MTAT.07.007 Graduate Seminar in Cryptography Duality Between Encryption and Commitments

Liina Kamm
University of Tartu

May 21, 2007

1 Introduction

When one looks at commitment and encryption schemes, it is rather easy to spot the similarities between the two cryptographic notions. Although their functionalities are different and they are used for different purposes, they share a similar structure. Both encryptions and commitment make use of public keys, encryption schemes have an additional secret key. In both cases the secret hiding and revealing stages occur, only the way these phases are handled differ. In the case of an encryption scheme, we use the public key to lock a message and the secret key to open it again. Commitment schemes also use the public key to lock a message, but to instead of unlocking, one usually needs to reveal the message in order to open a commitment.

It is always possible to make a commitment scheme from an encryption scheme. Making an encryption scheme from a commitment scheme is trickier though, because there exists no secret key in the simple commitment scheme construction. In order to make an encryption scheme from a commitment scheme, the latter needs to be extractable. The trapdoor information that an extractable commitment can reveal is not used in commitment schemes because it would break their security. It is however used for example in constructing encryption schemes and in zero knowledge proofs.

In this seminar paper we look at the components and properties of commitment and encryption schemes. In section 4, we take a closer look at extractability, the commitment scheme property necessary for constructing encryption schemes. We go on to look at how to make an encryption scheme from a commitment scheme and *vice versa* in section 5. And finally in sections 6 and 7 we take a look how different properties are dual for the two notions.

2 Commitment Schemes

In describing the commitment schemes, we follow the classical formalisation that has also been used in [LN06]. A commitment scheme consists of three parts:

key generation Gen, commitment Com and opening Open. The key generation algorithm generates the public parameters $\mathsf{pk} \leftarrow \mathsf{Gen}$. The commitment algorithm $\mathsf{Com}_{\mathsf{pk}} : \mathcal{M} \times \mathcal{R} \rightarrow \mathcal{C} \times \mathcal{D}$ computes the commitment string c of fixed length and a decommitment value d from the message $m \in \mathcal{M}$. Very often $d = (m, r)$, where $r \in \mathcal{R}$ is the randomness used in the commitment. The opening algorithm $\mathsf{Open}_{\mathsf{pk}} : \mathcal{C} \times \mathcal{D} \rightarrow \mathcal{M} \cup \{\perp\}$, given the correct commitment and decommitment values outputs the message m. If the decommitment value is incorrect, the algorithm outputs the abort value $\perp$.

Commitment schemes have two essential properties—hiding and binding.

Definition 2.1. *A commitment scheme is (t, ε)-hiding, if any t-time adversary A achieves advantage*

$$\mathrm{Adv}^{\mathsf{hid}}_{\mathsf{Com}}(A) = 2 \cdot \left| \Pr \left[\begin{array}{l} \mathsf{pk} \leftarrow \mathsf{Gen}, s \leftarrow \{0,1\}, (m_0, m_1, \sigma) \leftarrow A(\mathsf{pk}), \\ (c_s, d_s) \leftarrow \mathsf{Com}_{\mathsf{pk}}(m_s, r) : A(\sigma, c_s) = s \end{array} \right] - \frac{1}{2} \right| \leqslant \varepsilon \ .$$

Two special cases of this property are statistical and perfect hiding. Statistical hiding is (∞, ε)-hiding. This means that if the adversary has infinite computing power, he gets information about the message being committed to with negligible i.e. very, very small probability. Perfect hiding is $(\infty, 0)$-hiding. This means that a commitment to a message reveals no information about about the message, even to an infinitely powerful adversary. We use the term computationally hiding to refer to the cases where t is not infinite.

Definition 2.2. *A commitment scheme is (t, ε)-binding, if any t-time adversary A achieves advantage*

$$\mathrm{Adv}^{\mathsf{bind}}_{\mathsf{Com}}(A) = \Pr \left[\begin{array}{l} \mathsf{pk} \leftarrow \mathsf{Gen}, (c, d_0, d_1, \sigma) \leftarrow A(\mathsf{pk}) : \\ \perp \neq \mathsf{Open}_{\mathsf{pk}}(c, d_0) \neq \mathsf{Open}_{\mathsf{pk}}(c, d_1) \neq \perp \end{array} \right] \leqslant \varepsilon \ .$$

Two special cases of this property are statistical and perfect binding. Statistical binding is (∞, ε)-binding. This means that even if the adversary has infinite computing power, he can cheat with negligible probability. Perfect binding is $(\infty, 0)$-binding. This means that even with infinite computing power, the adversary cannot change his mind after committing to a message. We use the term computationally binding to refer to the cases where t is not infinite.

A commitment scheme cannot be statistically binding and hiding at the same time. If there were such a scheme, then if the sender chooses a random r and sends a commitment $C = \mathsf{Com}(0, r)$, there must exist another random r_0 such that $C = \mathsf{Com}(1, r_0)$. If not, the receiver can conclude that the committed value could not be 1 with quite a high probability, violating *statistical hiding*. But then if the sender has unlimited computing power, he can find that r_0 and change his mind from 0 to 1, violating the *statistical binding* property. The same thing applies to perfect binding and hiding [DN06].

