On CCA1-Security of Elgamal And Damgård Cryptosystems

Helger Lipmaa

University College London, UK

Abstract. Denote by $X^{Y[i]}$ the assumption that the adversary, given a non-adaptive oracle access to the Y oracle with i free variables cannot break the assumption X. We show that Elgamal is CCA1-secure under the $\mathsf{DDH}^{\mathsf{CDH}[1]}$ assumption. We then give a simple proof that the Damgård cryptosystem is CCA1-secure under the $\mathsf{DDH}^{\mathsf{DDH}[2]}$ assumption, where the proof uses a recent trapdoor test trick by Cash, Kiltz and Shoup.
Keywords. CCA1-security, Damgård cryptosystem, DDH, Elgamal cryptosystem.

1 Introduction

While the Elgamal cryptosystem [Elg85] is one of the best-known public-key cryptosystems, results on its security have been slow to come. Only in 1998, it was proven that Elgamal is CPA-secure [TY98]. On the other hand, it is clearly not CCA2-secure because it is homomorphic. However, not much is known about its CCA1-security.

Denote by $X^{Y[i]}$ the assumption that the adversary, given a non-adaptive oracle access to the Y oracle with i free variables cannot break the assumption X. We show that Elgamal is CCA1-secure under the $\mathsf{DDH}^{\mathsf{CDH}[1]}$ assumption. We also give an opposite result, showing that if Elgamal is CCA1-secure then the $\mathsf{DDH}^{\mathsf{CDH}[1]}$ assumption holds. In some sense, this result just states that Elgamal is CCA1-secure iff it is CCA1-secure, but it serves as an introduction to the second main result. Moreover, this is the weakest assumption up to know under which Elgamal has been proven CCA1-secure.

In 1991, Damgård proposed the Damgård cryptosystem [Dam91] and proved it to be CCA1-secure under a knowledge-of-the-exponent assumption. Only recently it was shown that Damgård is CCA1-secure under a more standard but still relatively strong $\mathsf{DDH}^{\mathsf{DDH}[2]}$ assumption [Gjø06]. We first give a simple proof that Damgård is CCA1-secure iff the $\mathsf{2DDH}^{\mathsf{2CDH}[1]}$ assumption holds, where both $\mathsf{2DDH}$ and $\mathsf{2CDH}$ are related to recent assumptions by Cash, Kiltz and Shoup [CKS08]. We also show that one can derive the $\mathsf{2DDH}^{\mathsf{2CDH}[1]}$ assumption under the more traditional $\mathsf{DDH}^{\mathsf{DDH}[2]}$ assumption. Finally, we give a simple direct proof that Damgård is CCA-1 secure under the $\mathsf{DDH}^{\mathsf{DDH}[2]}$ assumption. Our security proof is considerably simpler than the proof given in [Gjø06] that used a sequence of several games. Our proof uses a simple reduction.

In addition, we conjecture that $\mathsf{DDH}^{\mathsf{CDH}[2]}$ is a stronger assumption than $\mathsf{DDH}^{\mathsf{DDH}[1]}$, which on the other hand is a stronger assumption than $\mathsf{2DDH}^{\mathsf{2CDH}[1]}$, which is a stronger assumption than DDH. That is, we prove that:

$$\text{Elgamal-CCA1} = \mathsf{DDH}^{\mathsf{CDH}[2]} \geq \mathsf{DDH}^{\mathsf{DDH}[1]} \geq \mathsf{2DDH}^{\mathsf{2CDH}[1]} = \text{Damgård-CCA1} \geq \mathsf{DDH} \ ,$$

and we conjecture that one can replace $\geq$ always with a $>$.

2 Preliminaries

2.1 Assumptions

Denote $\mathsf{cdh}(g, g^x, g^y) := g^{xy}$, $\mathsf{ddh}(g, g^x, g^y, g^z) := [g^z =^? \mathsf{cdh}(g, g^x, g^y)]$, $\mathsf{2cdh}(g, g^{x_1}, g^{x_2}, g^y) := (g^{x_1 y}, g^{x_2 y})$.

Definition 1 (CDH game). *Fix a group $\mathbb{G} = \langle g \rangle$ of order q. The* CDH *game is defined as follows:*

Setup phase. *Challenger sets* $\mathsf{sk} \leftarrow \mathbb{Z}_q$, $\mathsf{pk} \leftarrow g^{\mathsf{sk}}$. *He sends* pk *to the adversary* $\mathcal{A}$.
Challenge phase. *Challenger sets* $b_{\mathcal{A}} \leftarrow \{0,1\}$, $\hat{y} \leftarrow \mathbb{Z}_q$, $\hat{h} \leftarrow g^{\hat{y}}$. *Challenger sends* $\hat{h}$ *to* $\mathcal{A}$.
Guess phase. $\mathcal{A}$ *returns a group element* $\hat{h}_{\mathcal{A}} \in \mathbb{G}$. *Adversary wins if* $\hat{h}_{\mathcal{A}} = \mathsf{cdh}(g, \mathsf{pk}, \hat{h})$, *i.e., if* $\hat{h}_{\mathcal{A}} = \mathsf{pk}^{\hat{y}}$.

Group $\mathbb{G}$ *is a* (τ, ε)-CDH group *if for any adversary* $\mathcal{A}$ *working in time* τ, $\Pr[\mathcal{A}$ *wins in the* CDH *game*$] \leq \frac{1}{q} + \varepsilon$.

