EPOC 暗号自己評価書

1　まえがき

本資料では、秘匿通信を目的とする公開鍵暗号方式（暗号スキーム）「EPOC 暗号」の安全性および性能に関する自己評価について述べる。「EPOC 暗号」は、「EPOC-1 暗号」、「EPOC-2 暗号」、「EPOC-3 暗号」という 3 つのバージョンを持つが、それぞれについて評価・比較を行う。

なお、処理速度の評価においては、実装評価に加えて、特定のハードウェアやソフトウェア実装手法に依存しない最も客観的な評価方法である、基本的な演算（特定サイズの剰余乗算）の回数による机上評価を示す。

2　安全性評価

2.1　概要

1. **[EPOC-1 の安全性]**　p- 部分群仮定（定義については、次節を参照）が正しければ、ランダムオラクルモデルの下で、EPOC-1 は適応的選択暗号文攻撃に対して強秘匿（最強の意味で安全）である。

2. **[共通鍵暗号としてバーナム暗号を用いた EPOC-2 の安全性]**　素因数分解仮定が正しければ（もしくは、OU 暗号関数の一方向性を仮定すれば）、ランダムオラクルモデルの下で、EPOC-2（バーナム暗号を用いたもの）は適応的選択暗号文攻撃に対して強秘匿である。

3. **[一般の共通鍵暗号を用いた EPOC-2 の安全性]**　素因数分解仮定が正しく（もしくは、OU 暗号関数の一方向性を仮定し）、利用する共通鍵暗号が受動的攻撃の下で安全ならば、ランダムオラクルモデルの下で、EPOC-2（共通鍵暗号を用いたもの）は適応的選択暗号文攻撃に対して強秘匿である。

   EPOC-2（共通鍵暗号を用いたもの）は、公開鍵暗号と共通鍵暗号を組み合わせたハイブリッド暗号方式であるが、この方式の優位性は、そのハイブリッド暗号方式に対して最強の意味での安全性が保証されている点である。例えば、ここで用いる共通鍵暗号が受動的な意味でのみ安全で能動的な意味では安全でない場合でも、EPOC-2（共通鍵暗号を用いたもの）として統合的に構成されると能動的な意味でも安全な

暗号方式になる。もう一つの EPOC-2 の特長は、転送データに対する認証機能である。つまり、受信者が暗号文を復号する際にデータの正当性の検証をするため、暗号文が途中で改ざんされたり変更された場合には、そのことを検出することが可能となる。

4. **[共通鍵暗号としてバーナム暗号を用いた EPOC-3 の安全性]** Gap- 素因数分解仮定が正しければ（もしくは、OU 暗号関数の Gap- 一方向性を仮定すれば）、ランダムオラクルモデルの下で、EPOC-2（共通鍵暗号としてバーナム暗号を用いたもの）は適応的選択暗号文攻撃に対して強秘匿である。

   注： Gap- 素因数分解仮定は、新たに定義された仮定であるが、もし Gap- 素因数分解仮定が破られると n を素因数分解する問題の困難性と $mod n$ での高次剰余性の判定問題の困難性が等価となる。このようなことは大変起こり難いことと信じられており、Gap- 素因数分解仮定は妥当な仮定と考えられる。また、我々は、Gap- 仮定は（決定 Diffie-Hellman 仮定などの）決定仮定と双対な関係であることを示している [11]。

5. **[一般の共通鍵暗号を用いた EPOC-3 の安全性]** Gap- 素因数分解仮定が正しく（もしくは、OU 暗号関数の Gap- 一方向性を仮定し）、利用する共通鍵暗号が受動的攻撃の下で安全ならば、ランダムオラクルモデルの下で、EPOC-3（一般の共通鍵暗号を用いたもの）は適応的選択暗号文攻撃に対して強秘匿である。

   EPOC-3（共通鍵暗号を用いたもの）は、EPOC-2（共通鍵暗号を用いたもの）と同様に、公開鍵暗号と共通鍵暗号を組み合わせたハイブリッド暗号方式であり、そのハイブリッド暗号方式に対して最強の意味での安全性が保証されている。また、EPOC-2 と同様に、転送データに対する認証機能を保有している。つまり、受信者が暗号文を復号する際にデータの正当性の検証をするため、暗号文が途中で改ざんされたり変更された場合には、そのことを検出することが可能となる。

   EPOC-2 にない EPOC-3 の特徴は、ハイブリッド暗号方式としてセッション的利用方法（つまり、セッション開設時において公開鍵暗号方式を用いた鍵配送が行なわれ、以降の該セッション開設中は最初に配送された鍵を用いて複数回の共通鍵暗号によるデータ暗号化が行なわれる）が可能なことである（「EPOC 暗号仕様書」の 6.4 節を参照）。つまり、EPOC-2 では、鍵配送とデータ暗号が同期して使われるのに対し、EPOC-3 では、鍵を配送した後に、非同期に何回も共通鍵暗号による暗号化を行なうことが可能である。さらに、そのような暗号通信を全体として最強の意味の