3 Encryption Schemes

The structure of a commitment scheme, is very similar to that of an encryption scheme. The encryption scheme also consists of three parts: key generation Gen, encryption Enc and decryption Dec. The key generation algorithm generates the public and secret key $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$, revealing only the public key to all parties. The encryption algorithm $\mathsf{Enc}_{\mathsf{pk}} : \mathcal{M} \times \mathcal{R} \to \mathcal{E}$ computes the cryptotext e of fixed length from the message $m \in \mathcal{M}$ and the randomness $r \in \mathcal{R}$. The decryption algorithm $\mathsf{Dec}_{\mathsf{sk}} : \mathcal{E} \to \mathcal{M} \cup \{\perp\}$, given the encryption value outputs the message m. If the encryption value is corrupted, the algorithm outputs the abort value $\perp$.

We define two properties of encryption schemes. An encryption scheme can be *indistinguishable under chosen plaintext attack* (IND-CPA). This property defines the security of the encryption scheme against a time-bounded adversary, who outputs two messages and given the encryption of one of the two, has to decide which one has been encrypted. We give a more formal definition of the IND-CPA property [BDPR98].

Definition 3.1. *An encryption scheme is (t, ε)-IND-CPA secure, if any t-time adversary A achieves advantage*

$$\mathrm{Adv}^{\mathsf{ind-cpa}}(A) = 2 \cdot \left| \Pr \left[\begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}, s \leftarrow \{0,1\}, \\ (m_0, m_1, \sigma) \leftarrow A(\mathsf{pk}), \\ e \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_s; r) : A(\sigma, e) = s \end{array} \right] - \frac{1}{2} \right| \leqslant \varepsilon \ .$$

It is easy to see the similarities between this definition and definition 2.1 of the hiding property of commitment schemes.

An encryption schemes can be *Indistinguishable under adaptive chosen ciphertext attack* (IND-CCA2). This property defines the security of the encryption scheme against a time bounded adversary A, that works much like the adversary in definition 3.1, but in addition it has access to the decryption oracle at two stages of the game—first, when outputting the message pair, and second, when trying to determine which of the two messages was encrypted. It is assumed, that A does not ask the oracle to decrypt e. We give a more formal definition of the IND-CCA2 property [BDPR98].

Definition 3.2. *An encryption scheme is (t, ε)-IND-CCA2 secure, if any t-time adversary A achieves advantage*

$$\mathrm{Adv}^{\mathsf{ind-cca2}}(A) = 2 \cdot \left| \Pr \left[\begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}, s \leftarrow \{0,1\}, \\ (m_0, m_1, \sigma) \leftarrow A^{\mathsf{Dec}_{\mathsf{sk}}(\cdot)}(\mathsf{pk}), \\ e \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_s; r) : A^{\mathsf{Dec}_{\mathsf{sk}}(\cdot)}(\sigma, e) = s \end{array} \right] - \frac{1}{2} \right| \leqslant \varepsilon \ ,$$

where $\mathsf{Dec}_{\mathsf{sk}}(\cdot)$ is a decryption oracle.

4 Extractability

The notion of extractable commitments was proposed in the article [SCP00] in the context of non-interactive zero-knowledge proofs. Extractable commitment schemes have an additional property to the usual hiding and binding—if a party knows a certain secret value, they are able to extract the message from the commitment. There is an extra key generation algorithm Gen^* in an extractable commitment scheme. This algorithm outputs the secret key sk in addition to the public key pk. Everything else in the scheme works as before only there exists an additional function. The extraction function $\mathsf{Extr}_{\mathsf{sk}} : \mathcal{C} \to \mathcal{M}$ opens a commitment $c \in \mathcal{C}$ to the original message $m \in \mathcal{M}$. We give the formal definition for extractability [SCP00, Cre02, LAN05].

Definition 4.1. *A commitment scheme is* (t, ε)-extractable *if any* t*-time adversary* A *achieves advantage*

$$\mathrm{Adv}^{\mathrm{extr}}(A) = \Pr\left[\begin{array}{l}(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{Gen}^*, (c,d) \leftarrow A(\mathsf{pk}) : \\ \mathsf{Extr}_{\mathsf{sk}}(c) \neq \mathsf{Open}_{\mathsf{pk}}(c,d) \neq \perp\end{array}\right] \leqslant \varepsilon \ , \tag{1}$$

where the distributions of the public keys pk *output by* Gen *and* Gen^* *coincide.*

Less formally, there is only a negligible chance that a time-bounded adversary A can create such a commitment-decommitment pair (c, d) that the function $\mathsf{Extr}_{\mathsf{sk}}$ extracts a message m from the commitment, while $\mathsf{Open}_{\mathsf{pk}}(c, d)$ outputs a different message m'. The original Gen function is used, when we want to initiate the commitment scheme. The second generation function Gen^* is used, when we want to initiate the scheme that is equivalent to an encryption scheme. The extraction function can only be used in the second case. It is quite simple to see that as soon as the receiver knows sk, the commitment scheme is useless, because the receiver can open it at any time.

