$ is computationally indistinguishable from the uniform distribution over $\{0,1\}^{t'(n)}$ even when h, g_n^a and g_n^b are given to the distinguisher. This implies that the ElGamal scheme using the hashed value $h(g_n^{ab}, Y)$ instead of g_n^{ab} to mask a plaintext is indistinguishable if the underlying group satisfies $t(n)$-DDH assumption. In this case the universal hash function is common knowledge between Alice and Bob and $Y (\in \{0,1\}^{l(n)})$ is a piece of a ciphertext.

4 Encryption Scheme Based on Monoid Action

Before fixing the CSP-ElG, we integrate the ElGamal and CSP-ElG encryption in a generic scheme using the terminology of monoid actions. Then we apply the Gennaro et al.'s technique to smooth distributions to prove the indistinguishability of the generic scheme. A generic scheme covers more instantiations based on many other principles and its proof gives a universal security proof. In particular, the security proof of the generic scheme is applicable to the revised CSP-ElG.

4.1 Monoid action

We first define a *monoid action*. Let M be a monoid, and X be a nonempty set. The symbol "1" stands for the identity of the monoid M, that is, $m \cdot 1 = 1 \cdot m = m$ for every $m \in M$. Suppose that we have a mapping $\sigma : X \times M \to X$ satisfying:

1. $\sigma(\sigma(x, a), b) = \sigma(x, ab)$ for $x \in X$ and $a, b \in M$,
2. $\sigma(x, 1) = x$ for $x \in X$.

Then we say that σ is an *action* of M on X or M *acts on* X without mentioning σ if the context is clear. A group action is a special case of monoid actions.

A monoid action is frequently used in cryptology (see [10, 11]). For example, the mechanism of the Diffie-Hellman key exchange and the ElGamal encryption are explained in terms of the monoid action. Let p be a prime. Suppose a prime q divides $p - 1$. Take an element $g \in \mathbb{Z}_p^*$ such that $\mathrm{Ord}(g) = q$. We should note that $(\mathbb{Z}_q, \cdot)$ forms a monoid but not a group. Let $X = \langle g \rangle$ and $M = (\mathbb{Z}_q, \cdot)$. Then the mapping $\sigma : X \times M \to X$ given by $\sigma(g^i, s) = (g^i)^s = g^{is}$ is a monoid action of $(\mathbb{Z}_q, \cdot)$ on the cyclic group $\langle g \rangle$. The discrete logarithm problem is to compute s for given g and $\sigma(g, s)(= g^s)$. The Diffie-Hellman problem is to compute $\sigma(g, ab)(= g^{ab})$ for given $g, \sigma(g, a)(= g^a)$ and $\sigma(g, b)(= g^b)$, where $a, b \in (\mathbb{Z}_q, \cdot)$. These computational problems related to the monoid action play vital role in the security of the Diffie-Hellman key exchange and the ElGamal encryption. Furthermore, the RSA encryption is also explained in terms of monoid actions although we do not explain here due to lack of space (see [11]).

4.2 Computational assumption related to monoid actions

Suppose $\sigma : X \times M \to X$ is an action of commutative monoid M on a nonempty set X. A *monoid action search problem* (MA-SP for short) is to compute a for given $x, \sigma(x, a)$. A *monoid action computation Diffie-Hellman problem* (MA-CDH for short) is to compute $\sigma(x, ab)$ for given $x, \sigma(x, a)$ and $\sigma(x, b)$. A *monoid action decision Diffie-Hellman problem* (MA-DDH for short) is to decide whether or not $\sigma(x, ab) = \sigma(x, c)$ for given $x, \sigma(x, a), \sigma(x, b), \sigma(x, c)$.

The CSP related problems can also be characterized in terms of monoid actions. Suppose X is a monoid and $G(X)$ is the group of units of X. Suppose that $a \in G(X)$ and $\mathrm{Ord}(a) = n$ and if it is infinite then we set n large enough integer. An action $\sigma : X \times \langle a \rangle \to X$ is given by $\sigma(b, a^i) = a^i b a^{-i}$, where $b \in X$. We note that $M(= \langle a \rangle)$ is indeed a commutative group whereas the base monoid X is not necessarily commutative. Obviously, the conjugacy search, CSP-CDH and CSP-DDH problems are an instance of MA-SP, MA-CDH and MA-DDH problems, respectively. We should also note that the discrete logarithm, CDH and DDH problems are an instance of MA-SP, MA-CDH and MA-DDH, respectively.

