Unité Mixte de Recherche 5104 CNRS - INPG - UJF
Centre Equation
2, avenue de VIGNATE
F-38610 GIERES
tel : +33 456 52 03 40
fax : +33 456 52 03 50
http://www-verimag.imag.fr

Probabilistic Opacity for a Passive Adversary and its Application to Chaum's Voting Scheme

Yassine Lakhnech and Laurent Mazaré

Report n^o 4

February 15, 2005

Probabilistic Opacity for a Passive Adversary and its Application to Chaum's Voting Scheme

Yassine Lakhnech and Laurent Mazaré

VERIMAG
2, av. de Vignates, 38610 Gières - FRANCE
{yassine.lakhnech,laurent.mazare}@imag.fr

February 15, 2005

Abstract

A predicate is opaque for a given system, if an adversary will never be able to establish truth or falsehood of the predicate for any observed computation. This notion has been essentially introduced and studied in the context of transition systems whether describing the semantics of programs, security protocols or other systems. In this paper, we are interested in studying opacity in the probabilistic computational world. Indeed, in other settings, as in the Dolev-Yao model for instance, even if an adversary is 99% sure of the truth of the predicate, it remains opaque as the adversary cannot conclude for sure. In this paper, we introduce a computational version of opacity in the case of passive adversaries called cryptographic opacity. Our main result is a composition theorem: if a system is secure in an abstract formalism and the cryptographic primitives used to implement it are secure, then this system is secure in a computational formalism. Security of the abstract system is the usual opacity and security of the cryptographic primitives is IND-CPA security. To illustrate our result, we give two applications: a short and elegant proof of the classical Abadi-Rogaway result and the first computational proof of Chaum's visual electronic voting scheme.

Keywords: Opacity, Non-Interference, Chaum's Voting Scheme, Computational Model, Probabilistic Encryption.

Reviewers:

How to cite this report:

@techreport { verimag-TR-2005-4,
title = { Probabilistic Opacity for a Passive Adversary and its Application to Chaum's Voting Scheme},
authors = { Yassine Lakhnech and Laurent Mazaré},
institution = { Verimag Technical Report },
number = {4},
year = { 2005},
note = { }
}

predicate is opaque for a given system, if an adversary will never be able to establish truth or falsehood of the predicate for any observed computation. This notion has been essentially introduced and studied in the context of transition systems whether describing the semantics of programs, security protocols or other systems. In this paper, we are interested in studying opacity in the probabilistic computational world. Indeed, in other settings, as in the Dolev-Yao model for instance, even if an adversary is 99% sure of the truth of the predicate, it remains opaque as the adversary cannot conclude for sure. In this paper, we introduce a computational version of opacity in the case of passive adversaries called cryptographic opacity. Our main result is a composition theorem: if a system is secure in an abstract formalism and the cryptographic primitives used to implement it are secure, then this system is secure in a computational formalism. Security of the abstract system is the usual opacity and security of the cryptographic primitives is IND-CPA security. To illustrate our result, we give two applications: a short and elegant proof of the classical Abadi-Rogaway result and the first computational proof of Chaum's visual electronic voting scheme.

Keywords: Opacity, Non-Interference, Chaum's Voting Scheme, Computational Model, Probabilistic Encryption.

Introduction

Roughly speaking, a predicate is opaque for a given system, if an adversary will never be able to establish truth or falsehood of the predicate, for any observed execution of the system. It is clear that this notion only makes sense when the adversary does not have access to the complete state of the system but rather accesses its execution through an observation function. Typically, the predicate of interest concerns some non-determinism that is resolved initially such as the votes in a voting scheme. This notion has been essentially introduced and studied in the context of transition systems whether describing the semantics of programs, security protocols or other systems. It generalizes well-known security properties such as anonymity and non-interference (See [7] for a discussion).

Opacity has been essentially studied in the so-called formal [1] world of security, where cryptography is assumed perfect. A typical formal model, for security protocols for instance, is the Dolev and Yao model [11] where messages are described by algebraic terms and there is one single adversary that subsumes all possible attacks. In this paper, we are interested in studying opacity in the probabilistic computational world. Indeed, in the formal world, even if an adversary is 99% sure of the truth of the predicate, it remains opaque as the adversary cannot conclude for sure. Such a definition is clearly not useful in the probabilistic computational setting. Therefor, we introduce a computational version of opacity. We restrict ourselves to the case of passive adversaries and call our security notion *cryptographic opacity*. More generally, we introduce probabilistic opacity that includes : strict opacity, plausible deniability and cryptographic opacity. All three notions are defined by experiments and in terms of the advantage of the adversaries. Strict opacity requires that the advantage is null; plausible deniability requires that the probability to win the experiment is different from 1 and cryptographic opacity requires that the advantage is negligible.

Then, the question of how to prove probabilistic cryptographic opacity rises. For strict opacity, we show a decidability result for finite systems. The main core of the paper, however, deals with cryptographic opacity. Our answer to this question is inspired by a recent trend started by [2] and pushed further in [4, 18, 15, 10] in bridging the gap that separates the Dolev-Yao model and its perfect cryptography assumption on one hand and the computational model on the other hand. Indeed, we prove our main result that states the following: if a predicate is opaque in the formal model for an abstraction of the considered system and if the cryptographic primitives are IND-CPA, then cryptographic opacity of the predicate holds in the computation model. The previously mentioned results on the relationship between the formal and the computational models do not immediately apply in our case as we have to carefully deal with random coins used for encryption. Indeed, in the case of Chaum's voting scheme, for instance, some of these coins appear as plain-text. On the other hand, for some IND-CPA schemes, e.g. [6], knowledge of the random coins induces knowledge about the encrypted message. To deal with this problem, we introduce a new security criterion, called n-RPAT-CPA. We then show that any IND-CPA secure cryptographic scheme is also n-RPAT-CPA secure.

An other important contribution of our paper is the proof of opacity for Chaum's visual voting scheme [8]. This is done by applying our main result. We also give a sort proof of Abadi and Rogaway's result as an application of our main theorem.

Related work. The initial work of Abadi and Rogaway was pushed further in [1]. This last paper considers systems with cryptographic primitives and studies indistinguishability, but this property lacks the generality of opacity. Another interesting work is [3] which links a computational version of probabilistic non-interference [14] to the notion of simulatability. This work is very general as it considers active adversaries but as a consequence, their main theorem is more difficult to apply. In [17], a computational definition of indistinguishability (or strong secrecy) is given. This security notion is less general than opacity. Laud also formulates an analysis allowing verification of programs using cryptographic primitives but these primitives are still abstracted.

This paper is structured as follows. The first section recalls some necessary preliminaries. The second section introduces strict opacity, plausible deniability and cryptographic opacity . The following section proves a decidability result for strict opacity. Section 4 presents an approach to verification of cryptographic opacity. This approach is applied to Chaum's voting scheme in Section 5. The proof of main result and the security criterion n-RPAT-CPA are discussed in Section 6.

[1] One might prefer the word symbolic here since formal is not used in the sense of rigorous.

1 Preliminaries

In this section, we recall some basic definitions that are useful when considering probabilistic systems and introduce a general definition of security criteria along with a decomposition theorem that is used later in the paper. All these notions are detailed in [16].

1.1 Cryptographic Primitives

Let η be the security parameter of the system, it characterizes the strength of the cryptographic primitives as well as the length of nonces.

An *asymmetric encryption scheme* $\mathcal{AE} = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$ is defined by three algorithms. The key generation algorithm $\mathcal{KG}$ is a randomized function which given a security parameter η outputs a pair of keys (pk, sk), where pk is a public key and sk the associated secret key. The encryption algorithm $\mathcal{E}$ is also a randomized function which given a message and a public key outputs the encryption of the message by the public key. The random part is explicitly represented as a nonce (bit-string of length η) bs which is given as argument to $\mathcal{E}$. Finally the decryption algorithm $\mathcal{D}$ takes as input a secret key and a cypher-text and outputs the corresponding plain-text, i.e., $\mathcal{D}(\mathcal{E}(m, pk, bs), sk) = m$ for any bs and any pair (pk, sk) produced by the key generation algorithm. The execution time of the three algorithms is assumed polynomially bounded by η.

A function $g : \mathbb{R} \to \mathbb{R}$ is *negligible*, if it is ultimately bounded by x^{-c}, for each positive $c \in \mathbb{N}$, i.e., for all $c > 0$ there exists N_c such that $|g(x)| < x^{-c}$, for all $x > N_c$.