Theorem 4.1. *For every reasonable time bound* t*, a* (t, ε)*-extractable commitment scheme is only computationally hiding.*

Proof. Let $\mathcal{C}om = (\mathsf{Gen}, \mathsf{Gen}^*, \mathsf{Com}, \mathsf{Open}, \mathsf{Extr})$ be an extractable commitment scheme then for any t-time adversary A the inequality (1) holds. Let us fix the adversary as the challenger in the hiding game. For some fixed and valid messages $m_0, m_1 \in \mathcal{M}$, the adversary A will be the following

$A(\mathsf{pk})$

$$\left[\begin{array}{l}s \leftarrow \{0,1\} \\ (c,d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(m_s) \\ \mathsf{Return}\ (c,d)\end{array}\right.$$

From the extractability condition and the commitment c that A outputs,

$$\Pr\left[\mathsf{Extr}_{\mathsf{sk}}(c) \neq \mathsf{Open}(c,d) = m_i\right] \leq \varepsilon \ . \tag{2}$$

Now consider an adversary $B = (B_1, B_2)$ who plays the hiding game, finding the secret key and, thus, computing $\mathsf{Extr}_{\mathsf{sk}}(c)$. That adversary achieves advantage

$$\mathrm{Adv}(B) = \Pr\left[\begin{array}{l}\mathsf{pk} \leftarrow \mathsf{Gen}, s \leftarrow \{0,1\}, (m_0, m_1, \sigma) \leftarrow B_1(\mathsf{pk}),\\ c \leftarrow \mathsf{Com}_{\mathsf{pk}}(m_s) : B_2(\sigma, m_s) = s\end{array}\right] > 1 - \varepsilon \ .$$

The only problem that remains is how the adversary B can find the secret key. First, we look at schemes where for any fixed pk there exists exactly one sk. In this case, B simply runs the Gen^* function until he finds the secret key corresponding to the given public key pk. This means that the scheme cannot be statistically hiding, because given enough time, the adversary finds the message with a very high probability.

Secondly, we look at schemes, where for each pk there exist one or more sk. Then for a fixed pk it is possible to choose sk^* that achieves the best error probability against A. This probability can be explicitly computed by generating all possible keys, computing the commitments for m_0 and m_1, and finding the corresponding probabilities. Obviously the inequality (2) must still hold since sk^* can be taken as sk. Hence, the best error probability is not greater than ε. For the same reasons, these schemes are also only computationally hiding. □

Theorem 4.1 implies that the extraction function must be efficient. Since the scheme is computationally hiding, it can be opened if we have enough time, and we do not need the secret key at all. So we do not require a separate inefficient extraction function that needs a secret key. Intuitively we can say that the scheme must be at least statistically binding. Otherwise there exist double openings for at least some commitments and the extraction function could not uniquely open the commitments. Unfortunately, this is not always true [Cre02].

5 Canonical Correspondence

We show how to construct an encryption scheme from a commitment scheme and *vice versa*. In the following, let $\mathcal{C}om = (\mathsf{Gen}_{\mathcal{C}om}, \mathsf{Gen}^*_{\mathcal{C}om}, \mathsf{Com}, \mathsf{Open}, \mathsf{Extr})$ be an extractable commitment scheme and $\mathcal{E}nc = (\mathsf{Gen}_{\mathcal{E}nc}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme. Also, let $m \in \mathcal{M}$ be a message and $r \in \mathcal{R}$ be the used randomness.

First, to construct an encryption scheme from an extractable commitment scheme, we map the functions of $\mathcal{C}om$ to those of $\mathcal{E}nc$. The encryption scheme needs a key generation, an encryption and a decryption function.

We use the key generation algorithm $\mathsf{Gen}^*_{\mathcal{C}om}$ of the commitment scheme as the key generation algorithm $\mathsf{Gen}_{\mathcal{E}nc}$ of the encryption scheme. The $\mathsf{Gen}_{\mathcal{E}nc}$ function outputs the public and secret key pair $(\mathsf{pk}, \mathsf{sk})$ that it gets from $\mathsf{Gen}^*_{\mathcal{C}om}$. To show canonical correspondence between a commitment function and an encryption function, we also need to specify the randomness r used in both schemes. Therefore, we use the functions with specified randomness. The $\mathsf{Enc}(m; r)$ function uses the $\mathsf{Com}_{\mathsf{pk}}(m; r)$ function to encrypt the message m, outputting the

commitment part c from the commitment-decommitment pair (c, d) that it gets from $\mathsf{Com}_{\mathsf{pk}}(m; r)$. The decryption function $\mathsf{Dec}(c)$ uses the extraction function $\mathsf{Extr}_{\mathsf{sk}}(c)$ of the commitment scheme to open the encryption and output the message m. It is straightforward from the Definition 4.1 that the decryption function succeeds with very probability $(1 - \varepsilon)$.

Next, to construct a commitment scheme from an encryption scheme, we map the functions of $\mathcal{E}nc$ to those of $\mathcal{C}om$. An ordinary commitment scheme needs a key generation, a commitment and an opening function. We also add an extraction function and a second key generation algorithm that outputs the key pair. This way the constructed commitment scheme is extractable.

