Soundness of the Formal Model for Active Adversaries

Kevin Donnelly

Much of the content of these notes (in particular most of the definitions) is taken verbatim from [3].

1 Introduction

There are two distinct models that are used to reason about cryptography:

1. The computational model, which we are quite familiar with, assumes messages are bit-strings and adversaries are arbitrary (probabilistic polynomial-time) algorithms. This model uses computational assumptions to prove (usually asymptotic) limits on the probability that an adversary will be successful in some attack. This model is precise and low-level, and is well-suited to proving the security of cryptographic primitives. However, when analyzing cryptographic protocols, such a low-level model is very difficult to use. Reductions from an attack on the protocol to an attack on the cryptographic primitives must be produced by hand for each protocol, and this can be a time-consuming and error-prone task.
2. The formal model, often called the Dolev-Yao model, assumes that messages are terms of some free algebra (i.e. they are just syntax trees) and encryption is merely a term constructor. In this model the adversary is a non-deterministic state machine which may only produce messages that may be obtained by some restricted set of operations from the messages that it already has access to. This model places strong limitations on the adversary, and is not appropriate for reasoning about the security of primitives. However, its simplicity allows for automated analysis of cryptographic protocols.

In 2000, Abadi and Rogaway [1] proved the first result linking security in the formal model with security in the cryptographic model. This original result applied only to shared-key cryptography and used stronger than usual assumptions about the properties of the cryptographic primitives involved. Since this time, the soundness of formal models has been proven for several scenarios and under various assumptions for the cryptographic primitives.

In these notes we primarily summarize Herzog [3]. We prove the result that when encryption primitives are IND-CCA2 secure, indistinguishability in the Dolev-Yao model implies computational indistinguishability, and go on to show that this implies that, when sufficiently strong primitives are used, the Dolev-Yao adversary is not more limited than computational adversary.

2 The Formal Model

In the formal model, messages are merely formal pieces of syntax:

$$
\begin{array}{rll}
\text{(messages) } M \in \mathcal{A} := & I & \text{(identifiers, } I \in \mathcal{I}\text{)} \\
| & R & \text{(nonces, } R \in \mathcal{R}\text{)} \\
| & K_p & \text{(public keys, } K_p \in \mathcal{K}_{Pub}\text{))} \\
| & K_s & \text{(private keys, } K_s \in \mathcal{K}_{Priv}\text{)} \\
| & (M_1, M_2) & \text{(pairing)} \\
| & \{\!|M|\!\}_{K_p} & \text{(encryption of } M \text{ with } K_p\text{)}
\end{array}
$$

We use $K \in (\mathcal{K} = \mathcal{K}_{Pub} \cup \mathcal{K}_{Priv})$ for a K_p or a K_s. There is a bijection $inv : \mathcal{K}_{Pub} \to \mathcal{K}_{Priv}$ which associates a public key with its private key. We write K^{-1} for $inv(K)$ if K is a public key or $inv^{-1}(K)$ if K is a private key.

Example 1. $\left(\{\!|R_1|\!\}_{K_1}, K_1^{-1}\right) \in \mathcal{A}$

Protocols involve honest participants and the adversary. To model the possibility of active attacks, the adversary has complete control over message propagation, honest participants send messages to the adversary and the adversary sends messages back to the participants. Protocols proceed in a sequence of rounds in which the adversary sends a message and receives responses from the environment. Each execution of the protocol is modeled as an alternating sequence of adversary messages ($q_i \in \mathcal{A}$) and participant responses ($r_i \subseteq \mathcal{A}$):

$$r_0\ q_1\ r_1\ q_2\ r_2\ \ldots\ q_{n-1}\ r_{n-1}\ q_n\ r_n$$

The adversary is limited in that q_i must be derivable with a limited set of operations from what it initially knows and from $r_0, \ldots, r_{i-1}$. In the setting of public-key cryptography, the initial knowledge of the adversary consists of: The set of public keys ($\mathcal{K}_{Pub}$), the private keys of cooperating participants ($\mathcal{K}_{Adv}$), the identifiers of the principals ($\mathcal{I}$), and the nonces that the adversary generates ($\mathcal{R}_{Adv}$) which are distinct from the nonces of other participants. The operations that the adversary may use to generate new messages are: decryption of known messages with known keys, encryptions of known messages, pairing of known messages, and separations of known messages.