Definition 2 (DDH game). *Fix a group $\mathbb{G} = \langle g \rangle$ of order q. The* DDH *game is defined as follows:*

Setup phase. *Challenger sets* $\mathsf{sk} \leftarrow \mathbb{Z}_q$, $\mathsf{pk} \leftarrow g^{\mathsf{sk}}$. *He sends* pk *to the adversary* $\mathcal{A}$.
Challenge phase. *Challenger sets* $b_{\mathcal{A}} \leftarrow \{0,1\}$, $\hat{y}, \hat{z} \leftarrow \mathbb{Z}_q$, $\hat{h}_1 \leftarrow g^{\hat{y}}$, *and* $\hat{h}_2 \leftarrow g^{\hat{z}}$ *if* $b_{\mathcal{A}} = 0$ *and* $\hat{h}_2 = \mathsf{cdh}(g, \mathsf{pk}, \hat{h}_1) = \mathsf{pk}^{\hat{y}}$ *if* $b_{\mathcal{A}} = 1$. *Challenger sends* $(\hat{h}_1, \hat{h}_2)$ *to* $\mathcal{A}$.
Guess phase. $\mathcal{A}$ *returns a bit* $b_{\mathcal{A}} \in \{0,1\}$. *Adversary wins if* $b'_{\mathcal{A}} = \mathsf{ddh}(g, \mathsf{pk}, \hat{h}_1, \hat{h}_2)$, *i.e., if* $b'_{\mathcal{A}} = b_{\mathcal{A}}$.

Group $\mathbb{G}$ *is a* (τ, ε)-DDH group *if for any adversary* $\mathcal{A}$ *working in time* τ, $\Pr[\mathcal{A}$ *wins in the* DDH *game*$] \leq \frac{1}{2} + \varepsilon$.

Based on arbitrary assumptions $X = X(x_1, \ldots, x_m)$ and $Y = Y(y_1, \ldots, y_n)$ we define a new assumption $X^{Y[i]}$. In the $X^{Y[i]}$ game, adversary has oracle access to an oracle solving assumption Y with parameters y_j, and she has to break a random instance of the X assumption with parameters x_i. In addition, the queries are restricted so that only i last variables $(y_{n-i+1}, \ldots, y_n)$ in any query are chosen by the adversary, while other variables $(y_1, \ldots, y_{n-i})$ must coincide with the first variables $(x_1, \ldots, x_{n-i})$ of the instance of X assumption she is trying to solve. For the sake of clarity, we now give a concrete definition of the $\mathsf{DDH}^{\mathsf{DDH}[2]}$ game.

Definition 3 ($\mathsf{DDH}^{\mathsf{DDH}[2]}$ game). *Fix a group $\mathbb{G} = \langle g \rangle$ of order q. The* $\mathsf{DDH}^{\mathsf{DDH}[2]}$ *game is defined as follows:*

Setup phase. *Challenger sets* $\mathsf{sk} \leftarrow \mathbb{Z}_q$, $\mathsf{pk} \leftarrow g^{\mathsf{sk}}$. *He sends* pk *to adversary.*
Query phase. *Adversary has an access to oracle* $\mathsf{ddh}(g, \mathsf{pk}, \cdot, \cdot)$.
Challenge phase. *Challenger sets* $b_{\mathcal{A}} \leftarrow \{0,1\}$, $\hat{y}, \hat{z} \leftarrow \mathbb{Z}_q$, $\hat{h}_1 \leftarrow g^{\hat{y}}$, *and* $\hat{h}_2 \leftarrow g^{\hat{z}}$ *if* $b_{\mathcal{A}} = 0$ *and* $\hat{h}_2 = \mathsf{cdh}(g, \mathsf{pk}, \hat{h}_1) = \mathsf{pk}^{\hat{y}}$ *if* $b_{\mathcal{A}} = 1$. *Challenger sends* $(\hat{h}_1, \hat{h}_2)$ *to adversary.*
Guess phase. *Adversary returns a bit* $b'_{\mathcal{A}} \in \{0,1\}$. *Adversary wins if* $b'_{\mathcal{A}} = b_{\mathcal{A}}$, *i.e., if* $b'_{\mathcal{A}} = \mathsf{ddh}(g, \mathsf{pk}, \hat{h}_1, \hat{h}_2)$.

Group $\mathbb{G}$ *is a* (τ, ε)-$\mathsf{DDH}^{\mathsf{DDH}[2]}$ group *if for any adversary* $\mathcal{A}$ *working in time* τ, $\Pr[\mathcal{A}$ *wins in the* $\mathsf{DDH}^{\mathsf{DDH}[2]}$ *game*$] \leq \frac{1}{2} + \varepsilon$.

Clearly, $X^{Y[i]}$ becomes stronger if either X becomes stronger, Y becomes weaker or i becomes larger. The gap DH assumption of [OP01] is equal to $\mathsf{CDH}^{\mathsf{DDH}[3]}$. The strong DH assumption of [ABR01] is slightly different, giving first access to $\hat{h}_1$ and the oracle $\mathsf{ddh}(g, \cdot, \cdot, \mathsf{sk})$, and then asking to compute $\mathsf{cdh}(g, \mathsf{pk}, \hat{h}_1)$.

2.2 Cryptosystems

A public-key cryptosystem Π is a triple of efficient algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$, where $\mathcal{G}(1^k)$ outputs a key pair $(\mathsf{sk}, \mathsf{pk})$, $\mathcal{E}_{\mathsf{pk}}(m; r)$ returns a ciphertext and $\mathcal{D}_{\mathsf{sk}}(c)$ returns a plaintext, so that $\mathcal{D}_{\mathsf{sk}}(\mathcal{E}_{\mathsf{pk}}(m; r)) = m$ for any $(\mathsf{sk}, \mathsf{pk}) \in \mathcal{G}(1^k)$. Here, k is a security parameter that we will just handle as a constant.