1.2 Security Criteria

Security criteria define the correctness (or the expected properties) of an encryption scheme. They are defined as an experiment involving an adversary. Given access to a set of oracles, the adversary has to guess a randomly chosen data. [2] Roughly speaking, a scheme is safe w.r.t. a given criterion, if no adversary has a better probability to win than an adversary who does not have access to the oracles. Therefor, the strength of a criterion depends on the allowed adversaries and the offered oracles. In this paper, we consider adversaries that are terminating random Turing machines (RTM) or polynomial-time random TM (PRTM) when considering computational encryption. The time is bounded in the security parameter η. An RTM $\mathcal{B}$ is said to have a complexity *similar* to the complexity of $\mathcal{A}$ if the execution of $\mathcal{B}$ is polynomially bounded in the (maximum) execution duration of $\mathcal{A}$.

Let us now define formally *security criteria*. A criterion γ is a triple $(\Theta; F; V)$ where

- Θ is a (P)RTM that randomly generates some challenge θ (for example, a bit b and a pair of key (pk, sk)).
- F is a (P)RTM that takes as arguments a string of bits s and a challenge θ and outputs a new string of bits. F represents the oracles that an adversary can call to solve its challenge.
- V is a (P)RTM that takes as arguments a string of bits s and a challenge θ and outputs either true or false. It represents the verification made on the result computed by the adversary. The answer true (resp. false) means that the adversary solved (resp. did not solve) the challenge.

Note that Θ can generate an arbitrary number of parameters and F can represent an arbitrary number of oracles. Thus, it is possible to define criteria with multiples Θ and F. When no confusion may rise, we use the same notation for the challenge generator Θ and the generated challenge θ (both are denoted using θ).

A criterion $(\Theta; F; V)$ and adversary $\mathcal{A}$ produce the following experiment. First θ is generated randomly. The adversary can now make some computation using the oracle F, this is denoted by $\mathcal{A}/F$. The behavior of the oracle depends on θ. At the end of computation, the adversary has to return a string of bits which is verified by an algorithm V. Also V uses θ (e.g. θ includes a bit b and the adversary has to output the value of b). The aim of the adversary $\mathcal{A}$ is produce a bit string that is verified by V. More formally, the experiment $\mathbf{Exp}^{\gamma}_{\mathcal{A}}(\eta)$ involving $\mathcal{A}$ and γ is defined by the following Turing machine:

[2] In some cases, as for symmetric encryption or signature, the adversary has other ways to win. This is not relevant for this paper.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\gamma}(\eta)$:	**Experiment** $\mathbf{Exp'}_{\mathcal{A}}^{\gamma}(\eta)$:
$\theta \leftarrow \Theta(\eta)$	$\theta \leftarrow \Theta(\eta)$
$d \leftarrow \mathcal{A}/\eta, \lambda s.F(s,\theta)$	$d \leftarrow \mathcal{A}/\eta$
return $V(d,\theta)$	**return** $V(d,\theta)$

We can now define the advantage of $\mathcal{A}$ against γ as follows:

$$\mathbf{Adv}_{\mathcal{A}}^{\gamma}(\eta) = 2.\big(pr(\mathbf{Exp}_{\mathcal{A}}^{\gamma}(\eta) = true) - PrRand^{\gamma}\big),$$

where $PrRand^{\gamma}$ is the best probability to solve the challenge that an adversary can have without using oracle F. Formally, $PrRand^{\gamma}$ is the maximum of $pr(\mathbf{Exp'}_{\mathcal{A}}^{\gamma}(\eta) = true)$ where $\mathcal{A}$ ranges over any possible adversaries and $\mathbf{Exp'}$ is similar to $\mathbf{Exp}$ except that F cannot be used by A.

1.3 Decomposition of Security Criteria

In this section, we recall the reduction theorem given in [16]. Let $\gamma = (\theta_1, \theta_2; F_1, F_2; V_2)$ be a criterion. Let γ_1 and γ_2 be two criteria such that:

- There exist two PRTM G and H such that:

$$\begin{array}{rcl} G(H(s,\theta_2,\theta_2'),1,\theta_1) & = & F_1(s,\theta_1,\theta_2) \\ G(H(s,\theta_2,\theta_2'),0,\theta_1) & = & F_1(s,\theta_1,\theta_2') \end{array}$$

  Oracle G operates on a string of bits, thus it must receive two challenge informations, a bit b and θ_1.

- $\gamma_2 = (\theta_2; F_2; V_2)$ and $\gamma_1 = (b, \theta_1; G; verif_b)$ where b generates a random bit and $verif_b$ is the PRTM verifying that the output of the adversary is b: $verif_b(s, b, \theta_1) = (s \Leftrightarrow b)$.

- $F_2(s, \theta_1, \theta_2)$ and $V_2(s, \theta_1, \theta_2)$ do not depend on θ_1.

Then we say that (γ_1, γ_2) is a *valid simplified partition* of γ.

Theorem 1.1 *Let (γ_1, γ_2) be a valid simplified partition of γ. For any RTM $\mathcal{A}$, there exist two RTM $\mathcal{A}^o$ and $\mathcal{B}$ of similar complexity such that*

$$|\mathbf{Adv}_{\mathcal{A}}^{\gamma}(\eta)| \leq 2.|\mathbf{Adv}_{\mathcal{B}}^{\gamma_1}(\eta)| + |\mathbf{Adv}_{\mathcal{A}^o}^{\gamma_2}(\eta)|$$

2 Probabilistic Opacity

2.1 Systems and Observations

First, we give a general definition for random systems then we explain how their behaviors can be observed by an eavesdropper. As the results of this section do not depend on any particular model of systems, we simply consider randomized functions.

Let Σ be a finite alphabet representing actions made by a system. A trace is a finite sequence of actions, i.e., a word over Σ. Let Σ^* be the set of words over Σ and ϵ be the empty word.

A *system* is a random function Δ from a finite set S of initial states to sequences of actions. Random means that the system can perform some non-deterministic operations. For example it can pick up a random bit b. If $b = 0$, it performs action a else action b. The set of possible traces of a system Δ is denoted by $\Delta(S)$. In a similar way, $\Delta(\{s\})$ is the set of possible traces starting from s in S. This set can have more than one element as function Δ is random.

An *observation function* allows the eavesdropper to see only limited information about traces produced by the studied system. These functions are mappings from Σ to $\Sigma \cup \{\epsilon\}$. Hence, it is possible for an action to be totally invisible from the outside if the observation function replaces it with ϵ.

2.2 Opacity

Let us consider a system Δ with possible initial states S and an observation function obs. A property ψ is a predicate over S. A property ψ is *opaque* if given $s \in S$ and $t \in \Delta(\{s\})$, it is not possible, for an adversary that has access uniquely access to $obs(t)$ to know whether s verifies ψ. Here, not possible means that it should not be possible to achieve this with "reasonable" probability. This notion of opacity is introduced under the name of *initial opacity* in [7].

More than in opacity itself, we are here interested in the advantage that an adversary can get by having access to the observation of the trace. If a very vast majority of initial states in S verify ψ, the adversary can suppose that s verifies ψ even without looking at the trace. However, by looking at the trace, it is possible to get some new information and to deduce the result for sure. To define this advantage, we consider that the adversary $\mathcal{A}$ tries to win the following game/experiment:

1. An initial state s is chosen randomly in S;
2. The adversary $\mathcal{A}$ is given the observation of a trace in $\Delta(\{s\})$ and has to output a bit b;
3. $\mathcal{A}$ wins its challenge, when b is equivalent to the property "s satisfies ψ".

This game is represented by an experiment which is a random Turing machine. The experiment related to adversary $\mathcal{A}$ and to obs is the following RTM.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{obs}$:
$s \leftarrow S$
$t \leftarrow \Delta(s)$
$b \leftarrow \mathcal{A}(obs(t))$
return $b \Leftrightarrow (s \in \psi)$

The advantage is the difference between the probability that $\mathcal{A}$ solves its challenge and the best probability that one can get without access to the observation. Hence, it is defined by the following formula.

$$\mathbf{Adv}_{\mathcal{A}}^{obs} = 2.\big(pr(\mathbf{Exp}_{\mathcal{A}}^{obs} \rightarrow true) - PrRand^{\psi}\big)$$

Where $PrRand^{\psi}$ is the greatest possible value for $pr(\mathbf{Exp}_{\mathcal{B}}^{\epsilon} \rightarrow true)$ for any $\mathcal{B}$ and ϵ represents the observation function that associates ϵ to any action in Σ.

Note that the above definitions for the experiment and the advantage can easily be defined in an equivalent way by using the general notion of security criterion.

- Θ randomly generates an initial parameter s and a trace t;
- F gives access to the trace observation $obs(t)$;
- V verifies that the output bit b correctly answers the question: does s verify ψ ?

Criterion $(\Theta; F; V)$ has exactly the same related experiment and advantage as those given above.