Definition 1 (Closure). *The* closure *of S, written $C[S]$, is the smallest subset of $\mathcal{A}$ such that:*

1. $S \subseteq C[S]$,
2. $\mathcal{I} \cup \mathcal{K}_{Pub} \cup \mathcal{K}_{Adv} \cup \mathcal{R}_{Adv} \subseteq C[S]$,
3. *If* $\{\!|M|\!\}_{K_p} \in C[S]$ *and* $K_p^{-1} \in C[S]$, *then* $M \in C[S]$,
4. *If* $M \in C[S]$ *and* $K_p \in C[S]$, *then* $\{\!|M|\!\}_{K_p} \in C[S]$,
5. *If* $(M_1, M_2) \in C[S]$, *then* $M_1 \in C[S]$ *and* $M_2 \in C[S]$, *and*
6. *If* $M_1 \in C[S]$ *and* $M_2 \in C[S]$, *then* $(M_1, M_2) \in C[S]$.

Example 2.

$$\begin{array}{l} R_1 \in C[\{\left(\{\!|R_1|\!\}_{K_1}, K_1^{-1}\right)\}], \\ K_1 \in C[\{\{\!|R_1|\!\}_{K_1}\}], \\ R_1 \notin C[\{\{\!|R_1|\!\}_{K_1}\}] \end{array}$$

In the formal model, the adversary is assumed to only be able to produce messages that are in the closure of the set of messages it has received from honest participants.

Definition 2 (Dolev-Yao Adversary). *Suppose that*

$$r_0\ q_1\ r_1\ q_2\ r_2\ \ldots\ q_{n-1}\ r_{n-1}\ q_n\ r_n$$

is a protocol execution in the Dolev-Yao model, then for all i, $q_i \in C[r_0 \cup \ldots \cup r_{i-1}]$.

3 Relating Formal and Computational Adversaries

In order to relate messages in the formal and computational models, we define a meaning function that maps formal messages to probability distributions on bit-strings.

Definition 3 (Semantics of Messages). *Let $\eta \in \mathcal{N}$ be the security parameter. Let $t \in \{0,1\}^\omega$ be a random tape, partitioned into an η-length segment for each nonce and public key in $\mathcal{A}$, and let σ_M be the value of the tape associated with M. Let* D *be an adversary nonce distribution. Let*

$(\mathsf{G}, \mathsf{E}, \mathsf{D})$ *be a public-key cryptography scheme. Then for any* $M \in \mathcal{A}$*, the encoding of* M*, written* $[\![M]\!]_\eta^t$ *is defined recursively by:*

$$
\begin{aligned}
[\![R]\!]_\eta^t &= \begin{cases} \langle \sigma_M, \text{“}nonce\text{”} \rangle & \text{if } R \notin \mathcal{R}_{Adv} \\ \langle \mathtt{D}(\sigma_M), \text{“}nonce\text{”} \rangle & o.w. \end{cases} \\
[\![K_p]\!]_\eta^t &= \langle e, \text{“}pubkey\text{”} \rangle \quad \textit{(where } (K_p, K_p^{-1}) \textit{ is a public/private key pair} \\
[\![K_p^{-1}]\!]_\eta^t &= \langle d, \text{“}privkey\text{”} \rangle \quad \textit{and } (e,d) \textit{ is the output of } \mathsf{G}(1^\eta, \sigma_{K_p})\textit{.)} \\
[\![I]\!]_\eta^t &= \langle m_I, \text{“}id\text{”} \rangle \quad \textit{(} m_I \textit{ is an arbitrary short bit-string uniquely associated to } I\textit{)} \\
[\![(M_1, M_2)]\!]_\eta^t &= \left\langle [\![M_1]\!]_\eta^t, [\![M_2]\!]_\eta^t, \text{“}pair\text{”} \right\rangle \\
[\![\{\!|M|\!\}_K]\!]_\eta^t &= \left\langle \mathsf{E}([\![M]\!]_\eta^t, [\![K]\!]_\eta^t), [\![K]\!]_\eta^t, \text{“}enc\text{”} \right\rangle
\end{aligned}
$$