Definition 4. *Fix a group $\mathbb{G} = \langle g \rangle$ of order q. The Elgamal cryptosystem [Elg85] is defined as follows:*

Key generation $\mathcal{G}(1^k)$. *Select a random $\mathsf{sk} \leftarrow \mathbb{Z}_q$, set $\mathsf{pk} \leftarrow g^{\mathsf{sk}}$. Publish pk.*
Encryption $\mathcal{E}_{\mathsf{pk}}(m; \cdot)$. *Select a random $r \leftarrow \mathbb{Z}_q$, set $\mathcal{E}_{\mathsf{pk}}(m; r) := (m \cdot \mathsf{pk}^r, g^r)$.*
Decryption $\mathcal{D}_{\mathsf{sk}}(c)$. *Parse $c = (c_1, c_2)$, return $\perp$ if $c_i \notin \mathbb{G}$. Return $m := c_1 / c_2^{\mathsf{sk}}$.*

Definition 5. *Fix a group $\mathbb{G} = \langle g \rangle$ of order q. The Damgård cryptosystem [Dam91] is defined as follows:*

Key generation $\mathcal{G}(1^k)$. *Select random $\mathsf{sk}_1, \mathsf{sk}_2 \leftarrow \mathbb{Z}_q$, set $\mathsf{pk}_1 \leftarrow g^{\mathsf{sk}_1}, \mathsf{pk}_2 \leftarrow g^{\mathsf{sk}_2}$. Publish $\mathsf{pk} = (\mathsf{pk}_1, \mathsf{pk}_2)$, set $\mathsf{sk} = (\mathsf{sk}_1, \mathsf{sk}_2)$.*
Encryption $\mathcal{E}_{\mathsf{pk}}(m; \cdot)$. *Select a random $r \leftarrow \mathbb{Z}_q$, set $\mathcal{E}_{\mathsf{pk}}(m; r) := (m \cdot \mathsf{pk}_1^r, \mathsf{pk}_2^r, g^r)$.*
Decryption $\mathcal{D}_{\mathsf{sk}}(c)$. *Parse $c = (c_1, c_2, c_3)$, return $\perp$ if $c_i \notin \mathbb{G}$. Return $\perp$ if $c_2 \neq c_3^{\mathsf{sk}_2}$. Return $m := c_1 / c_3^{\mathsf{sk}_1}$.*

Definition 6. *Let $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a public-key cryptosystem. The CCA1-game is defined as follows:*

Setup phase. *Challenger chooses $(\mathsf{pk} = g^{\mathsf{sk}}, \mathsf{sk}) \leftarrow \mathcal{G}(1^k)$ and sends pk to adversary.*
Query phase. *Adversary has access to oracle $\mathcal{O}(\cdot)$ such that $\mathcal{O}(c) = \mathcal{D}_{\mathsf{sk}}(c)$.*
Challenge phase. *Adversary submits (m_0, m_1) to challenger, who picks a random bit $b \leftarrow \{0, 1\}$ and a random $r \leftarrow \mathbb{Z}_q$, and returns $\mathcal{E}_{\mathsf{pk}}(m_b; r)$.*
Guess phase. *Adversary returns a bit $b_{\mathcal{A}} \in \{0, 1\}$. Adversary wins if $b_{\mathcal{A}} = b$.*

Public-key cryptosystem is $(\tau, \gamma, \varepsilon)$-CCA1-secure if for any adversary $\mathcal{A}$ working in time τ and making γ queries, $\Pr[\mathcal{A}\ wins] \leq \frac{1}{2} + \varepsilon$. A $(\tau, 0, \varepsilon)$-CCA1-secure cryptosystem is also said to be (τ, ε)-CPA-secure.

Damgård's cryptosystem was proven to be CCA1-secure under the $\mathsf{DDH}^{\mathsf{DDH}[2]}$ assumption in [Gjø06]. Elgamal's cryptosystem is known to be CPA-secure but not known to be CCA1-secure for $\gamma = \mathrm{poly}(k)$.

3 CCA1-Security of ElGamal

To prove the security of ElGamal we need the next assumption (see the end of the section for discussion). As seen from the security proof, this assumption in some sense just asserts that Elgamal is CCA1-secure.

Definition 7 ($\mathsf{DDH}^{\mathsf{CDH}[1]}$ game). *Fix a group $\mathbb{G} = \langle g \rangle$ of order q. The $\mathsf{DDH}^{\mathsf{CDH}[1]}$ game is defined as follows:*

Setup phase. *Challenger sets $\mathsf{sk} \leftarrow \mathbb{Z}_q$, $\mathsf{pk} \leftarrow g^{\mathsf{sk}}$. He sends pk to adversary $\mathcal{A}$.*
Query phase. *$\mathcal{A}$ has access to oracle $\mathsf{cdh}(g, \mathsf{pk}, \cdot)$, i.e., $\mathsf{cdh}(g, \mathsf{pk}, h) := h^{\mathsf{sk}}$.*
Challenge phase. *Challenger sets $b_{\mathcal{A}} \leftarrow \{0, 1\}$, $\hat{y}, \hat{z} \leftarrow \mathbb{Z}_q$, $\hat{h}_1 \leftarrow g^{\hat{y}}$, and $\hat{h}_2 \leftarrow g^{\hat{z}}$ if $b_{\mathcal{A}} = 0$ and $\hat{h}_2 = \mathsf{cdh}(g, \mathsf{pk}, \hat{h}_1) = \mathsf{pk}^{\hat{y}}$ if $b_{\mathcal{A}} = 1$. Challenger sends $(\hat{h}_1, \hat{h}_2)$ to $\mathcal{A}$.*

Guess phase. $\mathcal{A}$ *returns a bit* $b'_{\mathcal{A}} \in \{0,1\}$. $\mathcal{A}$ *wins if* $b'_{\mathcal{A}} = b_{\mathcal{A}}$, *i.e., if* $b_{\mathcal{A}} = \mathsf{ddh}(g, \mathsf{pk}, \hat{h}_1, \hat{h}_2)$.

Group $\mathbb{G}$ *is a* $(\tau, \gamma, \varepsilon)$-$\mathsf{DDH}^{\mathsf{CDH}[1]}$ group *if for any adversary* $\mathcal{A}$ *working in time* τ *and making* γ *queries,* $\Pr[\mathcal{A}$ *wins in the* $\mathsf{DDH}^{\mathsf{CDH}[1]}$ *game*$] \le \frac{1}{2} + \varepsilon$.