The $\mathsf{Gen}_{\mathcal{E}nc}$ function outputs a key pair $(\mathsf{pk}, \mathsf{sk})$. We only need the public key for the original key generation algorithm $\mathsf{Gen}_{\mathcal{C}om}$ of the commitment scheme. This algorithm takes pk from the pair and discards the rest, as in the commitment scheme scenario, the secret key is not known to anyone. The additional key generation function $\mathsf{Gen}^*_{\mathcal{C}om}$, on the other hand, outputs the whole key pair $(\mathsf{pk}, \mathsf{sk})$ that it received from $\mathsf{Gen}_{\mathcal{E}nc}$. The commitment function $\mathsf{Com}_{\mathsf{pk}}(m; r)$ outputs the commitment-decommitent pair (c, d), where c is the encryption of the message m output by $\mathsf{Enc}(m; r)$, and d simply contains the message m and the used randomness r. The opening function $\mathsf{Open}(c, d)$ recommits to the message m with randomness r and outputs the message m if the commitment it computed matches c; otherwise, the function outputs a special character $\perp$. The extraction function has to open the commitment c without the decommitment value. The function $\mathsf{Extr}_{\mathsf{sk}}(c)$ outputs the message m output by $\mathsf{Dec}(c)$.

The described transformations provide a canonic correspondence between encryption schemes and commitment schemes.

6 IND-CPA Security and Extractability

We will show the duality between the IND-CPA security property of encryption schemes and the computational hiding property of commitment schemes. In the following, let $\mathcal{E}nc = (\mathsf{Gen}_{\mathcal{E}nc}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme and $\mathcal{C}om = (\mathsf{Gen}_{\mathcal{C}om}, \mathsf{Gen}^*_{\mathcal{C}om}, \mathsf{Com}, \mathsf{Open}, \mathsf{Extr})$ be an extractable commitment scheme.

Theorem 6.1. *Let $\mathcal{C}om$ and $\mathcal{E}nc$ be in canonical correspondence. Then (t, ε)-hiding implies (t, ε)-IND-CPA security.*

Proof. We show that the constructed encryption scheme $\mathcal{E}nc$ has the necessary properties, i.e., it is IND-CPA secure. We show that any time-bounded adversary that complies to the assumption that $\mathcal{C}om$ is computationally hiding, is also subject to the IND-CPA security property. To do this, we take an adversary A that plays the hiding game and transform it into an adversary that plays the IND-CPA game. A time-bounded $A = (A_1, A_2)$, playing the hiding game, achieves advantage

$$\mathrm{Adv}(A) = \left| 2 \cdot \Pr \left[\begin{array}{l} \mathsf{pk} \leftarrow \mathsf{Gen}_{\mathcal{C}om}, s \leftarrow \{0,1\}, \\ (m_0, m_1, \sigma) \leftarrow A_1(\mathsf{pk}), \\ (c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(m_s) : A_2(\sigma, c) = s \end{array} \right] - 1 \right| \leq \varepsilon \ .$$

Instead of $\mathsf{Gen}_{\mathcal{C}om}$ used above, the challenger can use $\mathsf{Gen}^*_{\mathcal{C}om}$, since the distributions of public keys output by both algorithms coincide, and just discard the generated secret key from the acquired pair $(\mathsf{pk}, \mathsf{sk})$. In that case, and considering that $\mathcal{C}om$ and $\mathcal{E}nc$ are in canonical correspondence, we can rewrite the advantage as

$$\mathrm{Adv}(A) = \left| 2 \cdot \Pr \left[\begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}_{\mathcal{E}nc}, s \leftarrow \{0,1\}, \\ (m_0, m_1, \sigma) \leftarrow A_1(\mathsf{pk}), \\ c \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_s) : A_2(\sigma, c) = s \end{array} \right] - 1 \right| \leq \varepsilon \ .$$

Hence the corresponding encryption scheme $\mathcal{E}nc$ is (t, ε)-IND-CPA secure. □

Theorem 6.2. *Let $\mathcal{E}nc$ and $\mathcal{C}om$ be in canonical correspondence. Then (t, ε)-IND-CPA security implies (t, ε)- hiding.*

Proof. We show that the constructed commitment scheme $\mathcal{C}om$ has the necessary properties, i.e., it is computationally hiding. We show that any time-bounded adversary that adheres to the assumption that $\mathcal{E}nc$ is IND-CPA secure, is also subject to the computational hiding property. We do this similarly to the proof of the previous theorem—we take an adversary A that plays the IND-CPA game and transform it into an adversary that plays the hiding game. A time-bounded adversary $A = (A_1, A_2)$, playing the IND-CPA game, achieves advantage

$$\mathrm{Adv}(A) = \left| 2 \cdot \Pr \left[\begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}_{\mathcal{E}nc}, s \leftarrow \{0,1\}, \\ (m_0, m_1, \sigma) \leftarrow A_1(\mathsf{pk}), e \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_s) : \\ A_2(\sigma, e) = s \end{array} \right] - 1 \right| \leq \varepsilon \ .$$