We now give formal definition of the MA-CDH and MA-DDH. Suppose $\sigma_n : X_n \times M_n \to X_n$ is a family of actions of commutative monoid M_n on nonempty set X_n. Let us consider the following two ensembles

$$\mathcal{R}_n = \{(x, \sigma_n(x, a), \sigma_n(x, b), \sigma_n(x, c) \mid x \in X_n, a, b, c \in_U M_n\},$$

$$\mathcal{MA}\text{–}\mathcal{DH}_n = \{(x, \sigma_n(x,a), \sigma_n(x,b), \sigma_n(x,ab) \mid x \in X_n, a, b \in_U M_n\}.$$

We say that the *MA-DDH assumption* holds over σ_n if $\mathcal{R}_n$ and $\mathcal{MA}\text{–}\mathcal{DH}_n$ are computationally indistinguishable (with respect to non-uniform distinguishers).

Following [5], we introduce a $t(n)$-MA-DDH assumption. We say that $t(n)$-*MA-DDH assumption* holds over $\sigma_n : X_n \times M_n \to X_n$ if for all n there exists a family of probability distributions $\mathcal{X}_n(x, \sigma(x,a), \sigma(x,b))$ over X_n such that

1. $\mathsf{min\text{-}ent}(\mathcal{X}_n(x, \sigma(x,a), \sigma(x,b))) \geq t(n)$
2. The probability ensemble $\mathcal{MA}\text{–}\mathcal{DH}_n$ is computationally indistinguishable from the ensemble

$$\mathcal{R}^*_n = \{(x, \sigma(x,a), \sigma(x,b), C \mid a, b \in_U M_n \text{ and } C \in_{\mathcal{X}_n(x,\sigma(x,a),\sigma(x,b))} \sigma(x, M_n)\}.$$

The probability distributions $\mathcal{X}_n(x, \sigma(x,a), \sigma(x,b))$ may be different for each triple $x, \sigma(x,a), \sigma(x,b)$. Intuitive meaning of the assumption above is that a MA-DH output $\sigma(x,ab)$ has some degree of unpredictability.

4.3 Public key encryption based on monoid actions

Suppose $\sigma : X \times M \to X$ is an action of a commutative monoid M on a set X. Alice chooses $a \in M$ and Bob chooses $b \in M$. An element $x \in X$ is chosen and fixed and publicized. Alice sends $\sigma(x,a)$ to Bob, and Bob sends $\sigma(x,b)$ to Alice. Then $\sigma(x,ab)(= \sigma(x,ba))$ turns out to be a shared key between Alice and Bob. Recall that we assume M is commutative. If the shared key is indistinguishable from the uniform distribution over $\{0,1\}^l$, where l is the size of representation of elements of X, then it can be used to mask a plaintext P of length l, where the ciphertext C is given by $P \oplus \sigma(x,ab)$. In the case of ElGamal scheme, the length of a plaintext is equal to that of a DH transform output and so it is unnecessary to operate a hash function if the DDH assumption holds.

In the case of the CSP based scheme, the length of a plaintext is not necessarily equal to that of the shared bit string $\sigma(x,ab)$ and so we need to match the length of plaintexts and that of the bit strings extracted from $\sigma(x,ab)$ and make it uniform distribution. We expect the resulting bit sequences $\sigma(x,ab)$ to have randomness to some extent. The MA-DDH assumption implies that the MA-DH transform outputs $\sigma(x,ab)$ distribute uniformly over the set $\sigma(x,M)$.

On the other hand, we should note that the range $\sigma(x,M)$ does not necessarily form the set $\{0,1\}^{l(n)}$ of bit strings of fixed length $l(n)$. This is indeed the same case as the ElGamal scheme. We would like to obtain uniform distribution over the set $\{0,1\}^{l(n)}$ of bit strings of some fixed length $l(n)$ using a MA-DH transform. In particular, if we apply it to encryption scheme by masking a plaintext of length n, we require the ensemble $\mathcal{X}$ of masking sequences to have $\mathsf{min\text{–}ent}(\mathcal{X}) = n$.

Applying the leftover hash lemma to MA-DH outputs from a family of monoid actions in which $t(n)$-MA-DDH assumption holds, we obtain the next theorem.