Using the definition of advantage, it is possible to tell if an observation function has any use in trying to solve the challenge.

Definition 2.1 *Let Δ be a system, S be the set of its initial states and ψ a property over S. An observation function obs is called*

- safe for **strict** opacity *of ψ , if for any RTM $\mathcal{A}$,* $\mathbf{Adv}_{\mathcal{A}}^{obs} = 0$
- safe for **cryptographic** opacity *of ψ, if for any PRTM $\mathcal{A}$,* $\mathbf{Adv}_{\mathcal{A}}^{obs}$ *is negligible*
- safe for **plausible deniability** *of ψ, if for any RTM $\mathcal{A}$,* $\mathbf{Adv}_{\mathcal{A}}^{obs} \neq 2-2.PrRand^{\psi}$ *i.e.* $pr(\mathbf{Exp}_{\mathcal{A}}^{obs} \rightarrow true)$ *is different from* 1.

Plausible deniability coincides with the opacity notion introduced in [7], which is itself closely related to anonymity [19] and non-interference [12, 20]. This link and some useful basic properties are detailed in appendix A.1.

For strict opacity, if an observation function is safe then an adversary gets no advantage at all by looking at the observation. This is for example the informations exchanged by cryptographers during the cryptographs diner (if we consider that one of them paid the diner for any element of S).

For cryptographic opacity, an observation function may return some relevant information that cannot be exploited in a reasonable (i.e. polynomial) time. For example, if we consider that all the actions made by a system are encrypted using a safe encryption scheme, then the observation is safe. In this context, a safe encryption scheme is an IND-CPA encryption scheme, this is detailed further in this document.

The idea is that with plausible deniability, if the adversary observes some trace t, then it cannot conclude for sure whether property ψ is verified or not. There exists at least one initial state satisfying ψ and one not satisfying ψ that both produce the observation $obs(t)$.

There is no clear hierarchy among strict opacity and plausible deniability as the first notion does not imply the second one (this implication is only true when $PrRand$ is different from one).

3 Decidability of Strict Opacity for Finite Systems

Let us consider the case where only a finite number of traces can occur. Thus, we suppose that both S and $\Delta(S)$ are finite. With this assumption, the greatest advantage for any adversary can be computed. Moreover, there exists an adversary that reaches this advantage.

Let O be the set of all possible observations, i.e. $O = obs(\Delta(S))$. We first define the interest of an observation function obs. This definition is rather intuitive as an observation function can bring some advantage if the probability for ψ to be true knowing the observation is different from the general probability of ψ. This explains why this definition uses the term $|pr(\psi) - pr(\psi|o)|$.

Definition 3.1 *The interest I_{obs} of an observation function obs is given by:*

$$I_{obs} = 2.\sum_{o \in O} pr(o).\big|pr(\psi) - pr(\psi|o)\big|$$

Then, the main result of this section is that the interest is the greatest possible advantage. For that reason, as it is possible to effectively compute the interest of a given observation, safety for strictly opacity of an observation function is a decidable problem. The following proposition states the main result. Its proof is given in Appendix A.

Proposition 3.1 *For any adversary $\mathcal{A}$ and observation function obs, $|\mathbf{Adv}_{\mathcal{A}}^{obs}| \le I_{obs}$. Moreover, there exists an RTM $\mathcal{A}_{obs}$ whose advantage is exactly I_{obs}.*

It is important to notice that the second statement of the previous proposition asserts existence of an RTM $\mathcal{A}_{obs}$ with $\mathbf{Adv}_{\mathcal{A}}^{obs} = I_{obs}$. This RTM is not necessarily a legal adversary. Indeed, $\mathcal{A}_{obs}$ has an execution time which is linear in the number of possible observations. This is not a problem when considering strict opacity or plausible denying as adversaries are RTM. However, for cryptographic opacity, we only admit adversaries in PRTM. Worse, even if we assume the quite fair hypothesis (for an eavesdropper) that there is only a bounded number of messages which all have some bounded size, the number of possible observations may be exponential in the security parameter η, and hence, $\mathcal{A}_{obs}$ may not a PRTM.

Nevertheless, this result is interesting at least for strict opacity as shown now. Indeed, a consequence of this proposition is that no adversary has an advantage if and only if for any o in O, $pr(\psi) = pr(\psi|o)$. When one wants to verify strict opacity, it is possible to test that for any observation, the probability for ψ to be true assuming that observation is exactly the general probability for ψ to be true. Hence, we have

Proposition 3.2 *Let obs be an observation function and ψ a property. Then, obs is safe for strict opacity of ψ if and only if for any $o \in obs(\Delta(S))$, $pr(s \in \psi) = pr(s \in \psi|obs(s) = o)$*

4 An Approach to the Verification of Cryptographic Opacity

As we noted in the previous section, we make the finite behavior hypothesis for cryptographic systems (with passive adversaries). That is, we assume that S and $\Delta(S)$ are finite. The approach proposed for proving strict opacity using the interest of the observation function and the existence of an adversary matching this interest is not applicable for cryptographic opacity. Indeed, when considering cryptographic opacity, adversaries are restricted PRTM. Therefor, we present here a different approach. The main clue in this approach is to decompose the verification of cryptographic opacity into the verification of strict opacity for an abstracted system on one hand and the safety of the underlying encryption scheme on the other hand.

It is useful to notice that this approach is similar to the approach followed for proving secrecy properties of cryptographic protocols, where one proves an abstraction of the secrecy property while making the perfect cryptography hypothesis and relies on the fact that this verification is valid in the computational model, if the cryptographic primitives satisfy some well-defined properties. The formal justification of this approach is the result of recent research aiming at relating the formal and the computational models for security protocols [2, 4, 10]. These papers show that the Dolev-Yao [11] model is a safe abstraction of the computational model (where adversaries are poly-time Turing machines) as soon as the cryptographic primitives (e.g. the encryption scheme) verify some computational properties.

4.1 Specifications and Patterns

In the cryptographic setting, the alphabet Σ consists of the symbols, 0 and 1. Thus, an action of the alphabet is a bit-string. We consider systems that produce some finite size bit-string (usually, their size is polynomial in the security parameter η).

To define the abstract systems, we introduce patterns which are simply elements of the free algebra of terms almost as in the Dolev-Yao model. It is *almost* because in our setting and as we are interested in opacity, we have to be careful in handling the random coins[3] used in encryption. Let us explain. In the simplest Dolev-Yao model (also called the formal or symbolic model) an encryption of a message m with key pk is represented by the term $\{m\}_{pk}$. Thus, two message $\{m_1\}_{pk_1}$ and $\{m_2\}_{pk_2}$ are equal iff $m_1 = m_2$ and $pk_1 = pk_2$. Moreover, an adversary who does not know the inverse key of pk cannot get any information from $\{m\}_{pk}$. This means that the random nonce is completely abstracted away. In some refinements of this model, however, labels are introduced to distinguished encryptions made at different instants during a protocol execution [10]. Such labels are only an approximation of random coins as the latter may be equal even when two encryptions are performed at different instants. As we want to verify the Chaum voting scheme, we have to include explicitly random coins in our patterns. Therefor, we write $\{m; N\}_{pk}$ to represent the result of encrypting m with key pk using nonce N as random coins for the encryption algorithm.

Let $\mathcal{K}$ be an infinite set of keys (as explained above rather key names); k^{-1} represents the private key corresponding to a public key k. Moreover, let $\mathcal{N}$ be a set of nonces. Patterns are defined by the following grammar where k is a key, bs a bit-string and N is a nonce:

$$pat ::= bs|N|\langle pat, pat\rangle|\{pat; N\}_k|k \quad k \text{ may be a public or private key}$$

Without loss of generality, we consider abstract systems that only produce one pattern and not a list of patterns as it is possible to concatenate patterns using pairing. Thus, a *specification* Δ_s is a function from S to pat.

Obviously, given a pattern pat the information that can be extracted from pat depends on the set of private keys that can be computed from pat. The set of patterns that can be learned/computed from a pattern is defined as follows:

Definition 4.1 *Let p be a pattern, the set $dec(p)$ is inductively defined by the following inferences.*

- *p is in $dec(p)$.*
- *If $\langle p_1, p_2\rangle$ is in $dec(p)$, then p_1 and p_2 are in $dec(p)$.*

[3] Random coins are also nonces but some times we use rather random coins to insist on the fact they are used to randomize encryptions

- *If* $\{p_1; N\}_k$ *and* k^{-1} *are in* $dec(p)$*, then* p_1 *and* N *are in* $dec(p)$*.*
- *If* $\{p_1; N\}_k$ *and* N *are in* $dec(p)$*, then* p_1 *is in* $dec(p)$*.*

Notice that since, we only consider atomic keys, we only have to consider decompositions. It is also useful to notice that the last clause is usually not considered in the Dolev-Yao model. This clause is motivated by the existence of IND-CPA algorithms such that the knowledge of the random used for encryption allows to decrypt the message. An example of such algorithm is presented in [6].