Considering the canonical correspondence between $\mathcal{E}nc$ and $\mathcal{C}om$, and as the distributions of the public keys output by $\mathsf{Gen}_{\mathcal{C}om}$ and $\mathsf{Gen}^*_{\mathcal{C}om}$ coincide, we can rewrite the advantage as

$$\mathrm{Adv}(A) = \left| 2 \cdot \Pr \left[\begin{array}{l} \mathsf{pk} \leftarrow \mathsf{Gen}_{\mathcal{C}om}, s \leftarrow \{0,1\}, (m_0, m_1, \sigma) \leftarrow A_1(\mathsf{pk}), \\ (e, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(m_s) : A_2(\sigma, e) = s \end{array} \right] - 1 \right| \leq \varepsilon \ .$$

Hence the corresponding commitment scheme $\mathcal{C}om$ is (t, ε)-hiding. □

7 CCA2 Security and Non-Malleability

Usually, when talking about different protocols, we think about two parties exchanging information. But there is always a chance that a malicious party

might listen to or even interfere with the communication—this is known as the man-in-the-middle attack. The most classical example of this attack concerns ballot boxes. When the votes have been cast and the malicious party gains access to the ballot box, it is very easy for him to include his votes in the box, thus altering the results to his advantage. Note that this does not require the adversary to break the hiding or binding property, just adding a related message is sufficient. This kind of attack is a threat to commitment schemes as well—malleability allows an adversary Ed to alter a commitment received from $Alice$, in a meaningful way so that the receiver Bob cannot make sure whether the commitment is original or it has been tampered with (Figure 1).

$$Alice \xrightarrow{x} Ed \xrightarrow{x+y} Bob$$

Figure 1: Man-in-the-middle attack

Non-malleability is a property that prevents an adversary from making meaningful changes to the messages being passed from one party to the other. Non-malleability with respect to commitment denies the adversary the possibility to create a new commitment from an existing one, whereas non-malleability with respect to opening allows the adversary to make a commitment but not open it. The difference between non-malleability with respect to commitment and with respect to opening was first defined in the article [FF00]. We give both of the definitions here. Although non-malleabiliy w.r.t. commitment is a stronger notion, non-malleability w.r.t. opening has often been considered enough for all practical applications [FF00]. In the following, when we talk about non-malleability, we mean non-malleability w.r.t. commitment, if not specified otherwise.

We give the descriptions of the two non-malleability games in figures. Non-malleability w.r.t. opening can be seen in Fig. 2 and non-malleability w.r.t. commitment is given in Fig. 3. In both of the games, the adversary has to decide which message the commitment was made for. The two games begin similarly—first the key generation algorithm Gen is run to produce the public key pk and the A_1 part of the adversary outputs two messages m_0, m_1 and an internal state σ_1. Next the challenger uniformly chooses a bit s and creates a commitment-decommitment pair for message m_s. The commitment value c from this pair is given along with σ_1 to A_2 that outputs a tuple of commitments $(\hat{c}_1, \ldots, \hat{c}_n)$ and σ_2. At this point the two non-malleability games go their separate ways.

Non-malleability w.r.t. opening means that an adversary, given a commitment is not able to create a correct related commitment that he is able to open himself. In this case the tuple created by A_2 is given along with the decommitment value d and the state σ_2 to A_3 that outputs a tuple of decommitment values $(\hat{d}_1, \ldots, \hat{d}_n)$. The commitments $(\hat{c}_1, \ldots, \hat{c}_n)$ are opened using the decommitments, and a tuple $(y_1, \ldots, y_n)$ is received. If the original commitment c is in the tuple $(\hat{c}_1, \ldots, \hat{c}_n)$ or any of the opened commitments opened to $\perp$, the

$\mathcal{G}^A_{\mathsf{nm-open}}$

$\mathsf{pk} \leftarrow \mathsf{Gen}$

$(m_0, m_1, \sigma_1) \leftarrow A_1(\mathsf{pk})$

$s \leftarrow \{0,1\}$

$(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(m_s)$

$(\hat{c}_1, \ldots, \hat{c}_n, \sigma_2) \leftarrow A_2(c, \sigma_1)$

$(\hat{d}_1, \ldots, \hat{d}_n) \leftarrow A_3(d, \sigma_2)$

$y_i \leftarrow \mathsf{Open}_{\mathsf{pk}}(\hat{c}_i, \hat{d}_i), i = (1, \ldots, n)$

halt if $c \in (\hat{c}_1, \ldots, \hat{c}_n) \vee \perp \in (y_1, \ldots, y_n)$

if $A_4(m_1, y_1, \ldots, y_n, \sigma_2) = s$, return 1

else return 0

Figure 2: The game for non-malleability w.r.t. opening

game is halted, otherwise, the game outputs the decision made by A_4 that is given the message m_1, the tuple $(y_1, \ldots, y_n)$ and the internal value σ_2 as input.