Note that for fixed t and η, $[\![M]\!]_\eta^t$ remains a distribution, which depends on the coin flips of $(\mathsf{G}, \mathsf{E}, \mathsf{D})$. However, for $X \in (\mathcal{R} \cup \mathcal{K})$ with fixed t and η, $[\![X]\!]_\eta^t$ is a singleton distribution. We briefly explain the intuition behind parts of this definition. The meaning of each message includes a tag that identifies what sort of message it is, so different sorts of messages have disjoint meanings. The meaning of an honest participant's nonce is a random bit-string, while the adversary's nonces come from whatever distribution the adversary chooses with algorithm $\mathtt{D}$. The meaning of a public/private key pair is the public/private key pair generated by G given the randomness assigned to the public key. The meaning of an identity is just an arbitrary (but short and unique) identifying bit-string. The meaning of an encryption, $\{\!|M|\!\}_K$, includes the meaning of K because in general public-key encryption does not hide what key was used to create it.

Encryption also does not generally hide the length of the message that was encrypted. In order to accommodate this fact we will make use of the *type tree* of a message M, denoted by $\mathcal{T}_M$, which is just the same as M with each atomic element replaced by its type (i.e. each R with $\mathcal{R}$, each I with $\mathcal{I}$, each K with $\mathcal{K}$). We assume each message of the same type tree has the same length.

In order to prove that security in the formal model implies security in the computational model, we will need to assume that encryptions are acyclic in the following sense. (This strengthens the definition as given in [3], which is a bit ambiguous.)

Definition 4 (Acyclic). *For an expression* M*, construct a graph* G_M *where the nodes are public/private key pairs (where the public key is used in an encryption in* M*) and there is an edge from* p_1 *to* p_2 *if the private key* K_2^{-1} *(or a term containing* K_2^{-1}*, including encryptions of* K_2^{-1}*) from pair* p_2 *is encrypted with* K_1*, the public key from pair* p_1*.* M *is acyclic if the graph* G_M *is acyclic.*

Example 3. $\left(\{\!|K_1|\!\}_{K_2}, \{\!|K_2|\!\}_{K_1}\right)$ is acyclic, but $\left(\left\{\!\left|K_1^{-1}\right|\!\right\}_{K_2}, \left\{\!\left|K_2^{-1}\right|\!\right\}_{K_1}\right)$ is not.

There has been recent work showing the soundness of the formal model without the acyclicity assumption. However, this result requires a stronger encryption primitive that is key-dependent message (KDM) secure and this is neither implied by nor implies IND-CCA2 security [2].

We can now formalize the computational interpretation of the limitations on the adversary that is imposed by the formal model. Note that we assume that the adversary's knowledge of $\mathcal{I}, \mathcal{K}_{Pub}, \mathcal{K}_{Adv}, \mathcal{R}_{Adv}$ is put into a canonical ordering and represented as a set of oracles, $\mathtt{I}_\eta^t(\cdot)$, $\mathtt{PbK}_\eta^t(\cdot)$, $\mathtt{PrK}_\eta^t(\cdot)$, $\mathtt{R}_\eta^t(\cdot)$, which given a natural number i, return the i^{th} element of the canonical ordering of this knowledge. This is necessary so that the adversary does not get an unreasonable running time (in particular it is possible that its knowledge is infinite).

Definition 5 (Weak Dolev-Yao public-key non-malleability). *The encryption scheme* $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ *provides* weak Dolev-Yao public-key non-malleability *if, when used in* $[\![\cdot]\!]^t_\eta$*:*

$$
\begin{array}{l}
\forall \textit{PPT adversaries } \mathtt{A}, \forall \textit{nonce distributions } \mathtt{D}, \\
\forall \textit{acyclic finite } S \in \mathcal{A}, \forall M \notin C[S], \\
\forall \textit{polynomials } q, \forall \textit{sufficiently large } \eta : \\
\Pr[\ t \leftarrow \{0,1\}^\omega; \\
\quad s \leftarrow [\![S]\!]^t_\eta; \\
\quad m \leftarrow \mathtt{A}^{\mathtt{I}^t_\eta(\cdot),\mathtt{PbK}^t_\eta(\cdot),\mathtt{PrK}^t_\eta(\cdot),\mathtt{R}^t_\eta(\cdot)}(1^\eta, s) : \\
\quad m \in \mathit{supp}[\![M]\!]^t_\eta \qquad\qquad\qquad\qquad\qquad\qquad] \leq \frac{1}{q(\eta)}
\end{array}
$$

where suppD is the support of the distribution D (i.e. the smallest set of points whose complement has zero probability).

