GM-Security and Semantic Security Revisited

Yevgeniy Dodis Matthias Ruhl
MIT Laboratory for Computer Science
Cambridge MA, 02139, USA
{yevgen,ruhl}@theory.lcs.mit.edu

February 5, 1999

Abstract

We give a simple proof that GM-security and semantic security are equivalent security notions for public key cryptosystems. By this we drastically simplify the original proof given by Goldwasser, Micali et al.

1 Introduction

What does it mean for a public key cryptosystem to be secure? In their seminal paper on notions of security for public-key cryptosystems [2], Goldwasser and Micali introduced two security definitions which are still the mostly used ones today. The two notions are called 'GM-security' (also called polynomial security or indistinguishability) and 'semantic security'. Shortly thereafter Micali et al [3] showed that these two notions (and a third one, Y-security introduced by Yao [4]) actually coincide. By far the most involved part of their proof is to show that semantic security implies GM-security, i.e. all public key cryptosystems secure according to the former definition are also secure according to the latter definition. This has led to the view that these two notions of security are in some sense very "different", and the proof is almost never taught in cryptography classes due to its complexity.

In this paper, we show that by a slight (and very reasonable) modification of the definition of semantic security, the proof that semantic security implies GM-security becomes very intuitive and compact. This much shorter proof not only makes the relationship between these two security notions much clearer, it also provides a more "efficient" reduction from one to the other.

The rest of this paper is organized as follows. In section 2 we formally define "GM-security" and "semantic security" for public key cryptosystems. We also detail how our definition of semantic security differs from Goldwasser and Micali original definition in [2], and discuss why this change is reasonable. In section 3 we prove the two notions equivalent. We conclude the paper in section 4 with a discussion of the results.

2 Definitions

For the rest of this paper we follow the notation introduced in [3].

2.1 Public Key Cryptosystems

Suppose we are given a family $\mathcal{M} = \{M_n \mid n \in \mathbb{N}\}$ of message spaces with associated probability distributions. A *public key cryptosystem* for $\mathcal{M}$ is a probabilistic polynomial time (PPT) algorithm $\mathcal{C}$, that on input 1^n outputs two polynomial-size circuits E and D, such that the following holds:

- E is probabilistic and D is deterministic
- E takes an input from M_n and outputs a string from $\{0,1\}^l$, D takes an input from $\{0,1\}^l$, and outputs an element from M_n, for some l.
- For all $m \in M_n$, $\Pr(D(\alpha) = m \mid (E,D) \leftarrow \mathcal{C}(1^n);\ \alpha \leftarrow E(m)) = 1$.

2.2 GM-Security

This definition (found in [3]) is essentially what Goldwasser and Micali [2] called *polynomial security.* It is also called *indistinguishability.*

Definition 1 (GM-Security)
A public-key cryptosystem $\mathcal{C}$ is called GM-secure *if for all $c > 0$, for all families of polynomial-size probabilistic circuits T_n that take four inputs and output 0 or 1, and for all sufficiently large n, the following holds:*

$$\forall m_0, m_1 \in M_n \ : \ \Pr(T_n(E, m_0, m_1, \alpha) = i \mid i \leftarrow \{0,1\};\ E \leftarrow \mathcal{C}(1^n);\ \alpha \leftarrow E(m_i)) < \frac{1}{2} + n^{-c}.$$

2.3 Semantic Security

Definition 2 (Polynomially verifiable)
We call a function family $\mathcal{F} = \{f_n : M_n \to \Sigma^*\}$ polynomially verifiable *if there is some k such that $|f(x)| < n^k$ for all $x \in M_n$, and there exists a family of polynomial-size probabilistic circuits V_n such that for all $x \in M_n$: $V_n(x, f(x)) = 1$, and $V_n(x,y) = 0$ if $y \neq f(x)$.*

Definition 3 (Semantic Security)
Let $\mathcal{C}$ be a public-key cryptosystem, and let $\mathcal{M} = \{M_n\}$ be a sequence of message spaces. Let $\mathcal{F} = \{f_E \ : \ M_n \to \Sigma^*\}$ be a set of polynomially verifiable functions. Let $p_E := \max\{\sum_{m \in f^{-1}(v)} \Pr_n(m) \mid v \in \Sigma^*\}$. ($p_E$ is the maximum probability with which one could guess $f_E(m)$ knowing only the probability distribution from which m has been drawn.) Let $\tilde{p} = \mathbf{E}[p_e]$ be the expected value of p_E over the random choice of E from $\mathcal{C}(1^n)$.