A pattern has also a denotation in the cryptographic setting. This depends on a context θ that associates keys and nonces to their corresponding bit-string values. Thus, the cryptographic (or computational) value of a term $\{pat; N\}_k$ is $\mathcal{E}(m, bs, bs')$, where m is the value of pat, bs the value of pk and bs' the value N. Let θ be a mapping associating bit-strings to nonces and keys. The value of a pattern in the context θ is defined recursively:

$$\begin{array}{lll} v(bs,\theta) = bs & v(\langle p_1,p_2\rangle,\theta) = v(p_1,\theta).v(p_2,\theta) & \\ v(N,\theta) = \theta(N) & v(\{p;N\}_k,\theta) = \mathcal{E}(v(p,\theta),\theta(k),\theta(N)) & v(k,\theta) = \theta(k) \end{array}$$

Let us briefly summarize what we have introduced. We defined the systems we want to consider whose behavior in each initial state s is a set of patterns and we have associated to each pattern its value, a bit-string, in a given context.

We now turn our attention to the observations we can make about a pattern. We define two observations. The concrete observation of a pattern pat in a context θ is defined as follows: $obs_c(pat, \theta) = v(pat, \theta)$, that is, obs_c corresponds to the observations that can be made in the cryptographic setting. The abstract observation obs_a applied to a pattern replaces every sub-terms of the form $\{pat; N\}_k$ with $\Diamond^N$, that is, it simply replaces it by a black box. Formally, patterns are transformed in obfuscated patterns which are given by the following grammar:

$$opat ::= bs | N | \langle opat, opat\rangle | \{opat; N\}_k | \Diamond^N | k$$

And observation obs_a of a pattern pat is recursively defined by the following rules.

$$\begin{array}{ll} obs_a(bs) = bs & obs_a(\langle p_1,p_2\rangle) = \langle obs_a(p_1), obs_a(p_2)\rangle \\ obs_a(N) = N & obs_a(\{p;N\}_k) = \{obs_a(p);N\}_k \text{ if } k^{-1} \in dec(pat) \vee N \in dec(pat) \\ obs_a(k) = k & obs_a(\{p;N\}_k) = \Diamond^N \text{ else} \end{array}$$

As encryption cycles may lead to some vulnerabilities, we restrict ourselves to *well-formed* patterns. For that purpose, we define an ordering on pairs consisting of a key and a nonce. Let pat be a pattern and let $E_<$ be the set of pairs (k, N) such that there is a pattern of the form $\{pat'; N\}_k$ in $dec(pat)$ with k and N not in $dec(pat)$. Then, for $(k, N), (k', N') \in E_<$, $(k, N) < (k', N')$ iff there exist two patterns $\{pat_1; N\}_k$ and $\{pat_2; N'\}_{k'}$ in $dec(pat)$ verifying one of the following conditions:

1. N, k or k^{-1} is a sub-term of $pat_2; N'$;
2. $N = N'$ and $\{pat_1; N\}_k \neq \{pat_2; N\}_{k'}$.

A pattern pat is *well-formed*, if the projection of $<$ on keys is acyclic. Finally, we only consider *well-formed specifications*, i.e. specifications that output well-formed patterns.

The conditions above imply that if pat is well-formed, then for $(k, N) \in E_<$, there is only one encoding using each N (and a non-deducible key) in $dec(pat)$. Hence when obs_a transforms an encoding into $\Diamond^N$, this always denotes the exact same encoding (in particular, there is no randomness-reuse as described in [5]). Thus the N label can be seen as a constraint over encodings (specifying possible bit-to-bit equalities). This is why, equality between two $opat$ is defined modulo renaming of the nonces. To illustrate this, let us consider two patterns $pat_0 = \langle\{m; N\}_k, \{m; N\}_k\rangle$ and $pat_1 = \langle\{m; N''\}_k, \{m; N'\}_k\rangle$. Then $obs_a(pat_0) = \langle\Diamond^N, \Diamond^N\rangle$ and $obs_a(pat_1) = \langle\Diamond^{N'}, \Diamond^{N''}\rangle$. As $obs_a(pat_1)$ and $obs_a(pat_2)$ are different, and hence, pat_0 and pat_1 are distinguishable. If we consider $pat_0 = \{m; N\}_k$ and $pat_1 = \{m; N''\}_k$. Then $obs_a(pat_0) = \Diamond^N$ and $obs_a(pat_1) = \Diamond^{N'}$. In this case, $obs_a(pat_1)$ and $obs_a(pat_2)$ are equal (modulo renaming), and hence, pat_0 and pat_1 are indistinguishable. And in fact, if the encryption scheme is IND-CPA, pat_0 and pat_1 are indistinguishable even in the computational setting.

Main result The main result of this paper, that we prove in Section 6, is that for each adversary $\mathcal{A}$, there exist two adversaries $\mathcal{A}^o$ and $\mathcal{B}$ such that

$$|\mathbf{Adv}_{\mathcal{A}}^{obs_c \times obs_a}| \leq |\mathbf{Adv}_{\mathcal{A}^o}^{obs_a}| + 2.|\mathbf{Adv}_{\mathcal{B}}^{n-RPAT-CPA}|$$

Where n is the number of keys, n-RPAT-CPA is a security criterion verified by any IND-CPA algorithm, observable $obs_c \times obs_a$ gives access to both obs_c and obs_a (see appendix A.1 for details). This means that if the encryption scheme used is IND-CPA, opacity in the formal world implies opacity in the computational world.

4.2 Application: the Classical Abadi-Rogaway Result

Using our main theorem, it is possible to prove a slightly extended version of the seminal result of Abadi and Rogaway [2]. This result states that provided the used encryption scheme is IND-CPA indistinguishability in the formal (Dolev-Yao) model implies indistinguishability in the computational model. In fact, obfuscated patterns are close to patterns as introduced in [2]. The main difference is that our patterns explicitly represent random coins. However, it is still possible to get exactly Abadi and Rogaway's result by assuming fresh distinct nonces for every encryption. If we consider messages with no encryption cycles, then the corresponding patterns (using fresh nonces) are well-formed. Moreover as nonces used for encryption are fresh, each $\Diamond^N$ has a different label, thus to test equality these labels are not considered. The Abadi-Rogaway theorem can be stated as an opacity problem. Let m_0 and m_1 be two well-formed patterns. There are two initial states in S: 0 and 1. Specification $\Delta_S(s)$ outputs m_s. Then m_0 and m_1 are *indistinguishable* if for any adversary $\mathcal{A}$, $|\mathbf{Adv}_{\mathcal{A}}^{obs_c}|$ is negligible.

Proposition 4.1 *Let m_0 and m_1 be two well-formed patterns such that $obs_a(m_0) = obs_a(m_1)$. If the encryption scheme $\mathcal{AE}$ used in v is IND-CPA then m_0 and m_1 are indistinguishable.*

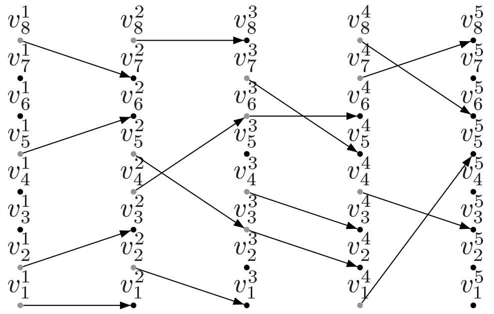
This result is immediate if we apply the above theorem: as obs_a returns the same result for the two patterns, $|\mathbf{Adv}_{\mathcal{A}^o}^{obs_a}|$ is equal to zero. Hence as $|\mathbf{Adv}_{\mathcal{B}}^{n-RPAT-CPA}|$ is negligible, the advantage of $\mathcal{A}$ is also negligible and we get the desired result.

5 Application: Chaum's Visual Electronic Voting

To illustrate our results, we consider a slightly modified version of the electronic voting scheme proposed by Chaum [9]. The main advantage of this scheme is that it is verifiable using an audit procedure that preserves opacity of the votes [13], i.e., what did voter V vote?. However, this paper still makes the perfect cryptography hypothesis, encryptions are considered as black-box and are not taken into account. We give here a proof of security for Chaum's voting scheme in a computational setting. For that purpose, we assume that the encryption scheme is IND-CPA and prove that then, security results still hold (but we may have to add some negligible terms representing brute force attack against the encryption scheme).