This definition captures the computational meaning of Dolev-Yao adversaries because it says that any adversary has a negligible chance of producing any particular message outside of the closure of its knowledge.

4 Dolev-Yao Indistinguishability

Abadi and Rogaway [1] proved that formal messages that are equivalent in a certain sense have computational meanings which are indistinguishable in the usual computational sense. In this section, we state a related result for the public key case. In the next section we will show that this indistinguishability result implies weak Dolev-Yao non-malleability as defined in the previous section.

The following definition of the public-key pattern of a message captures what information can be gained from a particular message. Let M be an arbitrary message. Note that M may contain sub-terms that are encrypted and thus hidden from an adversary ($\{\!|M'|\!\}_K$). Let T be a set of public keys. We will define $\mathit{pattern}_{pk}(M,T)$ to have the following property. Given a set of public keys T, the set of sub-terms of $\mathit{pattern}_{pk}(M,T)$ is exactly the set of things that an adversary learns from M, if he is able to decrypt messages encrypted with public keys in T.

Definition 6 (Public-key pattern). *Let* $T \subseteq \mathcal{K}_{Pub}$*. We recursively define the function* $p(M,T)$ *by:*

$$
\begin{array}{rl}
p(K,T) = K & \textit{if } K \in \mathcal{K} \\
p(A,T) = A & \textit{if } A \in \mathcal{I} \\
p(R,T) = R & \textit{if } R \in \mathcal{R} \\
p((M_1,M_2),T) = (p(M_1,T), p(M_2,T)) & \\
p(\{\!|M|\!\}_K, T) = \begin{cases} \{\!|p(M,T)|\!\}_K & \textit{if } K \in T \\ \langle\!|\mathcal{T}_M|\!\rangle_K & \textit{o.w. (where } \mathcal{T}_M \textit{ is the type tree of } M) \end{cases} &
\end{array}
$$

Then we define $\mathit{pattern}_{pk}(M,T) = p(M, (\mathcal{K}_{Priv} \cap C[\{M\} \cup T^{-1}])^{-1})$*. (This definition is a corrected version of the definition given in [3], which has a small bug). We also write* $\mathit{pattern}_{pk}(M)$ *for* $\mathit{pattern}_{pk}(M, \emptyset)$*.*

Example 4.

$$
\begin{array}{rl}
\mathit{pattern}_{pk}(\left(R_1, \{\!|R_2|\!\}_{K_1}\right)) & = \left(R_1, \langle\!|\mathcal{R}|\!\rangle_{K_1}\right) \\
\mathit{pattern}_{pk}(\left(R_1, \{\!|R_2|\!\}_{K_1}\right), K_1) & = \left(R_1, \{\!|R_2|\!\}_{K_1}\right) \\
\mathit{pattern}_{pk}(\left(K_1^{-1}, \{\!|R_2|\!\}_{K_1}\right)) & = \left(K_1^{-1}, \{\!|R_2|\!\}_{K_1}\right)
\end{array}
$$

The following definition of ingredient, is the same as the usual formal notion of sub-term.

Definition 7 (Ingredient). *If M, M' are two patterns, then M is an* ingredient *of M', written $M \sqsubseteq M'$, if the parse tree of M is a sub-tree of the parse tree of M'.*

Example 5. $(R_1, R_2) \sqsubseteq (\{\!|(R_1, R_2)|\!\}_K, I)$

Note that messages are also patterns so this definition may be applied to messages as well. The following theorem shows that the definition for public-key pattern has the desired property.

Theorem 1. *If M, M' are messages and $M' \sqsubseteq pattern_{pk}(M)$, then $M' \in C[M]$.*

Definition 8 (Semantics of Patterns). *Let:*

- $[\![\langle\!| \mathcal{T}_M |\!\rangle]\!]^t_\eta = m_M$, *where m_M is any fixed bit-string of length $\left|[\![M]\!]^t_\eta\right|$, such as the all-zero string, and*
- $[\![\langle\!| \mathcal{T}_M |\!\rangle_K]\!]^t_\eta = \left\langle \mathsf{E}([\![\mathcal{T}_M]\!]^t_\eta, [\![K]\!]^t_\eta), [\![K]\!]^t_\eta, \text{“}enc\text{”} \right\rangle$.