Non-malleability with respect to commitment means that given a commitment, an adversary is not able to create a correct related commitment that can be opened at all. In this case the tuple created by A_2 is given to the extraction oracle Extr that opens them and receives a tuple $(y_1, \ldots, y_n)$. If the original commitment c is in the tuple $(\hat{c}_1, \ldots, \hat{c}_n)$, the game is halted, otherwise, the game outputs the decision made by A_4 that is given the message m_1, the tuple $(y_1, \ldots, y_n)$ and the internal value σ_2 as input.

Definition 7.1. *A commitment scheme is* (t, ε)-non-malleable with respect to decommitment *if any* t*-time adversary* $A = (A_1, A_2, A_3, A_4)$ *playing the game* $\mathcal{G}_{\mathsf{nm-open}}$ *achieves advantage*

$$\mathrm{Adv}^{\mathsf{nm}}_{\mathcal{C}om}(A) = \left|2 \cdot \Pr\left[\mathcal{G}^A = s\right] - 1\right| \leq \varepsilon \ .$$

The following definition [BS99, FF00] is sensible only, if the commitment scheme is either statistically binding or extractable.

Definition 7.2. *A commitment scheme is* (t, ε)-non-malleable with respect to commitment *if any* t*-time adversary* $A = (A_1, A_2, A_4)$ *playing the game* $\mathcal{G}_{\mathsf{nm-com}}$ *achieves advantage*

$$\mathrm{Adv}^{\mathsf{nm}}_{\mathcal{C}om}(A) = \left|2 \cdot \Pr\left[\mathcal{G}^A = s\right] - 1\right| \leq \varepsilon \ ,$$

where Extr *is a computable function such that* $\mathsf{Extr}_{\mathsf{pk}}(c) = x$ *if* $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(x)$.

If the commitment scheme is extractable, we can use the secret key and the oracle $\mathsf{Extr}_{\mathsf{sk}}(\cdot)$ instead of $\mathsf{Extr}_{\mathsf{pk}}(\cdot)$. It is interesting to note that a commitment

$\mathcal{G}^{A}_{\mathsf{nm-com}}$

$\mathsf{pk} \leftarrow \mathsf{Gen}$
$(m_0, m_1, \sigma_1) \leftarrow A_1(\mathsf{pk})$
$s \leftarrow \{0,1\}$
$(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(m_s)$
$(\hat{c}_1, \ldots, \hat{c}_n, \sigma_2) \leftarrow A_2(c, \sigma_1)$
$(y_1, \ldots, y_n) \leftarrow \mathsf{Extr}_{\mathsf{sk}}(\hat{c}_1, \ldots, \hat{c}_n)$
halt if $c \in (\hat{c}_1, \ldots, \hat{c}_n)$
if $A_4(m_1, y_1, \ldots, y_n, \sigma_2) = s$, return 1
else return 0

Figure 3: The game for non-malleability w.r.t. commitment

that is non-malleable with respect to commitment is also non-malleable with respect to opening. When the adversary in the game of non-malleability with respect to commitment has given the commitments to the challenger, he is no longer able to attack in any way, even if he gets to know a backdoor or gets infinite computing power. However, in the case of non-malleability with respect to opening, the adversary can influence the input of A_4 after it has given the commitments to the challenger. But, as mentioned before, non-malleability w.r.t. opening is usually enough in practical applications.

Next, we will show that non-malleability implies hiding and binding. It suffices to show that this is true for the non-malleability property with respect to opening. Non-malleability with respect to commitment implies non-malleability with respect to opening, so it also implies hiding and binding because implication is transitive.

Theorem 7.1. *A commitment scheme that is (t, ε)-non-malleable with respect to opening is also (τ, ε)-hiding, where $\tau = t - \mathcal{O}(1)$.*

Proof. We use proof by contradiction to show that non-malleability w.r.t. opening implies hiding. For the sake of contradiction, assume that the τ-time adversary $B = (B_1, B_2)$ playing the hiding game, achieves advantage

$$\mathrm{Adv}^{\mathsf{hid}}_{\mathsf{Com}}(B) = \left| 2 \cdot \Pr\left[\begin{array}{l} \mathsf{pk} \leftarrow \mathsf{Gen}, s \leftarrow \{0,1\}, (m_0, m_1, \sigma) \leftarrow B_1(\mathsf{pk}), \\ (c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(m_s) : B_2(\sigma, c) = s \end{array} \right] - 1 \right| > \varepsilon \ .$$

Then the adversary $A = (A_1, A_2, A_3, A_4)$ can use the adversary B to win the non-malleability game depicted in Fig. 2. At the beginning of the game, A_1 uses B_1 to output (m_0, m_1, σ). The commitment c of one of these messages and the inner state σ are given to A_2 that passes them on to B_2. Similarly to the hiding game, B_2 now outputs a guess s and succeeds with probability greater

than ε. This guess is put into the state σ_2. Everything works as before, except when A_4 receives σ_2 as part of its input, it no longer needs to output a guess itself, but can simply use the guess from B_2. So the adversary A also succeeds with probability greater than ε, but this contradicts the assumption that the scheme is non-malleable with respect to opening. □