5.1 System Description

Let us briefly recall how the Chaum's voting scheme works. We omit some important pieces (mostly the visual aspect) that are not relevant for this paper. The interested reader may consider reading [9] or [8] for details.



A vote session uses n trustees to guarantee the security of the procedure. Each trustee C_i has a public key pk_i and an associated secret key sk_i. The vote procedure works as follows: voters choose a vote value v, then the following bit-string is given to C_1 where each nonce N^i (unique for any voter) represents the random information used to compute the encryption layer using key pk_i:$\{...\{v; N^n\}_{pk_n}; N^1\}_{pk_1}$. Each trustee decodes its layer then makes a random permutation of all the votes and submits the resulting list to the next trustee. All the intermediate lists are made public and the last list allows anyone to compute the results of the vote.

After the decoding phase, an audit process allows to verify that trustees behave correctly with great probability. Hence each trustee C_i has to reveal the permutation it used for half the ballots. Thus it shows for these ballots the link between the input ballot encoded by pk_i and the output ballot encoded by pk_{i+1}, the trustee also shows nonce N^i to allow anyone to check that the link is valid (it is supposed that the encryption algorithm allows the trustee to get this nonce). Verified ballots are not chosen randomly but as described in figure above. The first set of verified ballot (for step 1) is chosen randomly. For step 2, verified ballots correspond to unconnected ballots w.r.t. step 1. For step 3, verified ballots are half unconnected ballots and half connected ones, the halves are chosen randomly. Finally, for step 4, verified ballots are unconnected ballots w.r.t. step 3. In the figure, v_j^i is the i^{th} ballots in the input of the j^{th} trustee and σ_j is the permutation chosen by this trustee. The set I_j consists of integers k such that the transition that reaches v_k^j is revealed.

5.2 Verification of the system

The property we are interested in is opacity of the vote. However, it should be possible to generalize our results to more complex properties like the bound over variation distance given in [13].

To simplify, let us suppose that there are two possible values for the vote: y and n. Then the set S of initial states contains all the vote distributions that give a fixed final result, i.e. for any element of S the number of voters that choose y is fixed, all the other variables are chosen at random (permutations, audit sets).

We study the opacity of property $\psi = (v_1^1 = y)$: are we able to deduce that the vote chosen by voter 1 is y ? We want to prove that the audit information cannot bring any advantage to an attacker. This requires that for any observation o, $pr(\psi|o) = pr(o)$. Then, as $PrRand$ is given by the vote result, it is clear that it is impossible to guess the value of v_1 with better efficiency than when answering the most probable vote with respect to the result. The specification of the system is pretty straightforward: Δ_s outputs the revealed permutations, the ballots lists and the nonces used to check the link for any $k \in I_j$. The output pattern is well-formed (there are no cycles for $<$, the form of the ballots gives $pk_n < pk_{n-1} < ... < pk_1$).

After applying obs_a, the abstract system gives information on the permutations and the final ballot line, indeed there are two cases for remaining encrypted ballots: they can be abstracted to $\diamondsuit^N$ or they can be linked by a permutation to a vote in the final (unencrypted) line and so these ballots are useless to the description because it would be possible from the rest of the description to rebuild them using the final vote and the revealed nonce. Let o be an observation in the formal world. Let p_y be the percentage of voter that choose y. Then a quick calculus detailed in appendix A.3 gives us that $pr(v_1^1 = y|o) = p_y$. This proves opacity of ψ in the abstract world.

Security in the computational world is easy to obtain by applying our composition theorem: let $\mathcal{A}$ be a PRTM, then there exist $\mathcal{A}^o$ and $\mathcal{B}$ two PRTMs such that:

$$|\mathbf{Adv}_{\mathcal{A}}^{obs_c \times obs_a}| \leq |\mathbf{Adv}_{\mathcal{A}^o}^{obs_a}| + 2.|\mathbf{Adv}_{\mathcal{B}}^{n-RPAT-CPA}|$$

The advantage related to obs_a is zero. Moreover, if we consider that the encryption scheme used is IND-CPA, then $\mathbf{Adv}_{\mathcal{B}}^{n-RPAT-CPA}$ is negligible. Thus the advantage of $\mathcal{A}$ is negligible and we can conclude that the observable obs_c is safe for the cryptographic opacity of ψ.

6 Formal Description of the Main Result

The aim of this section is to prove our main result. We proceed in two steps. We first define n-RPAT-CPA and relate it to IND-CPA. Then, we prove that the advantage of any adversary who accesses the observation functions obs_a and obs_c is bounded by a linear combination of the advantage of an adversary that has access to obs_a and the advantage of an adversary that has access to obs_c. In both steps, we apply Theorem 1.1. Although, the criterion we introduce is implied to IND-CPA, it is technically more appealing to use it to prove the main result. Besides this, our new criterion is of interest on its own as it clarifies and discloses some subtleties related to the treatment of random coins.

6.1 The RPAT Extension to IND-CPA

In IND-CPA, the experiment consists of generating a random bit b and a random public key pk. The adversary tries to guess the value of b. For that purpose, it accesses a *left-right oracle* submitting two bit-strings bs_0 and bs_1 and receives the encryption of bs_b using pk. The adversary also has access to the public key. An encryption scheme $\mathcal{AE}$ is said secure against IND-CPA if any PRTM has a negligible advantage in trying to find b (the advantage is two times the probability to answer correctly minus one). The criterion we introduce below allows the adversary to ask for encryption of patterns where challenged keys may be included and insisting on using the same random coins in different encryptions. Moreover, patterns may include encryption with the adversaries keys. As we show later these extensions do not give more power to an adversary, if he is deemed to produce well-formed patterns.

Let us now introduce n-RPAT-CPA. To do so, let n be a non-negative integer. We first define R-patterns:

$$rpat ::= bs|N|\langle rpat, rpat\rangle|\{rpat; N\}_k|\{rpat; N\}_{bs}|\{rpat; bs\}_k|\{rpat; bs\}_{bs'}|k$$

The only difference with respect to patterns introduced in Section 4 is the encryption with a non-challenge key or a non-challenge nonce. The evaluation function v is extended to R-patterns.

The experiment defining the criterion n-RPAT-CPA is as follows:

$$\begin{array}{ll}
pat_0, pat_1, \sigma \leftarrow \mathcal{A}_1; & \\
b \leftarrow \{0,1\}; & \\
(bs_i, bs'_i) \leftarrow \mathcal{KG}(\eta); & \text{for } i = 1, \cdots, n \\
bs''_i \leftarrow \{0,1\}^\eta; & \text{for } i = 1, \cdots, l \\
\theta \leftarrow [b, (pk_1, sk_1) \mapsto (bs_1, bs'_1), \cdots, (pk_n, sk_n) \mapsto (bs_n, bs'_n), & \\
\quad N_1 \mapsto bs''_1, \cdots, N_l \mapsto bs''_l]; & \\
y \leftarrow v(pat_b, \theta); & \\
d \leftarrow \mathcal{A}_2(y, \sigma) & \\
V(d, \theta) \leftarrow b = d &
\end{array}$$

The adversary is split up in two parts $\mathcal{A}_1$ and $\mathcal{A}_2$, $\mathcal{A}_1$ outputs two patterns pat_0 and pat_1. Pattern pat_b is computed (l is the number of nonces used by the pattern and n is the maximal number of keys that a pattern can use), it is given to $\mathcal{A}_2$ which has to answer the value of b. It is also possible to consider a single adversary $\mathcal{A}$ that access a left-right oracle F, giving it the two patterns. In this case, oracle F only answer its first call.

In the experiment of n-RPAT-CPA, it is mandated that $\langle pat_0, pat_1\rangle$ is well-formed.

We show that algorithms secure w.r.t. IND-CPA are secure w.r.t. n-RPAT-CPA and as, there are algorithms strongly believed to verify IND-CPA, these algorithms also verify n-RPAT-CPA.

Proposition 6.1 *If an asymmetric encryption scheme is secure against IND-CPA, then it is secure against n-RPAT-CPA for any number of keys n.*

The proof is detailed in appendix A.4.This proposition can be generalized to a polynomial number of keys and nonces using the technique introduced in [16].

6.2 Composition Result

Our main result states that given a specification, the advantage of an adversary against the concrete system is lower than the advantage of the abstract system and the advantage of another adversary against n-RPAT-CPA.