The following definition is slightly more general than the usual notion of computational indistinguishability in that it allows for an oracle.

Definition 9 (Computational indistinguishability). *Suppose that $\{D_\eta\}_\eta$ and $\left\{D'_\eta\right\}_\eta$ are two families of distributions indexed by the security parameter. Then they are* computationally indistinguishable with respect to a family of oracles $\mathtt{O}_x$, *written $D_\eta \cong_{\mathtt{O}_x} D'_\eta$, if*

$$\begin{array}{l} \forall PPT \text{ adversaries } \mathtt{A}, \forall \text{polynomials } q, \forall \text{sufficiently large } \eta : \\ \left|\Pr[d \leftarrow D_\eta : 1 \leftarrow \mathtt{A}^{\mathtt{O}_x(\cdot)}(d, \eta)] - \Pr[d \leftarrow D'_\eta : 1 \leftarrow \mathtt{A}^{\mathtt{O}_x(\cdot)}(d, \eta)]\right| \leq \frac{1}{q(\eta)} \end{array}$$

We will prove that if IND-CCA2 secure encryption is used, then for any formal message M, $[\![M]\!]^t_\eta \cong_{\mathtt{O}_x} [\![pattern_{pk}(M)]\!]^t_\eta$. We will provide oracles that correspond to the oracles in the IND-CCA2 game. The following definition specifies the keys with respect to which the oracle will decrypt:

Definition 10. *Let M be a pattern then $M|_{\mathcal{K}_{Pub}} = \{K \in \mathcal{K}_{Pub} : K \sqsubseteq M\}$. If S is a set of messages then $S|_{\mathcal{K}_{Pub}} = \{K \in \mathcal{K}_{Pub} : \exists M \in S.K \sqsubseteq M\}$.*

We must also specify the challenge ciphertexts which the oracle will not decrypt. These are the ciphertexts which are different between $[\![M]\!]^t_\eta$ and $[\![pattern_{pk}(M)]\!]^t_\eta$.

Definition 11 (Visible). *Let σ be a bit-string and τ a set of computational public keys. Then let $vis_\tau(\sigma)$ be the smallest set such that:*

- $\sigma \in vis_\tau(\sigma)$,
- *if $\langle a, b, \text{“}pair\text{”}\rangle \in vis_\tau(\sigma)$, then $a \in vis_\tau(\sigma)$ and $b \in vis_\tau(\sigma)$,*
- *if $\langle c, k, \text{“}enc\text{”}\rangle \in vis_\tau(\sigma)$, $k \in \tau$, and k' is the secret key corresponding to k, then $\mathsf{D}(c, k') \in vis_\tau(\sigma)$, and*
- *if $\langle c, k, \text{“}enc\text{”}\rangle \in vis_\tau(\sigma)$, $\langle k', \text{“}privkey\text{”}\rangle \in vis_\tau(\sigma)$, and k' is the secret key corresponding to k, then $\mathsf{D}(c, k') \in vis_\tau(\sigma)$.*

A bit-string m is a visible element *in σ relative to τ if $m \in vis_\tau(\sigma)$.*

Intuitively if x is an encoding of X, σ is an encoding of M, τ is an encoding of T then $x \in vis_\tau(\sigma)$ iff $X \sqsubseteq pattern_{pk}(M, T)$. The decryption oracle we provide will not decrypt with respect to any $x \in vis_\tau(\sigma)$. The following is a definition for formal indistinguishability, which closely mirrors that given for the symmetric case by Abadi and Rogaway[1].

Definition 12 (Abadi-Rogaway public-key indistinguishability). *An encryption scheme,* $(\mathsf{G}, \mathsf{E}, \mathsf{D})$*, provides* Abadi-Rogaway public-key indistinguishability *if, when used in* $[\![\cdot]\!]^t_\eta$*, for all nonce distributions* D*, acyclic formal messages* M*, and finite* $T \subseteq \mathcal{K}_{Pub}$*:*

$$[\![M]\!]^t_\eta \cong_{\mathsf{O}^{M,T}_x} [\![pattern_{pk}(M,T)]\!]^t_\eta$$

where $\mathsf{O}^{M,T}_x(\sigma, \mathsf{pk})$ *returns* $\perp$ *unless* pk *is a valid public key and*