Theorem 7.2. *A commitment scheme that is (t, ε)-non-malleable with respect to opening is also (τ, ε)-binding, where $\tau = t - \mathcal{O}(1)$.*

Proof. We use proof by contradiction to show that non-malleability w.r.t. opening implies binding. For the sake of contradiction, assume that the τ-time adversary B playing the binding game, achieves advantage

$$\mathrm{Adv}^{\mathsf{bind}}_{\mathsf{Com}}(B) = \Pr\left[\begin{array}{l}\mathsf{pk} \leftarrow \mathsf{Gen}, (\hat{c}, \hat{d}_0, \hat{d}_1) \leftarrow B(\mathsf{pk}) : \\ \bot \neq \mathsf{Open}_{\mathsf{pk}}(\hat{c}, \hat{d}_0) \neq \mathsf{Open}_{\mathsf{pk}}(\hat{c}, \hat{d}_1) \neq \bot\end{array}\right] > \varepsilon \ .$$

Then the adversary $A = (A_1, A_2, A_3, A_4)$ can use the adversary B to win the non-malleability game from Fig. 2. The A_1 part of the adversary outputs two messages m_0, m_1 and a state σ_1. Next, A_2 can use B to create a commitment $\hat{c}$ and find a double decommitment $(\hat{d}_0, \hat{d}_1)$ that opens $\hat{c}$ to either y_0 or y_1. The new commitment $\hat{c}$ and the decommitment pair $(\hat{d}_0, \hat{d}_1)$ are enclosed in σ_2. We let A_2 output $\hat{c}, \sigma_2$ when it receives c and σ_1. We know that by definition A_3 has access to the original decommitment value d, but the problem is that it cannot pass on any implicit information to A_4 about d or the message that was committed to. With the help of the adversary B we now have a situation, where A_3 knows which of the two messages the commitment c belongs to, and it is also capable of forwarding the information about this one bit to A_4. To do this, A_3 simply chooses the corresponding decommitment from the pair $(\hat{d}_0, \hat{d}_1)$—it chooses $\hat{d}_0$ when the commitment opens to m_0, and $\hat{d}_1$ otherwise.

It is quite straightforward to see that when the function Open is run on the commitment from A_2 and decommitment from A_3, the result y is either y_0 or y_1. This means that y represents the index of the message that the original commitment c was made to. Now, it is simple for A_4 to output the correct answer. The adversary A succeeds with the same probability as B. This probability, however, is larger than ε and this contradicts the assumption that the scheme is non-malleable with respect to opening. □

We show that IND-CCA2 security implies non-malleability. As discussed before, the commitment scheme needs to be extractable to achieve duality with encryption schemes. In the following theorems, let $\mathcal{E}nc = (\mathsf{Gen}_{\mathcal{E}nc}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme and $\mathcal{C}om = (\mathsf{Gen}_{\mathcal{C}om}, \mathsf{Gen}^*_{\mathcal{C}om}, \mathsf{Com}, \mathsf{Open}, \mathsf{Extr})$ be a commitment scheme.

Theorem 7.3. *Let $\mathcal{E}nc$ and $\mathcal{C}om$ be in canonical correspondence. Then (t, ε)-IND-CCA2 security implies (t, ε)-non-malleability with respect to commitment.*

Proof. Since $\mathcal{E}nc$ and $\mathcal{C}om$ are in canonical correspondence, we can unify the extraction and decryption oracles as $\mathsf{Extr}_{\mathsf{sk}}(\cdot)$. We use proof by contradiction to show that the constructed commitment scheme $\mathcal{C}om$ is non-malleable. For the sake of contradiction, we assume that, the encryption scheme is IND-CCA2 secure but the commitment scheme is not non-malleable. Let $A = (A_1, A_2, A_4)$ be a corresponding adversary that plays the game $\mathcal{G}^A_{\mathsf{nm-com}}$ from Fig. 3 and achieves advantage

$$\mathrm{Adv}^{\mathsf{nm}}_{\mathcal{C}om}(A) = \left|2 \cdot \Pr\left[\mathcal{G}^A = s\right] - 1\right| > \varepsilon \ .$$

Now we show that the scheme cannot be IND-CCA2 secure. We use the adversary A to construct another adversary $B = (B_1, B_2)$, where B_1 is the same as A_1 receiving pk and outputting (m_0, m_1, σ_1) and B_2 is constructed by uniting parts A_2 and A_4 of the adversary A. In the IND-CCA2 game B_2 gets the encryption e and additional information from B_1 in the form of σ_1. This state contains, without loss of generality, the messages m_0 and m_1. By definition, B_2 can use the decryption oracle and ask it to decrypt any encryption he wants except the one he was given, and then has to output its guess about which of the messages was encrypted. We can use parts of the adversary A to execute the described actions. The input for B_2 is given to A_2 that outputs the tuple of commitments $(\hat{c}_1, \ldots, \hat{c}_n)$ that will be given as a query to the extraction oracle $\mathsf{Extr}_{\mathsf{sk}}(\cdot)$, and an internal state σ_2. As mentioned before, the extraction and decryption oracles coincide because of the correspondence between $\mathcal{E}nc$ and $\mathcal{C}om$ and, thus, B_2 can submit the query to $\mathsf{Extr}_{\mathsf{sk}}(\cdot)$ in the IND-CCA2 game. The oracle outputs the tuple of results $(y_1, \ldots, y_n)$ that are given as input to A_4. Now, the output s' of A_4 is also given as the output of B_2. The algorithm for B_2 can compactly be written as