Theorem 6.1 *For each adversary $\mathcal{A}$, there exist two adversaries $\mathcal{A}^o$ and $\mathcal{B}$ such that*

$$|\mathbf{Adv}_{\mathcal{A}}^{obs_c \times obs_a}| \leq |\mathbf{Adv}_{\mathcal{A}^o}^{obs_a}| + 2.|\mathbf{Adv}_{\mathcal{B}}^{n-RPAT-CPA}|$$

The proof of the main theorem is given in appendix A.5

Using proposition 6.1, it is clear that if an encryption scheme is secure against IND-CPA, then it is secure against n-RPAT-CPA for any integer n. Therefor, we have

Corollary 6.1 *If the encryption scheme $\mathcal{AE}$ used in v is IND-CPA and obs_a brings negligible advantage to any adversary then obs_c brings negligible advantage to any adversary.*

Conclusion

Probabilistic opacity is far more realistic than classical opacity. However, our main result makes it simple to prove cryptographic opacity for systems involving cryptographic primitives by first proving opacity for the related abstract system and then using an IND-CPA cryptographic scheme. This composition result seems very general as it can be applied to get the classical Abadi-Rogaway result. Another interesting result is the implication from IND-CPA to the new criterion n-RPAT-CPA. This criterion allows us to consider systems where the random information used for public-key encryption is exchanged (usually, to allow checking of this encryption). This is necessary to deal with complex systems such as Chaum's vote protocol. Hence, our last result is the first (to our knowledge) proof of this voting scheme in a computational setting.

A natural extension of this work is to consider the case of active adversaries as in [3]. To do this, we need to consider simulation but modular proofs seems quite harder to obtain when using this relation. We also intend to extend our result to other cryptographic primitives such as digital signature, symmetric encryption or hashing as in [16]. Finally, it would be of interest to extend the computational security results for Chaum's voting scheme to properties given in [13].

References

[1] M. Abadi and J. Jürgens. Formal eavesdropping and its computational interpretation. In *4th Intern. Symp. on Theoretical Aspects of Computer Software*, volume 2215 of *lncs*. springer, 2001. (document)

[2] Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness o f formal encryption). In *IFIP International Conference on Theoretical Computer Science (IFIP TCS2000)*, Sendai, Japan, 2000. Springer-Verlag, Berlin Germany. (document), 4, 4.2

[3] M. Backes and B. Pfitzmann. Computational probabilistic non-interference, 2002. (document), 6.2

[4] Michael Backes, Birgit Pfitzmann, and Michael Waidner. A composable cryptographic library with nested operations. In *Proceedings of the 10th ACM conference on Computer and communication security*, pages 220–230. ACM Press, 2003. (document), 4

[5] M. Bellare, A. Boldyreva, and J. Staddon. Randomness re-use in multi-recipient encryption schemes. In Y. Desmedt, editor, *Public Key Cryptography – PKC 2003*, volume 2567 of *lncs*. springer, 2003. 4.1

[6] E. Bresson, D. Catalano, and D. Pointcheval. A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In C. S. Laih, editor, *Proc. of Asiacrypt'03*, volume 2894 of *LNCS*, pages 37–54, Taipei, TW, November-December 2003. IACR, Springer-Verlag. (document), 4.1

[7] Jeremy W. Bryans, Maciej Koutny, Laurent Mazaré, and Peter Y.A. Ryan. Opacity generalised to transition systems. Technical Report TR-2004-25, Verimag, Centre Équation, 38610 Gières, December 2004. (document), 2.2, 2.2, A.1

[8] D. Chaum, P. Ryan, and S. Schneider. A practical, voter-verifiable election scheme. Technical Report 880, University of Newcastle upon Tyne, School of Computing Science, Dec 2004. (document), 5.1

[9] David Chaum. E-voting: Secret-ballot receipts: True voter-verifiable elections. *IEEE Security & Privacy*, 2(1):38–47, January/February 2004. 5, 5.1

[10] V. Cortier and B. Warinschi. Computationally sound, automated proofs for security protocols. In *Proceedings of the 14th European Symposium on Programming (ESOP'05)*, Lecture Notes in Computer Science, Edinburgh, U.K., April 2005. Springer. To Appear. (document), 4, 4.1

[11] D. Dolev and A. C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983. (document), 4

[12] Riccardo Focardi and Roberto Gorrieri. A taxonomy of trace-based security properties for CCS. In *Proceedings of the Computer Security Foundations Workshop VII (CSFW '94)*, pages 126–137. IEEE, 1994. 2.2, A.1

[13] Marcin Gomukiewicz, Marek Klonowski, and Mirosaw Kutyowski. Rapid mixing and security of chaum's visual electronic voting. In *Proceedings of ESORICS 2003*, October 2003. 5, 5.2, 6.2

[14] J. W. Gray. Probabilistic interference. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 180–187, 1990. (document)

[15] Romain Janvier, Yassine Lakhnech, and Laurent Mazaré. Completing the picture: Soundness of formal encryption in the presence of active adversaries. In *Proceedings of the 14th European Symposium on Programming (ESOP'05)*, Lecture Notes in Computer Science, Edinburgh, U.K., April 2005. Springer. To Appear. (document)

[16] Romain Janvier, Yassine Lakhnech, and Laurent Mazaré. (de)compositions of cryptographic schemes and their applications to protocols. Technical report, Verimag, Centre Équation, 38610 Gières, To Appear 2005. 1, 1.3, 6.1, 6.2

[17] Peeter Laud. Semantics and program analysis of computationally secure information flow. *Lecture Notes in Computer Science*, 2028:77+, 2001. (document)

[18] Peeter Laud. Symmetric encryption in automatic analyses for confidentiality against active adversaries. In *IEEE Symposium on Security and Privacy*, pages 71–85, 2004. (document)

[19] Steve Schneider and Abraham Sidiropoulos. CSP and anonymity. In *ESORICS*, pages 198–218, 1996. 2.2, A.1

[20] Peter Y.A.Ryan. Mathematical models of computer security. In *Foundations of Security Analysis and Design*, number 2171 in LNCS, 2000. 2.2, A.1

A Appendix

A.1 Basic Properties

Using the definition of advantage given above, some rather obvious properties can be stated. They prove relations between different kind of advantages except the first one that details the value of $PrRand$. This proposition relates the value of $PrRand$ to the probability for an element of s to be in S.

Proposition A.1 *Let $pr(\psi)$ be the probability for a random element of S to verify ψ. Then the greatest possible advantage is obtained by answering 1 if $pr(\psi) > \frac{1}{2}$ and 0 otherwise. Hence, the best probability is: $PrRand^{\psi} = \frac{1}{2} + |pr(\psi) - \frac{1}{2}|$*

Proof: Let $\mathcal{A}$ be an adversary that can only use the ϵ observation function. As $\mathcal{A}$ does not have access to any oracle. Its behavior can be represented by its probability p to answer 1. Of course, $\mathcal{A}$ also has probability $1-p$ to answer 0. The probability that $\mathcal{A}$ answers correctly is:

$$\begin{aligned} pr(\mathbf{Exp}_{\mathcal{A}}^{\epsilon} \to true) &= pr(\psi).p + (1 - pr(\psi)).(1-p) \\ &= 1 - pr(\psi) + p.(2.pr(\psi) - 1) \end{aligned}$$

Then, if $pr(\psi) \geq \frac{1}{2}$,

$$pr(\mathbf{Exp}_{\mathcal{A}}^{\epsilon} \to true) \leq pr(\psi)$$

In the other case,

$$pr(\mathbf{Exp}_{\mathcal{A}}^{\epsilon} \to true) \leq 1 - pr(\psi)$$

These two inequalities allow us to deduce the following.

$$\begin{aligned} pr(\mathbf{Exp}_{\mathcal{A}}^{\epsilon} \to true) &\leq max(pr(\psi), 1 - pr(\psi)) \\ &\leq |pr(\psi) - \frac{1}{2}| + \frac{1}{2} \end{aligned}$$

■ By looking at the $PrRand$ definition, an adversary cannot get a better probability to succeed with no observations. Thus it is clear that an adversary cannot get a positive advantage.

Proposition A.2 *For any adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{\epsilon} \leq 0$.*

If an adversary gets a negative advantage, it is possible to inverse its behavior (by inversing its output) and this may create a greater advantage.

Proposition A.3 *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ of similar complexity such that $\mathbf{Adv}_{\mathcal{B}}^{obs} + 4.PrRand^{\psi} = 2 - \mathbf{Adv}_{\mathcal{A}}^{obs}$.*

The following properties state that some observation functions can bring no advantage compared to other observation functions. Basically, if an observation obs_2 has a "finer resolution" than obs_1, then the advantage related to obs_2 is greater than the one related to obs_1.