- *either* $\mathsf{pk} \in [\![K]\!]^t_\eta$ *for some* $K \in T$*, or*
- $\mathsf{pk} \in [\![K]\!]^t_\eta$ *for some* $K \in (M|_{\mathcal{K}_{Pub}} \setminus T)$ *and* $\sigma \notin vis_{[\![T]\!]^t_\eta}(x)$*.*

This definition basically says that an adversary cannot distinguish between

$$[\![M]\!]^t_\eta \text{ and } [\![pattern_{pk}(M,T)]\!]^t_\eta$$

when given an oracle that decrypts any messages encrypted with a key in $[\![T]\!]^t_\eta$ and decrypts any ciphertexts that are not derived from $[\![M]\!]^t_\eta$ that are encrypted with any key used in M. Note that for $M = \{\!|R|\!\}_K$, we have $[\![\{\!|R|\!\}_K]\!]^t_\eta \cong_{\mathsf{O}^{M,T}_x} [\![\{\!|Z|\!\}_K]\!]^t_\eta$ (where Z is a message such that $[\![Z]\!]^t_\eta$ is an η length bit-string of 0's) which implies that $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ is IND-CCA2 secure.

We can now use this definition to prove a soundness result relating security in the formal model with security in the computational model. Formally, we consider any message M equivalent to its pattern $pattern_{pk}(M, T_{Adv})$ where T_{Adv} is the set of public keys for which the adversary knows the corresponding private key. This notion of equivalence can be used to deductively *prove* that an adversary cannot gain certain information by observing a cryptographic protocol (even given powerful oracles). The following result says that this reasoning is valid if the public-key encryption used in these protocols is IND-CCA2 secure.

Theorem 2. *If* $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ *is IND-CCA2 secure then* $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ *provides Abadi-Rogaway public-key indistinguishability.*

Proof. (This is a corrected version of the proof in [3], which contains several errors.) Suppose that $[\![\cdot]\!]^t_\eta$ *uses an encryption scheme* $(\mathsf{G}, \mathsf{E}, \mathsf{D})$*, and there is a nonce distribution* $\mathtt{D}$*, a message* M*, and set of keys* T *and a PPT adversary* $\mathtt{A}$ *that can distinguish* $[\![M]\!]^t_\eta$ *from* $[\![pattern_{pk}(M,T)]\!]^t_\eta$ *when given access to the oracle,* $\mathsf{O}^{M,T}_x$*, from Definition 12. Then* $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ *is not IND-CCA2 secure.*

Since M *is acyclic, we can order the key-pairs in the parse tree of* M *as* $K_1, K_2, ..., K_k$ *so that if* $K_i \rightarrow K_j$ *in the graph* G_M*, then* $i \geq j$*. So the deeper the key in encryptions, the smaller the number. We construct a series of intermediate patterns,* $M_0, ..., M_k$ *between* M *and* $pattern_{pk}(M,T)$*:*

$$M_0 = M = pattern_{pk}(M, T \cup \{K_1, K_2, ..., K_k\})$$
$$M_k = pattern_{pk}(M,T)$$

M_{i+1} *is the same as* M_i *except that any encryptions with key* K_{k-i+1} *are blobbed, if the equivalent blob appears in* $pattern_{pk}(M,T)$*. This is almost the same as saying* $M_i = pattern_{pk}(M, T \cup \{K_1, K_2, ..., K_{k-i}\})$ *except that we only create blobs that appear at the same position in* $pattern_{pk}(M,T)$*. Setting up the hybrids this way (rather than the way proposed in [3]) is necessary to ensure that* $\mathtt{A}$ *can distinguish some "top-level" encryption from its blob, which is necessary in order to be able to properly answer oracle queries. Note that, because of the ordering on keys, whenever an encryption by key* K_i *has been blobbed, any encryptions of key* K_i^{-1} *have already been blobbed, this is*

necessary so that we can assume that $[\![K_i]\!]^t_\eta$ *is the given public key (for which we do not know the corresponding private key).*