Theorem 1. *Let $\mathcal{S} = \{\sigma_n : X_n \times M_n \to X_n\}$ be a family of monoid actions in which the $t(n)$-MA-DDH assumption holds, and $h : \{0,1\}^{|M_n|} \times \{0,1\}^{l(n)} \to$*

$\{0,1\}^{t'(n)}$ *be a universal hash function;* h_Y *maps* $\{0,1\}^{|M_n|}$ *into* $\{0,1\}^{t'(n)}$*, where* $Y \in \{0,1\}^{l(n)}$ *and* $t'(n) = t(n) - \omega(\log n)$*. Then the distribution of* $h_Y(\sigma(m, ab))$ *for* $a, b \in_U M_n$ *and* $Y \in_U \{0,1\}^{l(n)}$ *is computationally indistinguishable from the uniform distribution over* $\{0,1\}^{t'(n)}$*.*

Revised CSP-ElG scheme We revise the CSP-ElG scheme as follows. Suppose $t(n)$-CSP-DDH assumption (one of a concrete instance of the $t(n)$-MA-DDH assumption) holds for M, $d \in M$, $g \in G(M)$ and $h : \{0,1\}^{|M_n|} \times \{0,1\}^{l(n)} \to \{0,1\}^{t'(n)}$ is a universal hash function. A public key is a pair $(g, g^a dg^{-a})$. A plaintext $P \in \{0,1\}^{t'(n)}$ is encrypted as

$$(Y, g^b dg^{-b}, P \oplus h(Y, g^{a+b} dg^{-(a+b)})),$$

where $Y \in_U \{0,1\}^{l(n)}$. In this case, the universal hash function h is publicized and $Y (\in_U \{0,1\}^{l(n)})$ is a piece of a ciphertext. Decryption is obvious and we omit it.

Theorem 2. *The revised CSP-ElG scheme is indistinguishable against chosen plaintext attacks in the standard model under the* $t(n)$*-CSP-DDH assumption.*

Proof. One can prove the indistinguishability along the same line as the argument in Section 2.1. We replace the hash function H by a universal hash function h_Y ($Y \in_U \{0,1\}^{l(n)}$) here. We have to discuss the case that c is chosen randomly. In this case, we have $c^*_\beta = (g^{b+w} dg^{-(b+w)}, m_\beta \oplus h_Y(g^{c+v+w} dg^{-(c+v+w)}))$ Because c and $v + w$ are randomly chosen, the hash value $h_Y(g^{c+(v+w)} dg^{-(c+(v+w))})$ is uniformly distributed over $\{0,1\}^{t'(n)}$ by Theorem 1. It follows that both distributions c^*_0 and c^*_1 are computationally indistinguishable from the uniform distribution over $\{0,1\}^{t'(n)}$ and that c^*_0 and c^*_1 are computationally indistinguishable. Therefore, the original proof of Theorem 1 in [9] shows the indistinguishability of this revised CSP-ElG scheme in the standard model under the $t(n)$-CSP-DDH assumption holds. □

We can similarly construct an encryption scheme using any action of commutative monoid in which the t-MA-DDH assumption holds. Our scheme is generic as the one in [9], and therefore, it is necessary to study the underlying algebraic structures and determine whether a t-MA-DDH holds or not.

4.4 Direct product of submonoids

We can generalize Theorem 3 in [5] as follows. Suppose that M has a direct submonoid decomposition $M \cong L \times N$, that is, we have an isomorphism of $L \times N$ onto M by $(m_1, m_2) \mapsto m_1 m_2$, where $m_1 \in L$ and $m_2 \in N$. We should note that this is not always possible like groups. Note that we have naturally induced actions of L and N on X.

Theorem 3. *Let* $\sigma : X \times M \to X$ *be a monoid action. If MA-DDH assumption holds for the action* $\sigma : X \times L \to X$ *on the point* $x \in X$ *and* $|\mathrm{Ord}(L)| = t$*, then* t*-MA-DDH assumption holds for* $\sigma : X \times M \to X$ *on the point* $x \in X$*.*