Theorem 1 ($\mathsf{DDH}^{\mathsf{CDH}[1]} \Rightarrow$ Elgamal-CCA1). *Assume that* $\mathbb{G} = \langle g \rangle$ *is a* $(\tau, \gamma, \varepsilon)$-$\mathsf{DDH}^{\mathsf{CDH}[1]}$ *group. Then ElGamal is* $(\tau - \gamma \cdot (\tau_{\mathsf{cdh}} + \mathsf{small}) - \mathsf{small}, \gamma, \varepsilon/2)$-*CCA1-secure where* τ_{cdh} *is the working time of the* $\mathsf{cdh}(g, \mathsf{pk}, \cdot)$ *oracle.*

Proof. Assume $\mathcal{A}$ is an adversary who can $(\tau, \gamma, \varepsilon)$-break the CCA1-security of Elgamal with probability ε and in time τ, making γ queries. Construct the next adversary $\mathcal{B}$ who aims to break $\mathsf{DDH}^{\mathsf{CDH}[1]}$:

- Challenger generates new keypair $(\mathsf{sk}, \mathsf{pk} \leftarrow g^{\mathsf{sk}})$ and sends pk to $\mathcal{B}$. $\mathcal{B}$ forwards pk to $\mathcal{A}$.
- In the query phase, whenever $\mathcal{A}$ asks a decryption query (c_1, c_2), $\mathcal{B}$ rejects if either c_1 or c_2 is not a valid group element. Otherwise $\mathcal{B}$ asks a CDH query $c \leftarrow \mathsf{cdh}(g, \mathsf{pk}, c_2)$. $\mathcal{B}$ returns c_1/c.
- In the challenge phase, whenever $\mathcal{A}$ gives a pair $(\hat{m}_0, \hat{m}_1)$ of equal-length messages, $\mathcal{B}$ asks his challenge from the challenger. The challenger sets $b_{\mathcal{B}} \leftarrow \{0,1\}$, $\hat{y}, \hat{z} \leftarrow \mathbb{Z}_q$, $\hat{h}_2 \leftarrow g^{\hat{y}}$. If $b_{\mathcal{B}} = 0$ then he sets $\hat{h}_1 \leftarrow g^{\hat{z}}$, otherwise $\hat{h}_1 \leftarrow \mathsf{pk}^{\hat{y}}$. $\mathcal{B}$ picks a random bit $b_{\mathcal{A}} \leftarrow \{0,1\}$ and sends $(m_{b_{\mathcal{A}}} \cdot h_1, h_2)$ to $\mathcal{A}$. $\mathcal{A}$ returns a bit $b'_{\mathcal{A}}$. If $b'_{\mathcal{A}} = b_{\mathcal{A}}$ then $\mathcal{B}$ returns $b'_{\mathcal{B}} = 1$, otherwise $\mathcal{B}$ returns $b'_{\mathcal{B}} = 0$.

Now, $\Pr[\mathcal{B}$ wins in the $\mathsf{DDH}^{\mathsf{CDH}[1]}$ game$] = \Pr[b'_{\mathcal{B}} = b_{\mathcal{B}}] = \Pr[\mathcal{A}$ wins$|b_{\mathcal{B}} = 1] \Pr[b_{\mathcal{B}} = 1] + \Pr[\mathcal{A}$ wins$|b_{\mathcal{B}} = 0] \Pr[b_{\mathcal{B}} = 0] = \varepsilon/2 + 0/2 = \varepsilon/2$. Clearly $\mathcal{B}$ works in time $\tau_{\mathcal{B}} = \gamma \cdot (\tau_{\mathsf{cdh}} + \mathsf{small}) + \mathsf{small} + \tau$. □

Theorem 2 (Elgamal-CCA1 $\Rightarrow \mathsf{DDH}^{\mathsf{CDH}[1]}$). *Assume that ElGamal is* $(\tau, \gamma, \varepsilon)$-*CCA1-secure. Then* $\mathbb{G} = \langle g \rangle$ *is a* $(\tau - \gamma \cdot (\tau_{\mathsf{cdh}} + \mathsf{small}) - \mathsf{small}, \gamma, \varepsilon)$-$\mathsf{DDH}^{\mathsf{CDH}[1]}$ *group, where* $\tau_{\mathcal{D}}$ *is the working time of the* $\mathcal{D}$ *oracle.*

Proof. Assume $\mathcal{A}$ is an adversary who can $(\tau, \gamma, \varepsilon)$-break the $\mathsf{DDH}^{\mathsf{CDH}[1]}$ assumption in group $\mathbb{G}$. Construct the next adversary $\mathcal{B}$ who aims to break the CCA1-security of Elgamal:

- Challenger generates new keypair $(\mathsf{sk}, \mathsf{pk} \leftarrow g^{\mathsf{sk}})$ and sends pk to $\mathcal{B}$. $\mathcal{B}$ forwards pk to $\mathcal{A}$.
- In the query phase, whenever $\mathcal{A}$ asks a $\mathsf{cdh}(g, \mathsf{pk}, \cdot)$ query h, $\mathcal{B}$ asks a decryption query $(1, h)$, and receives back $c \leftarrow 1/h^{sk}$. $\mathcal{B}$ then returns $1/c$.
- In the challenge phase, whenever $\mathcal{A}$ asks for a challenge, $\mathcal{B}$ sends his challenge $(\hat{m}_0, \hat{m}_1) \leftarrow (g, 1)$ to challenger. Challenger pick a random bit $b_{\mathcal{B}} \leftarrow \{0,1\}$ and a random $\hat{r} \leftarrow \mathbb{Z}_q$, and sends $(\hat{c}_1, \hat{c}_2) \leftarrow (g^{1-b_{\mathcal{B}}} \cdot \mathsf{pk}^{\hat{r}}, g^{\hat{r}})$ to $\mathcal{B}$. $\mathcal{B}$ forwards $(\hat{c}_2, \hat{c}_1)$ to $\mathcal{A}$, who returns a guess $b'_{\mathcal{A}}$. $\mathcal{B}$ returns $b'_{\mathcal{B}} \leftarrow b'_{\mathcal{A}}$ to challenger.