Since $[\![M]\!]^t_\eta \not\cong_{\mathtt{O}_x^{M,T}} [\![pattern_{pk}(M,T)]\!]^t_\eta$, *there must be* i *s.t.* $[\![M_i]\!]^t_\eta \not\cong_{\mathtt{O}_x^{M,T}} [\![M_{i+1}]\!]^t_\eta$. *If there were* $n > 1$ *encryptions blobbed between* M_i *and* M_{i+1}, *we construct a further* n *hybrids,* $N_1, ..., N_n$ *with one further encryption being blobbed in each. There must be some* j *with* $[\![N_j]\!]^t_\eta \not\cong_{\mathtt{O}_x^{M,T}} [\![N_{j+1}]\!]^t_\eta$, *and let* $E = \{\!|P|\!\}_{K_i}$ *be the encryption changed into* $\langle\!|\mathcal{T}_P|\!\rangle_{K_i}$ *at this step.*

We can use $\mathtt{A}$ *to distinguish between* m_0 *and* m_1 *encrypted under key* pk *by first choosing a fresh random take* t *and setting:*

$$m_0 \leftarrow [\![P]\!]^t_\eta$$
$$m_1 \leftarrow [\![\langle\!|\mathcal{T}_P|\!\rangle]\!]^t_\eta$$

And treating pk *as the value of* K_i. *Note that* $[\![K_i]\!]^t_\eta$ *is a singleton distribution for fixed* η *and* t.

We get as input some public key pk. *We select a fresh random tape* t, *and sample* $p \leftarrow [\![P]\!]^t_\eta[\mathsf{pk}/K_i]$ *(where* $[\![M]\!]^t_\eta[x/X]$ *means* $[\![M]\!]^t_\eta$ *where we use* c *as* $[\![X]\!]^t_\eta$*). It is fine to assume* $[\![K_i]\!]^t_\eta = \{\mathsf{pk}\}$ *because we know* pk *was generated by* G *and the acyclicity of* M *guarantees that* K^{-1} *does not appear in* P *so we do not need to know the private key corresponding to* pk *in order to sample* $[\![P]\!]^t_\eta$. *Return* p *and* $[\![\langle\!|\mathcal{T}_P|\!\rangle]\!]^t_\eta$ *as candidate plaintexts.*

On input c, *we sample* $s \leftarrow [\![N_j]\!]^t_\eta[c/P, \mathsf{pk}/K_i]$. *(Note that this is okay, because the ordering on keys guarantees us that* N_j *does not depend on* K_i^{-1}*). If* c *is an encryption of* p *then* s *comes from the same distribution as* N_j *and if* c *in an encryption of* $[\![\langle\!|\mathcal{T}_P|\!\rangle]\!]^t_\eta$ *then* s *comes from the same distribution as* N_{j+1}. *We proceed by running* $\mathtt{A}(s, 1^\eta)$*:*

- *If* $\mathtt{A}$ *makes an oracle query on* (σ, pk), *we check that* $\{\mathsf{pk}\} = [\![K]\!]^t_\eta$ *for some* $K \in M|_{\mathcal{K}_{Pub}} \cup T$. *If not we return* $\perp$, *otherwise:*
  - *If* $K = K_i$, *we check that* σ *is not visible in* s *relative to* $\tau = [\![T]\!]^t_\eta$. *If it is, we return* $\perp$ *as the oracle is supposed to. Otherwise, since* σ *is not visible in* s *relative to* τ *and* c *came from a top-level encryption/blob so it is visible in* s *relative to* τ, *we know* $\sigma \neq c$. *Hence we can use the CCA-2 decryption oracle to decrypt* σ.
  - *If* $K \neq K_i$, *we can produce* $[\![K^{-1}]\!]^t_\eta$ *ourselves from the take* t. *If* $K \in T$, *we decrypt* σ *using* $[\![K^{-1}]\!]^t_\eta$. *If* $K \in M|_{\mathcal{K}_{Pub}} \setminus T$, *we also check if* σ *is visible in* s *relative to* $\tau = [\![T]\!]^t_\eta$. *We return* $\perp$ *if it is, and decrypt* σ *otherwise.*

Corollary 1. *Suppose that* M, N *are two acyclic messages,* $T \subseteq \mathcal{K}_{Pub}$, *and* $M|_{\mathcal{K}_{Pub}} = N|_{\mathcal{K}_{Pub}}$. *If* $pattern_{pk}(M,T) = pattern_{pk}(N,T)$ *and the encoding operation* $[\![\cdot]\!]^t_\eta$ *uses an IND-CCA2 secure encryption scheme, then for any nonce distribution* $\mathtt{D}$, $[\![M]\!]^t_\eta \cong_{\mathtt{O}_x^{M,T}} [\![N]\!]^t_\eta$.