$$\begin{array}{l} B_2(e, \sigma_1) \\ \left[\begin{array}{l} (\hat{c}_1, \ldots, \hat{c}_n, \sigma_2) \leftarrow A_2(e, \sigma_1) \\ (y_1, \ldots, y_n) \leftarrow \mathsf{Extr}_{\mathsf{sk}}(\hat{c}_1, \ldots, \hat{c}_n) \\ \mathsf{Return}\ A_4(m_1, y_1, \ldots, y_n, \sigma_2) \end{array}\right. \end{array}$$

The adversary B plays the IND-CCA2 game. Since B by construction behaves exactly like A, it achieves advantage

$$\mathrm{Adv}^{\mathsf{ind-cca2}}(B) = \left|2 \cdot \Pr\left[\begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}, s \leftarrow \{0,1\}, \\ (m_0, m_1, \sigma_1) \leftarrow B_1(\mathsf{pk}), \\ e \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_s) : B_2^{\mathsf{Extr}_{\mathsf{sk}}(\cdot)}(\sigma_1, e) = s \end{array}\right] - 1\right| > \varepsilon \ .$$

But this contradicts the assumption that the encryption scheme is IND-CCA2 secure. Hence, the commitment scheme must be non-malleable. □

Unfortunately, non-malleability under chosen plaintext attack (NM-CPA) does not imply IND-CCA2 security. On the other hand, it has been proved that non-malleability under chosen ciphertext attack (NM-CCA2) implies IND-CCA2 security and *vice versa* [BS99, DDN91].

8 Conclusion

In this seminar paper we give an overview of the duality of commitment schemes and encryption schemes. It is rather simple to construct a commitment scheme from any encryption scheme, but a commitment scheme must be extractable in order to be the basis for an encryption scheme. Extractability is an additional property of commitment schemes that is used for theoretical constructions and proofs. It uses certain information, that is usually not available, to open commitments without the decommitment value. The extractability property implies that the commitment scheme can only be computationally hiding.

We also show that computational hiding is dual with IND-CPA security and that IND-CCA2 security implies non-malleability.

References

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. In Hugo Krawczyk, editor, *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, pages 26–45. Springer, 1998.

[BS99] Mihir Bellare and Amit Sahai. Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 519–536. Springer, 1999.

[CIO98] Giovanni Di Crescenzo, Yuval Ishai, and Rafail Ostrovsky. Non-Interactive and Non-Malleable Commitment. In *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 141–150, New York, NY, USA, 1998. ACM Press.

[Cre02] Giovanni Di Crescenzo. Equivocable and Extractable Commitment Schemes. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *Security in Communication Networks, Third International Conference, SCN 2002, Amalfi, Italy, September 11-13, 2002. Revised Papers*, volume 2576 of *Lecture Notes in Computer Science*, pages 74–87. Springer, 2002.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-Malleable Cryptography (Extended Abstract). In *Proceedings of the Twenty Third Annual ACM Symposium on Theory of Computing, 6-8 May 1991, New Orleans, Louisiana, USA*, pages 542–552. ACM, 1991.

[DN06] Ivan Damgrd and Jesper Buus Nielsen. Commitment Schemes and Zero-Knowledge Protocols. Course Notes, 2006.

[FF00] Marc Fischlin and Roger Fischlin. Efficient Non-Malleable Commitment Schemes. In Mihir Bellare, editor, *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, pages 413–431, London, UK, 2000. Springer-Verlag.

[LAN05] Sven Laur, N. Asokan, and Kaisa Nyberg. Efficient Mutual Data Authentication Using Manually Authenticated Strings. Cryptology ePrint Archive, Report 2005/424, 2005.

[LN06] Sven Laur and Kaisa Nyberg. Efficient Mutual Data Authentication Using Manually Authenticated Strings. In David Pointcheval, Yi Mu, and Kefei Chen, editors, *Cryptology and Network Security, 5th International Conference, CANS 2006, Suzhou, China, December 8-10, 2006, Proceedings*, volume 4301 of *Lecture Notes in Computer Science*, pages 90–107. Springer, 2006.

[SCP00] Alfredo De Santis, Giovanni Di Crescenzo, and Giuseppe Persiano. Necessary and Sufficient Assumptions for Non-iterative Zero-Knowledge Proofs of Knowledge for All NP Relations. In Ugo Montanari, José D. P. Rolim, and Emo Welzl, editors, *Automata, Languages and Programming, 27th International Colloquium, ICALP 2000, Geneva, Switzerland, July 9-15, 2000, Proceedings*, volume 1853 of *Lecture Notes in Computer Science*, pages 451–462. Springer, 2000.