Now, $\Pr[\mathcal{B}$ wins$] = \Pr[b'_{\mathcal{B}} = b_{\mathcal{B}}] = \Pr[\mathcal{A}$ wins$] = \varepsilon$. Clearly $\mathcal{B}$ works in time $\tau_{\mathcal{B}} = \gamma \cdot (\tau_{\mathcal{D}} + \mathsf{small}) + \mathsf{small} + \tau$. □

Discussion. The $\mathsf{DDH}^{\mathsf{CDH}[1]}$ assumption is a direct opposite of the well-known StrongCDH assumption [OP01] where one has initial access to the DDH oracle and then has to compute CDH. Thus, $\mathsf{DDH}^{\mathsf{CDH}[1]}$ is strictly stronger than the StrongDH assumption. However, it is not clear if this assumption is actually stronger than the DDH assumption itself.

In the $\mathsf{DDH}^{\mathsf{CDH}[1]}$-game the adversary can always submit pk to the oracle and obtain g^{sk^2}, and recursively g^{sk^i} for polynomially many values i. This means that in some sense we can also assume that in the challenge phase those values are given for the adversary for free (though their number should be included in the number of made queries).

4 CCA1-Security of Damgård

Denote $\mathsf{2ddh}(g, g^{x_1}, g^{x_2}, g^y, g^{z_1}, g^{z_2}) := [(g^{z_1}, g^{z_2}) =^? \mathsf{2cdh}(g, g^{x_1}, g^{x_2}, g^y)]$ [CKS08].

The next assumption basically states that Damgård is CCA1-secure.

Definition 8 ($\mathsf{2DDH}^{\mathsf{2CDH}[1]}$ game). *Fix a group $\mathbb{G} = \langle g \rangle$ of order q. The $\mathsf{2DDH}^{\mathsf{2CDH}[1]}$ game is defined as follows:*

Setup phase. *Challenger sets $\mathsf{sk}_1, \mathsf{sk}_2 \leftarrow \mathbb{Z}_q$, $\mathsf{pk}_1 \leftarrow g^{\mathsf{sk}_1}$, $\mathsf{pk}_2 \leftarrow g^{\mathsf{sk}_2}$. He sends $\mathsf{pk} \leftarrow (\mathsf{pk}_1, \mathsf{pk}_2)$ to adversary and sets $\mathsf{sk} \leftarrow (\mathsf{sk}_1, \mathsf{sk}_2)$.*

Query phase. *Adversary has access to oracle $\mathcal{O}(\cdot,\cdot)$ such that: $\mathcal{O}(h_2, h_3)$ computes $(z_1, z_2) \leftarrow \mathsf{2cdh}(g, \mathsf{pk}_1, \mathsf{pk}_2, h_3)$. If $z_2 \neq h_2$, it returns $\perp$, otherwise it returns z_1.*

Challenge phase. *Challenger sets $b_{\mathcal{A}} \leftarrow \{0,1\}$, $\hat{y}, \hat{z} \leftarrow \mathbb{Z}_q$, $\hat{h}_3 \leftarrow g^{\hat{y}}$. If $b_{\mathcal{A}} = 0$ then $\hat{h}_2 = \mathsf{cdh}(g, \mathsf{pk}_2, \hat{h}_3)$ and $\hat{h}_1 \leftarrow \mathbb{G}$. If $b_{\mathcal{A}} = 1$ then $(\hat{h}_1, \hat{h}_2) \leftarrow \mathsf{2cdh}(g, \mathsf{pk}_1, \mathsf{pk}_2, \hat{h}_3)$. Challenger sends $(\hat{h}_1, \hat{h}_2, \hat{h}_3)$ to adversary.*

Guess phase. *Adversary returns a bit $b'_{\mathcal{A}} \in \{0,1\}$. Adversary wins if $b'_{\mathcal{A}} = b_{\mathcal{A}}$.*

Group $\mathbb{G}$ is a $(\tau, \gamma, \varepsilon)$-$\mathsf{2DDH}^{\mathsf{2CDH}[1]}$ group if for any adversary $\mathcal{A}$ working in time τ and making γ queries, $\Pr[\mathcal{A}\ wins] \leq \frac{1}{2} + \varepsilon$.

Theorem 3 ($\mathsf{2DDH}^{\mathsf{2CDH}[1]} \Rightarrow$ Damgård-CCA1). *Assume that $\mathbb{G} = \langle g \rangle$ is a $(\tau, \gamma, \varepsilon)$-$\mathsf{2DDH}^{\mathsf{2CDH}[1]}$ group. Then Damgård is $(\tau - \gamma \cdot (\tau_{\mathsf{ddh}} + \mathsf{small}) - \mathsf{small}, \gamma, \varepsilon/2)$-CCA1-secure where τ_{2cdh} is the working time of the $\mathsf{2cdh}(g, \mathsf{pk}_1, \mathsf{pk}_2, \cdot)$ oracle.*

Proof. Assume $\mathcal{A}$ is an adversary who can $(\tau, \gamma, \varepsilon)$-break the CCA1-security of Elgamal with probability ε and in time τ, making γ queries. Construct the next adversary $\mathcal{B}$ who aims to break $\mathsf{2DDH}^{\mathsf{2CDH}[1]}$:

- Challenger generates new $\mathsf{sk} \leftarrow (\mathsf{sk}_1, \mathsf{sk}_2)$, $\mathsf{pk}_1 \leftarrow g^{\mathsf{sk}_1}$, $\mathsf{pk}_2 \leftarrow g^{\mathsf{sk}_2}$ and sends pk to $\mathcal{B}$. $\mathcal{B}$ forwards $\mathsf{pk} = (\mathsf{pk}_1, \mathsf{pk}_2)$ to $\mathcal{A}$.
- In the query phase, whenever $\mathcal{A}$ asks a decryption query (c_1, c_2, c_3), $\mathcal{B}$ rejects if either c_1, c_2 or c_3 is not a valid group element. Otherwise $\mathcal{B}$ makes a $\mathcal{O}(c_3, c_2)$ query. $\mathcal{B}$ receives a c such that $c = \perp$, if $c_2 \neq c_3^{\mathsf{sk}_2}$, and $c = c_3^{\mathsf{sk}_1}$ otherwise. $\mathcal{B}$ returns $\perp$ in the first case, and c_1/c in the second case.
- In the challenge phase, whenever $\mathcal{A}$ submits her challenge $(\hat{m}_0, \hat{m}_1)$, $\mathcal{B}$ asks the challenger for his challenge. The challenger sets $b_{\mathcal{B}} \leftarrow \{0,1\}$, $\hat{y}, \hat{z} \leftarrow \mathbb{Z}_q$, $\hat{h}_3 \leftarrow g^{\hat{y}}$, $\hat{h}_2 \leftarrow \mathsf{pk}_2^{\hat{y}}$. If $b_{\mathcal{B}} = 0$ then he sets $\hat{h}_1 \leftarrow g^{\hat{z}}$, otherwise $\hat{h}_1 \leftarrow \mathsf{pk}_1^{\hat{y}}$. $\mathcal{B}$ picks a random bit $b_{\mathcal{A}} \leftarrow \{0,1\}$ and sends $(\hat{m}_{b_{\mathcal{A}}} \cdot \hat{h}_1, \hat{h}_2, \hat{h}_3)$ to $\mathcal{A}$. $\mathcal{A}$ returns a bit $b'_{\mathcal{A}}$. If $b'_{\mathcal{A}} = b_{\mathcal{A}}$ then $\mathcal{B}$ returns $b'_{\mathcal{B}} = 1$, otherwise $\mathcal{B}$ returns $b'_{\mathcal{B}} = 0$.

Now, $\Pr[\mathcal{B}\text{ wins}] = \Pr[b'_{\mathcal{B}} = b_{\mathcal{B}}] = \Pr[\mathcal{A}\text{ wins} | b_{\mathcal{B}} = 1] \Pr[b_{\mathcal{B}} = 1] + \Pr[\mathcal{A}\text{ wins} | b_{\mathcal{B}} = 0] \Pr[b_{\mathcal{B}} = 0] = \varepsilon/2 + 0/2 = \varepsilon/2$. Clearly $\mathcal{B}$ works in time $\tau_B = \mathsf{sk} \cdot (\tau_{\mathsf{2cdh}} + \mathsf{small}) + \mathsf{small} + \tau$. □

Theorem 4 (Damgård-CCA1 $\Rightarrow$ $\mathsf{2DDH}^{\mathsf{2CDH}[1]}$). *Fix a group $\mathbb{G} = \langle g \rangle$ of order q. Assume that Damgård is $(\tau, \gamma, \varepsilon)$-CCA1-secure. Then $\mathbb{G}$ is a $(\tau - \gamma \cdot (\tau_{\mathcal{D}} + \mathsf{small}) - \mathsf{small}, \gamma, \varepsilon)$-$\mathsf{2DDH}^{\mathsf{2CDH}[1]}$ group, where $\tau_{\mathcal{D}}$ is the working time of the $\mathcal{D}$ oracle.*

Proof. Assume $\mathcal{A}$ is an adversary who can $(\tau, \gamma, \varepsilon)$-break the $\mathsf{2DDH}^{\mathsf{2CDH}[1]}$ property. Construct the next adversary $\mathcal{B}$ who aims to break the CCA1-security of the Damgård cryptosystem:

- Challenger generates new $\mathsf{sk} \leftarrow (\mathsf{sk}_1, \mathsf{sk}_2)$, $\mathsf{pk}_1 \leftarrow g^{\mathsf{sk}_1}$, $\mathsf{pk}_2 \leftarrow g^{\mathsf{sk}_2}$ and sends pk to $\mathcal{B}$. $\mathcal{B}$ forwards $\mathsf{pk} = (\mathsf{pk}_1, \mathsf{pk}_2)$ to $\mathcal{A}$.

- In the query phase, whenever $\mathcal{A}$ asks a $\mathcal{O}$ query (h_2, h_3), $\mathcal{B}$ makes a decryption query $(1, h_2, h_3)$, and receives back either $\perp$ or $m \leftarrow 1/h_3^{\mathsf{sk}_2}$. $\mathcal{B}$ returns $\perp$ in the first case, and $1/m$ in the second case.
- In the challenge phase, whenever $\mathcal{A}$ asks for a challenge, $\mathcal{B}$ sends his challenge $(\hat{m}_0, \hat{m}_1) \leftarrow (g, 1)$ to the challenger. Challenger picks a random bit $b_{\mathcal{B}} \leftarrow \{0,1\}$ and a random $\hat{r} \leftarrow \mathbb{Z}_q$, and sends $(\hat{c}_1, \hat{c}_2, \hat{c}_3)) \leftarrow (g^{1-b_{\mathcal{B}}} \cdot \mathsf{pk}_1^{\hat{r}}, \mathsf{pk}_2^{\hat{r}}, g^{\hat{r}})$ to $\mathcal{B}$. $\mathcal{B}$ forwards $(\hat{c}_1, \hat{c}_2, \hat{c}_3)$ to $\mathcal{A}$, who returns a guess $b'_{\mathcal{A}}$.
- In the guess phase, $\mathcal{B}$ returns $b'_{\mathcal{B}} \leftarrow b'_{\mathcal{A}}$ to challenger.