$\mathcal{C}$ is called semantically secure *if for all message space sequences M, for all polynomially verifiable families of functions $\mathcal{F}$, for every family of polynomial-size probabilistic circuits A_n, for all $c > 0$, and for all sufficiently large n*

$$\Pr(\, A_n(E, \alpha) = f_E(m) \mid m \leftarrow M_n;\ E \leftarrow \mathcal{C}(1^n);\ \alpha \leftarrow E(m)\,) < \tilde{p} + \frac{1}{n^c}. \qquad (1)$$

Our definition differs from Goldwasser and Micali's original definition in [2] only in that we restrict $\mathcal{F}$ to contain polynomially verifiable functions, while the original definition allowed any function. Micali et al put it thus ([3]): "Intuitively, f should be thought of as some particular information about the plaintext that the adversary is going to try to compute from the ciphertext...". We think that our variant of the definition captures this notion of security much better than the original one. What good would it do any adversary to "guess" a function $f_E(m)$ from seeing an encryption of m, if he cannot even verify that his guess is correct. Making the functions f_E at least polynomially verifiable thus leads to a much more natural definition of semantic security.

And for this more natural definition of semantic security we give an elegant proof that it is equivalent to GM-security.

Our result can easily be extended to work for polynomially computable relations R_E instead of functions f_E. This is similar to the definition of semantic security used by Dolev et al [1].

3 Equivalence

In this section we prove that the the introduced notions of security are in fact equivalent. This is originally due to Micali et al [3], but our proof is significantly more concise.

Theorem 1
All cryptosystems $\mathcal{C}$ that are GM-secure are also semantically secure, and vice versa. That is, the two notions of security are equivalent.

3.1 Semantic Security $\implies$ GM-Security

This implication is very easy to prove, and we just provide it for completeness. We prove the contrapositive, i.e. that a cryptosystem which is not GM-secure is also not semantically secure.

Suppose A_n can distinguish two messages m_0 and m_1 from M_n, then if we impose the probability distribution $\Pr_{M_n}(m_0) = \Pr_{M_n}(m_1) = \frac{1}{2}$ on this space, any function f with $f(m_0) = 0, f(m_1) = 1$ can be predicted using A_n. So $\mathcal{C}$ is not semantically secure.

3.2 GM-Security $\implies$ Semantic Security

Again we prove the contrapositive. Assume that $\mathcal{C}$ is not semantically secure, and the function family $\mathcal{F} = \{f_E\}$ can be predicted by the family $A = \{A_n(\cdot, \cdot)\}$ of polynomial-size probabilistic circuits.
Consider the following algorithm $T_n : (E, m_0, m_1, \alpha) \to \{0, 1\}$.

1. Let $\beta \leftarrow A_n(E, \alpha)$.
2. If $\beta = f_E(m_0)$ but $\beta \neq f_E(m_1)$, output 0.
3. If $\beta = f_E(m_1)$ but $\beta \neq f_E(m_0)$, output 1.
4. Otherwise, output a random value from $\{0, 1\}$ with probability $\frac{1}{2}$ each.

The test is very intuitive. We simply run A_n on the challenge α. Since we expect A_n to correctly predict the value of f_E, we compare its output β with $f_E(m_0)$ and $f_E(m_1)$. If exactly one of the tests succeed, we output the corresponding message. Otherwise, we flip a coin as we did not learn anything. Note that T_n runs is polynomially bounded since $\beta \overset{?}{=} f_E(m_0)$ etc. can be tested in polynomial time since f_E is polynomially verifiable.

For specific m_0 and m_1, let

$$q(m_0, m_1) := \Pr[T_n(E, m_0, m_1, \alpha) = i \mid i \leftarrow \{0, 1\}, E \leftarrow \mathcal{C}(1^n), \alpha \leftarrow E(m_i)]$$

be the probability that T_n distinguishes encryptions of m_0 and m_1.