It turns out that Theorem 2 implies that the Dolev-Yao abstraction of an active adversary is not as limited as it first may seem. Because the Dolev-Yao indistinguishability definition allows for a strong oracle to be used to try to decrypt, it requires that the encryption used be IND-CCA2 secure, which means it is also NM-CCA2 secure. If the encryption is non-malleable than it is just as opaque to the computational adversary as to the Dolev-Yao adversary.

Theorem 3. *Suppose that* $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ *is a computational public-key encryption scheme that provides Abadi-Rogaway public-key indistinguishability. Then* $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ *provides weak Dolev-Yao public-key non-malleability.*

Proof. By contradiction.

Suppose there is an adversary that can produce a message outside the closure of its input set:

$$\begin{array}{l}
\exists PPT \text{ adversary } \mathtt{A}, \exists \text{nonce distributions } \mathtt{D}, \\
\exists \text{acyclic finite } S \in \mathcal{A}, \exists M \notin C[S], \\
\exists \text{polynomial } q, \text{for infinitely large } \eta : \\
\Pr[\ t \leftarrow \{0,1\}^{\omega}; \\
\quad s \leftarrow [\![S]\!]^t_\eta; \\
\quad m \leftarrow \mathtt{A}^{\mathtt{I}^t_\eta(\cdot),\mathtt{PbK}^t_\eta(\cdot),\mathtt{PrK}^t_\eta(\cdot),\mathtt{R}^t_\eta(\cdot)}(1^\eta, s) : \\
\quad m \in supp[\![M]\!]^t_\eta \qquad\qquad\qquad\qquad] \geq \frac{1}{q(\eta)}
\end{array}$$

We can construct $\mathtt{A}_1$ *that contradicts Abadi-Rogaway public-key indistinguishability of* $(\mathsf{G}, \mathsf{E}, \mathsf{D})$ *(recall Definition 12). Consider the parse tree of* M*. There must be some leaf,* M_l*, that is not in* $C[S]$ *or else* M *itself would be in* $C[S]$ *(since* $C[S]$ *is closed under pairing and encryption which are the only non-leaf constructors).*

The essence of the argument is aw follows. A *must be able to produce* $m \in supp[\![M_l]\!]^t_\eta$ *and it must be that* M_l *is in* $\mathcal{R} \setminus \mathcal{R}_{Adv}$ *or* $\mathcal{K}_{Priv} \setminus \mathcal{K}_{Adv}$*. We can then construct* S' *as:*

$$S' = \begin{cases} S \cup \{M_l\} & \text{if } M_l \in \mathcal{R} \setminus \mathcal{R}_{Adv} \\ S \cup \left\{N_p, \{\!|N_p|\!\}_{M_l^{-1}}\right\} & \text{(where } N_p \in \mathcal{R}_{Adv}\text{) if } M_l \in \mathcal{K}_{Priv} \setminus \mathcal{K}_{Adv} \end{cases}$$

$\mathtt{A}_1$ *can then use* A *to distinguish* $[\![S']\!]^t_\eta$ *from* $[\![pattern_{pk}(S')]\!]^t_\eta$*. There are a few more subtleties to deal with, see [3] for details.*

Theorem 3 shows that it is valid to consider adversaries in the Dolev-Yao model to encompass all possible adversaries if IND-CCA2 secure encryption is used. This result means that it is valid to enumerate all possible protocol runs in the Dolev-Yao model, and if each of these possible runs satisfies some desired security property, then in the computational model all actual protocol runs will satisfy this property as well.

References

1. Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In *TCS '00: Proceedings of the International Conference IFIP on Theoretical Computer Science, Exploring New Frontiers of Theoretical Informatics*, pages 3–22, London, UK, 2000. Springer-Verlag.
2. Pedro Adao, Gergei Bana, Jonathan Herzog, and Andre Scedrov. Soundness of formal encryption in the presence of key-cycles. In *Proceedings of the 10th European Symposium On Research In Computer Security (ESORICS 2005)*, September 2005.
3. Jonathan Herzog. A computational interpretation of Dolev-Yao adversaries. *Theoretical Computer Science*, 340:57–81, June 2005.