安全性（適応的選択暗号文攻撃に対して強秘匿／頑健）が保証されていることを証明できる。

6. **[安全性の比較]**

実用的かつ最強の意味の安全性の証明のついた他の公開鍵暗号としては、Cramer-Shoup 暗号や RSA 暗号に基づく OAEP が知られている。Cramer-Shoup 暗号は、ランダム関数に対する仮定が現実的な汎用一方向性ハッシュ関数である点で優れているが、整数論的な仮定では（基本的な離散対数仮定よりも強い仮定である）Diffie-Hellman 仮定よりもさらに強い DDH 仮定に基づいている。一方、EPOC-2 は、理想的なランダム関数に基づいているものの、整数論的な仮定では基本的な素因数分解仮定に基づいている点で OAEP や Cramer-Shoup 暗号よりも優れている。また、EPOC-3 は、整数論的な仮定が Gap- 素因数分解仮定であり素因数分解仮定よりも強い仮定である。（EPOC-3 は、安全性の仮定では EPOC-2 に比べより強い仮定に基づいているが、復号の処理速度やセッション的利用方法などの性能／機能面で、EPOC-2 よりも優れている。）

表 1: 安全性の比較

方式	安全性 （最強の意味で）	整数論的 仮定	ランダム関数 仮定
EPOC-1	安全性証明つき	p- 部分群	真にランダム
EPOC-2(OTP)	安全性証明つき	素因数分解	真にランダム
EPOC-3(OTP)	安全性証明つき	Gap- 素因数分解	真にランダム
Cramer-Shoup	安全性証明つき	DDH	UOWHF
OAEP-RSA	安全性証明つき	RSA	真にランダム
PKCS#1 Ver.1	攻撃可	–	–
単純 RSA	攻撃可	–	–

（注：OTP は、共通鍵暗号としてバーナム暗号利用を、DDH は、Diffie-Hellman 決定問題を、UOWHF は、汎用一方向性ハッシュ関数を意味する。）

2.2 理論的結果

本節では、EPOC-1, EPOC-2, EPOC-3 の安全性に関する結果を示す [13, 7, 8, 11]。

定義 2.1 $\mathcal{G}$ を *EPOC-1* の鍵生成演算とし、$(n, g, h, pLen, hLen)$ をその公開鍵とする。$b \in \{0,1\}$ と $r\{0,1\}^{hLen}$ はランダムかつ一様に選ばれたとし、$C := g^b h^r \bmod n$ とする。

どのような確率的多項式時間アルゴリズム Adv に対しても、全ての定数 c, 十分に大きな値 $k(= pLen)$ に対して

$$\Pr[Adv(k, hLen, n, g, h, C) = b] < 1/2 + 1/k^c.$$

が成立するとき、p- 部分群問題が難しいと言う。ここで、確率は $\mathcal{G}$ と Adv の確率空間上で取られている。

p- 部分群問題が難しいという仮定を p- 部分群仮定と言う。

定義 2.2 $\mathcal{G}_0$ を $\mathcal{G}_0(k) \to n$, $n = p^2 q$, $|p| = |q| = k$ であるような生成器とし $(p, q$: 素数*)*、n の分布は *EPOC-2* の公開鍵 n と同じとする。素因数分解問題とは、(n, k) を与えられて (p, q) を見つける問題である。

どのような確率的多項式時間アルゴリズム A に対しても、全ての定数 c, 十分に大きな値 k に対して

$$\Pr[A(k, n) = (p, q)] < 1/k^c$$

が成立するとき、素因数分解問題が難しいと言う。ここで、確率は $\mathcal{G}_0$ と A の確率空間上で取られている。

素因数分解問題が難しいという仮定を素因数分解仮定と言う。

定義 2.3 $\mathcal{G}$ を *EPOC-3* の鍵生成演算とし、$(n, g, h, pLen, hLen)$ をその公開鍵とする。$b \in \{0,1\}$ と $r\{0,1\}^{hLen}$ はランダムかつ一様に選ばれたとし、$C := g^b h^r \bmod n$ とする。$p - SG$ を p- 部分群問題に関するオラクルとしたとき、$p - SG$ を使えるどのような確率的多項式時間アルゴリズム Adv^{p-SG} に対しても、全ての定数 c, 十分に大きな値 $k(= pLen)$ に対して

$$\Pr[Adv(k, pLen, hLen, n, g, h) = (p, q)] < 1/k^c.$$

が成立するとき、*Gap*- 素因数分解問題が難しいと言う。ここで、確率は $\mathcal{G}$ と Adv の確率空間上で取られている。

Gap- 素因数分解問題が難しいという仮定を *Gap*- 素因数分解仮定と言う。

定義 2.4 Adv を 2 つの段階をもつ攻撃者とする。最初の段階では、Adv は 2 つの平文対, X_0 と X_1, および状態情報 s を生成する。ここで、$|X_0| = |X_1| \leq (gLen)^a$ *(a:* 定数*)*