Now, $\Pr[\mathcal{B} \text{ wins}] = \Pr[b'_{\mathcal{B}} = b_{\mathcal{B}}] = \Pr[\mathcal{A} \text{ wins}] = \varepsilon$. Clearly $\mathcal{B}$ works in time $\tau_B = \mathsf{sk} \cdot (\tau_{\mathcal{D}} + \mathsf{small}) + \mathsf{small} + \tau$. □

Theorem 5 ($\mathsf{DDH}^{\mathsf{DDH}[2]} \Rightarrow \mathsf{2DDH}^{\mathsf{2CDH}[1]}$)**.** *Any $(\tau, \gamma, \varepsilon)$-$\mathsf{DDH}^{\mathsf{DDH}[2]}$ group $\mathbb{G} = \langle g \rangle$ is also a $(\tau - \gamma \cdot (\tau_{\mathsf{cdh}} + \mathsf{small}) - \mathsf{small}, \gamma, \varepsilon)$-$\mathsf{2DDH}^{\mathsf{2DDH}[2]}$, where τ_{cdh} is the working time of the $\mathsf{cdh}(g, \mathsf{pk}, \cdot)$ oracle.*

Proof. We use the trapdoor test trick from [CKS08].

Fix a group $\mathbb{G} = \langle g \rangle$ of order q. Assume $\mathcal{A}$ is an adversary who can $(\tau, \gamma, \varepsilon)$-break the $\mathsf{2DDH}^{\mathsf{2CDH}[1]}$-property. Construct the next adversary $\mathcal{B}$ who aims to break $\mathsf{DDH}^{\mathsf{DDH}[2]}$ in the same group:

- Challenger generates new $(\mathsf{sk}, \mathsf{pk} \leftarrow g^{\mathsf{sk}})$ and sends pk to $\mathcal{B}$. $\mathcal{B}$ sets $\mathsf{pk}_1 \leftarrow \mathsf{pk}$, $r, s \leftarrow \mathbb{Z}_q$, $\mathsf{pk}_2 \leftarrow g^u / \mathsf{pk}_1^v$. Thus $\mathsf{pk}_2 = g^{\mathsf{sk}_2}$ for $\mathsf{sk}_2 = u - v \cdot \mathsf{sk}_1$. He forwards $\mathsf{pk} = (\mathsf{pk}_1, \mathsf{pk}_2)$ to $\mathcal{A}$.
- In the query phase, whenever $\mathcal{A}$ asks an $\mathcal{O}$ query (h_1, h_2), $\mathcal{B}$ makes a $\mathsf{ddh}(g, \mathsf{pk}_1, h_3, h_2)$ query. If this query returns 0, $\mathcal{B}$ rejects. Otherwise, $\mathcal{B}$ computes $z_2 \leftarrow h_3^u / h_2^v$ and outputs z_2. Note that in this case $z_2 = h_3^u / h_2^v = h_3^u / h_3^{v \cdot \mathsf{sk}_1} = h_3^{\mathsf{sk}_2}$ and thus $\mathcal{A}$ gets the correct output.
- In the challenge phase, if $\mathcal{A}$ asks for a challenge then $\mathcal{B}$ asks for a challenge. Challenger sets $b_{\mathcal{B}} \leftarrow \{0,1\}$, $\hat{y}, \hat{z} \leftarrow \mathbb{Z}_q$, $\hat{h}_3 \leftarrow g^{\hat{y}}$. If $b_{\mathcal{B}} = 0$ then he sets $\hat{h}_1 \leftarrow g^{\hat{z}}$ and $\hat{h}_2 \leftarrow g^{\hat{z}}$, otherwise $(\hat{h}_1, \hat{h}_2) \leftarrow (\mathsf{pk}_1^{\hat{y}}, \mathsf{pk}_2^{\hat{y}})$. $\mathcal{B}$ sets $\hat{h}_1 \leftarrow \hat{h}_3^u / \hat{h}_2^v$. $\mathcal{B}$ sends $(\hat{h}_1, \hat{h}_3)$ to $\mathcal{A}$. $\mathcal{A}$ returns a bit $b'_{\mathcal{A}}$. $\mathcal{B}$ returns $b'_{\mathcal{B}} = b'_{\mathcal{A}}$ to the challenger.

Clearly, $\mathcal{B}$ wins iff $\mathcal{A}$ wins. □

Corollary 1 ($\mathsf{DDH}^{\mathsf{DDH}[2]} \Rightarrow$ **Damgård-CCA1**)**.** *Assume that $\mathbb{G} = \langle g \rangle$ is a $(\tau, \gamma, \varepsilon)$-$\mathsf{DDH}^{\mathsf{DDH}[2]}$ group. Then the Damgård cryptosystem is $(\tau - \gamma \cdot (\tau_{\mathsf{cdh}} + \mathsf{small}) - \mathsf{small}, \gamma, \varepsilon)$-CCA1-secure, where τ_{cdh} is the working time of the $\mathsf{cdh}(g, \mathsf{pk}, \cdot)$ oracle.*

By following a very similar proof, a variant of the Damgård cryptosystem where the decryption, given an invalid ciphertext, returns a random plaintext instead of $\perp$, is secure under the DDH assumption.

Direct Proof of $\mathsf{DDH}^{\mathsf{DDH}[2]} \Rightarrow$ Damgård-CCA1. Finally, we give a direct proof of Cor. 1.

Theorem 6 ($\mathsf{DDH}^{\mathsf{DDH}[2]} \Rightarrow$ **Damgård-CCA1**)**.** *Assume that $\mathbb{G} = \langle g \rangle$ is a $(\tau, \gamma, \varepsilon)$-$\mathsf{DDH}^{\mathsf{DDH}[2]}$ group. Then Damgård is $(\tau - \gamma \cdot (\tau_{\mathsf{ddh}} + \mathsf{small}) - \mathsf{small}, \gamma, \varepsilon/2)$-CCA1-secure where τ_{ddh} is the working time of the $\mathsf{ddh}(g, \mathsf{pk}, \cdot, \cdot)$ oracle.*

Proof. Assume $\mathcal{A}$ is an adversary who can $(\tau, \gamma, \varepsilon)$-break the CCA1-security of Damgård with probability ε and in time τ, making γ queries. The proof again uses the trapdoor trick of [CKS08]. Construct the next adversary $\mathcal{B}$ who aims to break $\mathsf{DDH}^{\mathsf{DDH}[2]}$:

- Challenger generates new $(\mathsf{sk}, \mathsf{pk} \leftarrow g^{\mathsf{sk}})$ and sends pk to $\mathcal{B}$. $\mathcal{B}$ sets $\mathsf{pk}_2 \leftarrow \mathsf{pk}$, and $\mathsf{pk}_1 \leftarrow g^u/\mathsf{pk}_2^v$ for random $u, v \in \mathbb{Z}_q$. Thus $\mathsf{pk}_1 = g^{\mathsf{sk}_1}$ for $\mathsf{sk}_1 = u - v \cdot \mathsf{sk}_2$. He forwards $\mathsf{pk} = (\mathsf{pk}_1, \mathsf{pk}_2)$ to $\mathcal{A}$.
- In the query phase, whenever $\mathcal{A}$ asks a decryption query (c_1, c_2, c_3), $\mathcal{B}$ rejects if either c_1, c_2 or c_3 is not a valid group element. Otherwise $\mathcal{B}$ makes a $\mathsf{ddh}(g, \mathsf{pk}_2, c_3, c_2)$ query. If this query returns 0, $\mathcal{B}$ rejects. Otherwise, $\mathcal{B}$ computes $c \leftarrow c_3^u/c_2^v$ and outputs c_1/c. Note that $c = c_3^u/c_2^v = c_3^u/c_3^{v \cdot \mathsf{sk}_2} = c_3^{u - v \cdot \mathsf{sk}_2} = c_3^{\mathsf{sk}_1}$.
- In the challenge phase, whenever $\mathcal{A}$ submits her challenge $(\hat{m}_0, \hat{m}_1)$, $\mathcal{B}$ asks the challenger for his challenge. The challenger sets $b_{\mathcal{B}} \leftarrow \{0, 1\}$, $\hat{y}, \hat{z} \leftarrow \mathbb{Z}_q$, $\hat{h}_3 \leftarrow g^{\hat{y}}$. If $b_{\mathcal{B}} = 0$ then he sets $\hat{h}_2 \leftarrow g^{\hat{z}}$, otherwise $\hat{h}_2 \leftarrow \mathsf{pk}_2^{\hat{y}}$. $\mathcal{B}$ sets $\hat{h}_1 \leftarrow \hat{h}_3^u/\hat{h}_2^v$. Note that if $b_{\mathcal{B}} = 0$ then $\hat{h}_1 = g^{u\hat{y} - v\hat{z}}$ is completely random, and if $b_{\mathcal{B}} = 1$ then $\hat{h}_1 = \hat{h}_3^u/\mathsf{pk}_2^{v\hat{y}} = \hat{h}_3^{u - v \cdot \mathsf{sk}_2} = \hat{h}_3^{\mathsf{sk}_1}$. $\mathcal{B}$ picks a random bit $b_{\mathcal{A}} \leftarrow \{0, 1\}$ and sends $(\hat{m}_{b_{\mathcal{A}}} \cdot \hat{h}_1, \hat{h}_2, \hat{h}_3)$ to $\mathcal{A}$. $\mathcal{A}$ returns a bit $b'_{\mathcal{A}}$. If $b'_{\mathcal{A}} = b_{\mathcal{A}}$ then $\mathcal{B}$ returns $b'_{\mathcal{B}} = 1$, otherwise $\mathcal{B}$ returns $b'_{\mathcal{B}} = 0$.

Now, $\Pr[\mathcal{B} \text{ wins}] = \Pr[b'_{\mathcal{B}} = b_{\mathcal{B}}] = \Pr[\mathcal{A} \text{ wins} | b_{\mathcal{B}} = 1] \Pr[b_{\mathcal{B}} = 1] + \Pr[\mathcal{A} \text{ wins} | b_{\mathcal{B}} = 0] \Pr[b_{\mathcal{B}} = 0] = \varepsilon/2 + 0/2 = \varepsilon/2$. Clearly $\mathcal{B}$ works in time $\tau_B := \mathsf{sk} \cdot (\tau_{\mathsf{ddh}} + \mathsf{small}) + \mathsf{small} + \tau$. □

Acknowledgments. We thank Eike Kiltz for discussions.

References

[ABR01] Michel Abdalla, Mihir Bellare, and Phillip Rogaway. The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES. In David Naccache, editor, *Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 143–158, San Francisco, CA, USA, April 8–12, 2001. Springer-Verlag.

[CKS08] David Cash, Eike Kiltz, and Victor Shoup. The Twin Diffie-Hellman Problem and Applications. In Nigel Smart, editor, *Advances in Cryptology — EUROCRYPT 2008*, volume ? of *Lecture Notes in Computer Science*, pages ?–?, Istanbul, Turkey, April 13–17, 2008. Springer-Verlag.

[Dam91] Ivan Damgård. Towards Practical Public Key Systems Secure Against Chosen Ciphertext Attacks. In Joan Feigenbaum, editor, *Advances in Cryptology—CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 445–456, Santa Barbara, California, USA, August 11–15, 1991. Springer-Verlag, 1992.

[Elg85] Taher Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[Gjø06] Kristian Gjøsteen. A New Security Proof for Damgård's ElGamal. In David Pointcheval, editor, *Topics in Cryptology - CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006*, volume 3860 of *Lecture Notes in Computer Science*, pages 150–158, San Jose, CA, USA, February 13–17, 2006. Springer-Verlag.

[OP01] Tatsuaki Okamoto and David Pointcheval. The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes. In Kwangjo Kim, editor, *Public Key Cryptography 2001*, volume 1992 of *Lecture Notes in Computer Science*, pages 104–118, Cheju Island, Korea, February 13–15, 2001. Springer-Verlag.

[TY98] Yannis Tsiounis and Moti Yung. On the Security of ElGamal-Based Encryption. In Hideki Imai and Yuliang Zheng, editors, *Public Key Cryptography 1998*, volume 1431 of *Lecture Notes in Computer Science*, pages 117–134, Pacifico Yokohama, Japan, 5–6 February 1998. Springer-Verlag.