Proof. Given $\sigma(x,a), \sigma(x,b) \in X$, we define $\mathcal{X}(\sigma(x,a), \sigma(x,b))$ to be the uniform distribution over $\{\sigma(x,c) \,|\, c \in cL\}$. Then $\mathsf{min\text{-}ent}(\mathcal{X}(\sigma(x,a), \sigma(x,b))) = |\mathrm{Ord}(L)| = t$ since $\mathcal{X}(\sigma(x,a), \sigma(x,b))$ has the same number of elements as L. Let $\mathcal{R}^* = \{(\sigma(x,a), \sigma(x,b), y) \,|\, a, b \in_U M, y \in_{\mathcal{X}(\sigma(x,a),\sigma(x,b))} X\}$. Suppose t-MA-DDH does not hold for the action σ on M on the point x and so we have a distinguisher D between $\mathcal{MA}$–$\mathcal{DH}_M$ and $\mathcal{R}^*$. Using D, we construct a distinguisher D_1 between $\mathcal{MA}$–$\mathcal{DH}_L$ and $\mathcal{R}_L$. Given y_1, y_2, y_3 where $y_1 = \sigma(x, a_L), y_2 = \sigma(x, b_L)$ and y_3 is either $\sigma(x, a_L b_L)$ or $\sigma(x, c_L)$ for $c_L \in_U L$, the distinguisher D_1 does the following. First, choose $a_N, b_N \in_U N$. Second, set $x_1 = y_1 a_N, x_2 = y_2 b_N, x_3 = y_3 a_N b_N$. Third, pass D the triple (x_1, x_2, x_3). Lastly, output the same bits as does D. Note that $x_1 = y_1 a_N = \sigma(x, a_L) a_N = \sigma(x, a_L a_N)$ and $x_2 = y_2 b_N = \sigma(x, b_L) b_N = \sigma(x, b_L b_N)$. If $y_3 = \sigma(x, a_L b_L)$ then we have $x_3 = y_3 a_N b_N = \sigma(x, a_L b_L) a_N b_N = \sigma(x, a_L b_L) a_N b_N = \sigma(x, a_L a_N b_L b_N)$ and so the triple (x_1, x_2, x_3) is a member of $\mathcal{MA}$–$\mathcal{DH}_M$. If $y_3 = \sigma(x, c_L)$ then we have $x_3 = y_3 a_N b_N = \sigma(x, c_L) a_N b_N = \sigma(x, c_L a_N b_N)$. Then (x_1, x_2, x_3) is a member of $\mathcal{R}^*$ since $c_L(a_N b_N) = c_L(a_L b_L)^{-1}(a_L b_L a_N b_N)$ and $c_L(a_L b_L)^{-1} \in_U L$. □

Theorem 3 implies we can strengthen an encryption scheme just by taking a direct product of several monoid actions. We do not know whether or not a concrete monoid can be factorized into a direct product of submonoids and do not guarantee the theorem is always applicable.

References

1. M.Abdalla, M.Bellare and P.Rogaway, The oracle Diffie-Hellman assumptions and an analysis of DHIES, CT-RSA 2001, LNCS Vol. 2020, (2001) 143–158.
2. B.Barak, Y.Dodis, H.Krawczyk, O.Pereira, K.Pietrzak, F.-X.Standaert and Y.Yu, Leftover hash lemma, revisited, CRYPTO 2011, LNCS Vol. 6841, (2011) 1–20.
3. L.Carter and M.N.Wegman, Universal classes of hash functions, J. Computer and System Sciences, 18 (2), (1979) 143–154.
4. D.Cash, E.Kiltz and V.Shoup, The twin Diffie-Hellman problem and applications, EUROCRYPT 2008, LNCS Vol. 4965, (2008) 127–145.
5. R.Gennaro, H.Krawczyk and T.Rabin, Secure hashed Diffie-Hellman over non-DDH groups, EUROCRYPT 2004, LNCS Vol. 3027, (2004) 361–381.
6. D.Grigoriev and V.Shpilrain, Authentication from matrix conjugation, Ggourps, Complexity and Cryptology, 1 (2), (2009) 199–205.
7. J.Hastad, R.Impagliazzo, L.Levin and M.Luby, Construction of a pseudo-random generator from any one-way function, SIAM J. Computing, 28 (4), (1999) 1364–1396.
8. K.Ko, S.Lee, J.Cheon, J.Han, J.Kang and C.Park, New Public-Key Cryptosystem Using Braid Groups, CRYPTO 2000, LNCS Vol. 1880, (2000) 166–183.
9. L.Wang, L.Wang, Z.Cao, E.Okamoto and J.Shao, New Constructions of Public-Key Encryption Schemes from Conjugacy Search Problems, INSCRYPT 2010, LNCS Vol. 6584, (2011) 1–17.
10. A.Yamamura, A functional cryptosystem using a group action, ACISP 1999, LNCS Vol. 1587, (1999) 314–325.
11. A.Yamamura and K.Kurosawa, Generic algorithms and key agreement protocols based on group actions, ISAAC 2001, LNCS Vol. 2223, (2001) 208–218.