Since the algorithm T_n is symmetric in m_0 and m_1, q equals the expected probability that T_n outputs 0 if α is an encryption of m_0, i.e. without loss of generality we can assume that $i = 0$. Now, our experiment can be viewed as the following. Pick $m_0 \leftarrow M_n$, $E \leftarrow \mathcal{C}(1^n)$, $\alpha \leftarrow E(m_0)$, $\beta \leftarrow A_n(E, \alpha)$. Now we pick a *brand new* message $m_1 \leftarrow M_n$ and run steps 2–4 of T_n. q is the probability that we output 0. Before computing q, we claim that

$$\Pr[\beta = f_E(m_0)] \geq \tilde{p} + \frac{1}{n^c}; \qquad \Pr[\beta = f_E(m_1)] \leq \tilde{p} \tag{2}$$

Indeed, the first bound follows directly from (1), as $\beta \leftarrow A_n(E, \alpha)$ and $\alpha \leftarrow E(m_0)$. For the second bound, we observe that for any *fixed* E, the message m_1 is chosen *independent* of m_0, $\alpha \leftarrow E(m_0)$ and, therefore, $\beta \leftarrow A_n(E, \alpha)$. Hence, for any fixed E the probability that $f_E(m_1)$ equals β is at most the probability that it equals to any pre-specified element, which is at most p_E. Since for a fixed E, our probability is stochastically dominated by p_E, we can take the expectation over E to obtain the claimed bound.

We can now compute the probability q of outputting 0 in the following way (using $\Pr[A \wedge B] + \Pr[A \wedge \bar{B}] = \Pr[A]$):

$$\begin{aligned}
q &= \Pr[\beta = f_E(m_0) \wedge \beta \neq f_E(m_1)] + \frac{1}{2}(\Pr[\beta = f_E(m_0) = f_E(m_1)] + \Pr[\beta \notin \{f_E(m_0), f_E(m_1)\}]) \\
&= \frac{1}{2}\left(\Pr[\beta = f_E(m_0) \wedge \beta \neq f_E(m_1)] + \Pr[\beta = f_E(m_0) \wedge \beta = f_E(m_1)]\right) + \\
&\quad \frac{1}{2}\left(\Pr[\beta = f_E(m_0) \wedge \beta \neq f_E(m_1)] + \Pr[\beta \neq f_E(m_0) \wedge \beta \neq f_E(m_1)]\right) \\
&= \frac{1}{2}\left(\Pr[\beta = f_E(m_0)] + \Pr[\beta \neq f_E(m_1)]\right) = \frac{1}{2} + \frac{1}{2}\left(\Pr[\beta = f_E(m_0)] - \Pr[\beta = f_E(m_1)]\right) \\
&\overset{(2)}{\geq} \frac{1}{2} + \frac{1}{2}\left((\tilde{p} + \frac{1}{n^c}) - \tilde{p}\right) = \frac{1}{2} + \frac{1}{2n^c}
\end{aligned}$$

By the probabilistic method we therefore know that there are two specific messages m_0 and m_1 which are distinguished by T_n. Thus C is not GM-secure. □

4 Conclusion

By giving a short proof for the equivalence of GM-security and semantic security we have shown that these notions are actually much more similar then previously believed. Our reduction is also more efficient, i.e. the prediction advantage decreased only from n^{-c} to $\frac{n^{-c}}{2}$, while it went to n^{-2c} in the original reduction [2]. Finally, it is our hope that this simplified proof can be used in a classroom setting to teach the equivalence between the two security notions.

References

[1] D. Dolev, C. Dwork, M. Noar, "Non-malleable cryptography", Manuscript, December 1998.

[2] S. Goldwasser, S. Micali, "Probabilistic Encryption," *JCSS*, vol. 28, No. 2, April 1984, pp. 270–99.

[3] S. Micali, C. Rackoff, B. Sloan, "The Notion of Security for Probabilistic Encryption," *SIAM Journal of Computing.*

[4] A. C. Yao, "Theory and Applications of Trapdoor Functions (extended abstract)," *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, 1982, pp. 80–91.