Proposition A.4 *Let obs_1 and obs_2 be two observation functions such that for any pair of trace t, t' in $\Delta(S)$, $obs_2(t) = obs_2(t')$ implies $obs_1(t) = obs_1(t')$. Then, for any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ of similar complexity such that $\mathbf{Adv}_{\mathcal{B}}^{obs_2} = \mathbf{Adv}_{\mathcal{A}}^{obs_1}$.*

If obs_1 and obs_2 are two observation functions, $obs_1 \times obs_2$ gives access to both simultaneously. Formally, let obs_1 and obs_2 be two observation functions from Σ to $\Sigma \cup \{\epsilon\}$. Then $obs_1 \times obs_2$ is an observation function from Σ to $\Sigma \cup \{\epsilon\} \times \Sigma \cup \{\epsilon\}$ such that the result on action a is $(obs_1(a), obs_2(a))$. It is clear that the sum of two observation functions gives a better advantage than the advantages related to one of the two functions.

Proposition A.5 *Let obs_1 and obs_2 be two observation functions. For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ of similar complexity such that $\mathbf{Adv}_{\mathcal{B}}^{obs_1 \times obs_2} = \mathbf{Adv}_{\mathcal{A}}^{obs_1}$.*

Relation with Classical Opacity

Probabilistic opacity is closely linked to the opacity notion introduced in [7]. We consider plausible deniability for probabilistic opacity and initial opacity with a static observation function.

Let Δ be a deterministic system and S be the set of initial states. Let Π be a labeled transition system whose set of initial states is S and that has the same behavior as Δ. Let obs be an observation function.

Proposition A.6 *A predicate ψ over S is opaque with respect to obs iff for any adversary $\mathcal{A}$,*

$$\mathbf{Adv}_{\mathcal{A}}^{obs} < 2.(1 - PrRand^{\psi})$$

Classical opacity can be linked to anonymity [19] and non-interference [12, 20] and the same thing can be done with probabilistic opacity.

A.2 Proof of proposition 3.1

Proposition *For any adversary $\mathcal{A}$ and observation function obs,*

$$|\mathbf{Adv}_{\mathcal{A}}^{obs}| \leq I_{obs}$$

Moreover, there exists an adversary $\mathcal{A}_{obs}$ which advantage is exactly I_{obs}.

Proof: This proof is achieved in three steps:

Step 1 First, consider the case where there is only one possible observation, $|O| = 1$. Then, the calculus defining $PrRand$ can be applied. Adversary $\mathcal{A}$ has a probability p (resp. $1 - p$) to answer 1 (resp. 0).

$$\begin{aligned}
pr(\mathbf{Exp}_{\mathcal{A}}^{obs} \to true) &= pr(\mathbf{Exp}_{\mathcal{A}}^{obs} \to true|s \in \psi).pr(\psi) + pr(\mathbf{Exp}_{\mathcal{A}}^{obs} \to true|s \notin \psi).pr(\neg\psi) \\
&= p.pr(\psi) + (1-p).(1 - pr(\psi)) \\
&= (1 - pr(\psi)) + p.(2.pr(\psi) - 1)
\end{aligned}$$

Then, if $pr(\psi) \geq \frac{1}{2}$,

$$pr(\mathbf{Exp}_{\mathcal{A}}^{obs} \to true) \leq pr(\psi)$$

In the other case,

$$pr(\mathbf{Exp}_{\mathcal{A}}^{obs} \to true) \leq 1 - pr(\psi)$$

These two inequalities allow us to deduce the following.

$$\begin{aligned}
pr(\mathbf{Exp}_{\mathcal{A}}^{obs} \to true) &\leq max(pr(\psi), 1 - pr(\psi)) \\
pr(\mathbf{Exp}_{\mathcal{A}}^{obs} \to true) &\leq |pr(\psi) - \frac{1}{2}| + \frac{1}{2} \\
pr(\mathbf{Exp}_{\mathcal{A}}^{obs} \to true) &\leq PrRand^{\psi}
\end{aligned}$$

Hence, the advantage is negative.

Step 2 Now, it is possible to generalize the above result for any set O.

$$\begin{aligned}
pr(\mathbf{Exp}_{\mathcal{A}}^{obs} \to true) &= \sum_{o \in O} pr(o).pr(\mathbf{Exp}_{\mathcal{A}}^{obs} \to true|o) \\
&\leq \sum_{o \in O} pr(o).(|pr(\psi|o) - \frac{1}{2}| - \frac{1}{2}) \\
&\leq \frac{1}{2} + \sum_{o \in O} pr(o).|pr(\psi|o) - \frac{1}{2}|
\end{aligned}$$

We introduce $PrRand^{\psi}$ in the former equation using its form $|pr(\psi) - \frac{1}{2}| + \frac{1}{2}$. Hence,

$$\begin{aligned}
\mathbf{Adv}_{\mathcal{A}}^{obs} &\leq \sum_{o \in O} pr(o).(|pr(\psi|o) - \frac{1}{2}| - |pr(\psi) - \frac{1}{2}|) \\
|\mathbf{Adv}_{\mathcal{A}}^{obs}| &\leq \sum_{o \in O} pr(o).||pr(\psi|o) - \frac{1}{2}| - |pr(\psi) - \frac{1}{2}|| \\
&\leq \sum_{o \in O} pr(o).|pr(\psi|o) - pr(\psi)| \\
&\leq I_{obs}
\end{aligned}$$

Step 3 The machine $\mathcal{A}_{obs}$ which advantage is exactly I_{obs} is very simple:

Adversary $\mathcal{A}_{obs}(o)$:
$p \leftarrow pr(\psi|o)$
if $p \geq 0.5$, **return** $true$
else return $false$

Probability for the different obs equivalence classes are hardwired in the machine. As there are only a finite number of classes, $\mathcal{A}_{obs}$ works in polynomial time (w.r.t. the size of o). The advantage of this adversary can be computed in a similar way as step 1 and 2. ■

A.3 Opacity of Chaum Voting Scheme

There are two cases to consider. First case, the link starting from v_1^1 is revealed.

$$\begin{aligned} pr(v_1^1 = y|o) &= pr(v_{1\sigma_1}^2 = y|o) \\ &= \frac{2}{n}\sum_{i \notin I_3} pr(v_i^3 = y|o) \\ &= \frac{1}{n}\sum_{i=1}^{n} pr(v_i^4 = y|o) \\ &= p_y \end{aligned}$$

In the second case, a similar calculus can be done.

$$\begin{aligned} pr(v_1^1 = y|o) &= \frac{2}{n}\sum_{i \notin I_2} pr(v_i^2 = y|o) \\ &= \frac{2}{n}\sum_{i \in I_3} pr(v_i^3 = y|o) \\ &= \frac{1}{n}\sum_{i=1}^{n} pr(v_i^4 = y|o) \\ &= p_y \end{aligned}$$

A.4 Proof of proposition 6.1

In this section, we prove that IND-CPA implies n-RPAT-CPA. Henceforth, let $\mathcal{AE}$ be an encryption scheme. We proceed in three steps.

Let n-RPAT$_c$-CPA be the same criterion as n-RPAT-CPA except that adversaries can only output clean patterns, i.e. $\langle pat_0, pat_1\rangle$ such that $dec(pat_0, pat_1)$ does not contain any nonce nor private key. Our first step consists in proving that IND-CPA implies 1-RPAT$_c$-CPA.

Lemma A.1 *If $\mathcal{AE}$ is secure w.r.t. IND-CPA then it is secure w.r.t.* 1*-RPAT$_c$-CPA.*

Proof: Let us consider an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against 1-RPAT$_c$-CPA. We construct an adversary $\mathcal{B}$ against IND-CPA whose advantage the same as the advantage of $\mathcal{A}$.

Let k be the unique challenge key. As there are no cycles among keys, there does not exist any pair of nonces N, N' such that $(k, N) < (k, N')$. Hence relation $<$ is empty. For any nonce N such that $(k, N) \in E_<$, N appears in exactly one encoding (but this encoding can be used several times as in $\langle\{m; N\}_k, \{m; N\}_k\rangle$) and in this case it appears as a random coin.

The adversary $\mathcal{B}$ uses $\mathcal{A}_1$ and $\mathcal{A}_2$ as sub-machines. However, as $\mathcal{A}_1$ outputs patterns while $\mathcal{B}$ has to output messages, $\mathcal{B}$ has to simulate the evaluation function v using the IND-CPA left-right oracle. This is done using the function $vsim$ in the description of $\mathcal{B}$:

$pat_0, pat_1, \sigma \leftarrow \mathcal{A}_1;$
$y \leftarrow vsim(pat_0, pat_1);$
$d \leftarrow \mathcal{A}_2(y, \sigma);$
return d

We now have to describe the function $vsim$. First notice that nonces N that do not appear in $E_<$ appear encrypted in the patterns. Therefor, $vsim$ generates some random values for these nonces and creates the corresponding environment θ_{sim}. The context θ_{sim} is extended with public key k. Next, as pat_0 and pat_1 have the same obs_a (modulo renaming), the following recursive function $vrec_{\theta_{sim}}$ is applied to pat_0, pat_1:

$$\begin{array}{rcl} vrec_{\theta_{sim}}(bs, bs) & = & bs \\ vrec_{\theta_{sim}}(m_1.m_2, m_1'.m_2') & = & vrec_{\theta_{sim}}(m_1, m_1').vrec_{\theta_{sim}}(m_2, m_2') \\ vrec_{\theta_{sim}}(\{m; N\}_k, \{m'; N\}_k) & = & F\big(v(m, \theta_{sim}), v(m', \theta_{sim})\big) \end{array}$$

Note that for the last line, if $vrec_{\theta_{sim}}$ is called twice on the exact same patterns, then the same value has to be returned (so it is necessary to store the value although this is not done here to preserve simplicity). Finally, $vsim(pat_0, pat_1)$ returns $vrec_{\theta_{sim}}(pat_0, pat_1)$.

The experiments involving $\mathcal{B}$ and $\mathcal{A}$ are the same and as $PrRand$ is equal to $1/2$ for both criteria, the advantages of $\mathcal{B}$ and $\mathcal{A}$ are equal. □ ■ The second step is to show that 1-RPAT$_c$-CPA implies n-RPAT$_c$-CPA for any n.

Lemma A.2 *If $\mathcal{AE}$ is secure w.r.t.* 1*-RPAT$_c$-CPA then it is secure w.r.t. n-RPAT$_c$-CPA.*

Proof: Let us consider an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against n-RPAT$_c$-CPA.

Using the reduction Theorem 1.1, we split up the advantage between an advantage against $(n-1)$-RPAT$_c$-CPA and an advantage against 1-RPAT$_c$-CPA. We assume that adversary $\mathcal{A}$ accesses the left-right oracle F exactly once. Let k be a maximal key for $<$. The partition of θ is defined as follows: θ_1 contains key pairs k, k^{-1}, any nonce N such that $(k, N) \in E_<$. On the other hand, θ_2 contains the other informations from θ including the challenge bit. Oracle F_2 generates the encodings related to keys in θ_2 and F_1 those related to k.

As k is maximal, there are no keys k' different from k such that for a nonce N in θ_1 and any nonce N', $(k, N) < (k', N')$. This is why nonce N is only used as the random coins of an encryption using k.

F_1 can be separated into two layers G and H defined by:

$$H(\langle pat_0, pat_1\rangle, \theta_2, \theta_2') = \big\langle v(pat_{b_2}, \theta_2), v(pat_{b_2'}, \theta_2')\big\rangle$$

$$G(\langle pat_0, pat_1\rangle, b, \theta_1) = v(pat_b, \theta_1)$$

Where b_2 and b_2' are the challenge bits contained respectively in θ_2 and θ_2'.

Let pat_0 and pat_1 be two R-patterns such that $obs_a(pat_0) = obs_a(pat_1)$. Then both patterns are the concatenation of encodings and similar bit-strings. The call to F has to be simulated using F_1 and F_2. For that purpose, the valuation of their encodings is performed in a similar way as in $vrec$ except that F_1 and F_2 should only be called once. To achieve this, requests to F_1 and F_2 are stored in a single pattern as described for F_1 by function $vrec2$ which outputs a list of pair of patterns:

$$\begin{array}{rcl} vrec2(bs, bs) & = & [\,] \\ vrec2(m_1.m_2, m_1'.m_2') & = & vrec2(m_1, m_1').vrec2(m_2, m_2') \\ vrec2(\{m; N\}_k, \{m'; N\}_k) & = & \langle \{m; N\}_k, \{m'; N\}_k\rangle \end{array}$$

Then this list of pair $(\langle p_1, p_1'\rangle; ...; \langle p_n, p_n'\rangle)$ is transformed into the pair of list $\langle p_1; ...; p_n, p_1'; ...; p_n'\rangle$ which is the argument given to F_1. Another function should perform the same operation for keys different from k. After submitting the results to oracle F_1 and F_2, it it easy to rebuild the output of F.

Note that patterns submitted to F_1 and F_2 are pairs of encodings. F_2 receives two well-formed patterns that have the same obs_a and this is the same thing for G (both receives a concatenation of some $\diamondsuit^N$).

As F_2 only depends on θ_2, our partition is valid, criterion $(\theta_2; F_2; V_2)$ is $(n-1)$-RPAT-CPA and $(\theta_1, b; G; V_b)$ is 1-RPAT-CPA. The reduction theorem applies and gives that there exist two PRTM $\mathcal{A}^o$ and $\mathcal{B}$ such that

$$|\mathbf{Adv}_{\mathcal{A}}^{n-RPAT}(\eta)| \leq 2.|\mathbf{Adv}_{\mathcal{B}}^{1-RPAT}(\eta)| + |\mathbf{Adv}_{\mathcal{A}^o}^{(n-1)-RPAT}(\eta)|$$

A simple induction proves that as $\mathcal{AE}$ is secure against 1-RPAT$_c$-CPA, it is secure against n-RPAT$_c$-CPA for any integer n. □ ■ Finally, we show that n-RPAT$_c$-CPA implies n-RPAT-CPA.

Lemma A.3 *If $\mathcal{AE}$ is secure w.r.t. n-RPAT$_c$-CPA then it is secure w.r.t. n-RPAT-CPA.*

Proof: Let us consider an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against n-RPAT-CPA. As in step 1, an adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ is built such that $\mathcal{B}_1$ returns clean patterns. For that purpose, $\mathcal{B}_2$ is similar to $\mathcal{A}_2$. $\mathcal{B}_1$ executes $\mathcal{A}_1$ and computes $dec(pat_0, pat_1)$. Then it generates some keys and nonces and uses them for elements of dec on the answer of pat_0 and pat_1. The patterns remain well-formed and still have the same obs_a. $\mathcal{B}$ and $\mathcal{A}$ have the same advantage but $\mathcal{B}$ is an adversary against n-RPAT$_c$-CPA. As $\mathcal{AE}$ is secure against n-RPAT$_c$-CPA, it is also secure against n-RPAT-CPA. □ ■ Proposition 6.1 is a simple consequence of the three above lemma.

Proposition *If an asymmetric encryption scheme is secure against IND-CPA, then it is secure against n-RPAT-CPA for any number of keys n.*

A.5 Proof of the Main Theorem

This theorem is an application of the reduction theorem 1.1. Let Δ_s be a specification. Let n be the maximal number of keys used by Δ_s. Then the experiment related to $obs_c \times obs_a$ can be reformulated as the following experiment:

- Θ is split up on two parts: Θ_1 generates the n pairs of keys (pk_i, sk_i) and l nonces n^i; Θ_2 generates the initial state s in S and the pattern $p = \Delta_s(s)$.
- We have two oracles: F_2 gives access to $obs_a(p)$, F_1 gives access to $v(p, \theta_1)$ which is $obs_c(p)$.
- V_2 verifies that the output b made by the adversary is equivalent to $s \in \phi$.

F_1 can be be cut in two layers. G corresponds to the left-right encryption algorithm for n-RPAT-CPA, $H(x, \theta_2, \theta_2')$ takes any argument as input x and outputs the pair $\langle p', p\rangle$ where p and p' are the patterns respectively contained in θ_2 and θ_2'.

It is now possible to apply the reduction theorem 1.1 to obtain that for each adversary $\mathcal{A}$, there exist two adversaries $\mathcal{A}^o$ and $\mathcal{B}$ such that

$$|\mathbf{Adv}_{\mathcal{A}}^{\gamma}| \leq |\mathbf{Adv}_{\mathcal{A}^o}^{\gamma_2}| + 2.|\mathbf{Adv}_{\mathcal{B}}^{\gamma_1}|$$

Moreover, γ is equivalent to the criterion related to $obs_c \times obs_a$, γ_2 is equivalent to the one related to obs_a. Finally, $\gamma_1 = (b, \theta_1; G; \lambda x.x = b)$, G is only the left-right oracle, hence this criterion is the n-RPAT-CPA criterion except that there is no oracle to view the public keys. As this criterion is weaker than n-RPAT-CPA, it is possible to conclude that with a different machine $\mathcal{B}$ (but still of comparable complexity),

$$|\mathbf{Adv}_{\mathcal{A}}^{obs_c \times obs_a}| \leq |\mathbf{Adv}_{\mathcal{A}^o}^{obs_a}| + 2.|\mathbf{Adv}_{\mathcal{B}}^{n-RPAT-CPA}|$$