とする。第2段階では、Adv は暗号文 $Y := SymEnc(K, X_b)$ を与えられる。ここで、$K \in \{0,1\}^{gLen}$ および $b \in \{0,1\}$ はランダムかつ一様に定められる。

どのような確率的多項式時間アルゴリズム Adv に対しても、全ての定数 c, 十分に大きな値 $gLen$ に対して

$$\Pr[Adv(gLen, X_0, X_1, s, Y) = b] < 1/2 + 1/(gLen)^c$$

が成立するとき、$SymE$ が受動的攻撃に対して安全であるという。ここで、確率は (k, b) と Adv の確率空間上で取られている。

定理 2.5 p- 部分群仮定が正しければ、ランダムオラクルモデルの下で、$EPOC\text{-}1$ は適応的選択暗号文攻撃に対して強秘匿である。

定理 2.6 $EPOC\text{-}2$ で用いる $SymE$ をバーナム暗号とする。$rLen = pLen - 1$, かつ $hLen = (2 + c_0)pLen$ ($c_0 > 0$: 定数) とする。このとき、$n = p^2q$ に対する素因数分解仮定が正しければ、ランダムオラクルモデルの下で、$EPOC\text{-}2$ は適応的選択暗号文攻撃に対して強秘匿である。

定理 2.7 $rLen = pLen - 1$, かつ $hLen = (2 + c_0)pLen$ ($c_0 > 0$: 定数) とする。このとき、$n = p^2q$ に対する素因数分解仮定が正しく、利用する共通鍵暗号が受動的攻撃の下で安全ならば、ランダムオラクルモデルの下で、$EPOC\text{-}2$ は適応的選択暗号文攻撃に対して強秘匿である。

定理 2.8 OU 暗号関数の一方向性を仮定し、利用する共通鍵暗号が受動的攻撃の下で安全ならば、ランダムオラクルモデルの下で、$EPOC\text{-}2$ は適応的選択暗号文攻撃に対して強秘匿である。

定理 2.9 $EPOC\text{-}3$ で用いる $SymE$ をバーナム暗号とする。$hLen = (2+c_0)pLen$ ($c_0 > 0$: 定数) とする。このとき、$n = p^2q$ に対する Gap- 素因数分解仮定が正しければ、ランダムオラクルモデルの下で、$EPOC\text{-}3$ は適応的選択暗号文攻撃に対して強秘匿である。

定理 2.10 $hLen = (2 + c_0)pLen$ ($c_0 > 0$: 定数) とする。このとき、$n = p^2q$ に対する Gap- 素因数分解仮定が正しく、利用する共通鍵暗号が受動的攻撃の下で安全ならば、ランダムオラクルモデルの下で、$EPOC\text{-}3$ は適応的選択暗号文攻撃に対して強秘匿である。

定理 2.11 OU 暗号関数の Gap- 一方向性[1]を仮定し、利用する共通鍵暗号が受動的攻撃の下で安全ならば、ランダムオラクルモデルの下で、$EPOC\text{-}3$ は適応的選択暗号文攻撃に対して強秘匿である。

[1] pSG オラクルを利用できる多項式時間アルゴリズムに対して一方向性が保証できること

補足 1: 上記の安全性の（帰着）結果に関するよりタイトで精密な評価が得られている [7, 8, 11]: 例えば、EPOC-2 を（適応的選択暗号文攻撃に対する強秘匿の観点で）破る攻撃の存在を仮定すると、その攻撃に要する計算量とほぼ同じ計算量で n を素因数分解できることを示せる。

補足 2:（$n = p^2q$ の素因数分解の困難性について）

$n = p^2q$ の素因数分解が $n = pq$ よりも簡単かどうかについては分かってないが、$n = p^2q$ に特化されたいくつかのアルゴリズムが研究されている [14, 15, 16, 1]。しかし、これらのアルゴリズムはいずれも素因数分解の楕円曲線法でのアルゴリズムである。一方、$n = pq$ と $n = p^2q$ のいずれにおいても最高速の素因数分解アルゴリズムが数体ふるい法であり、このアルゴリズムの実行時間は n のサイズに依存して、素因数のサイズには依存しない。以上より、現時点では、$n = p^2q$ のサイズを $n = pq$ のサイズと同等にすれば、同等の安全性をもつものと考えられる。

3 実装評価

3.1 ハードウェアでの実装評価

- **使用したプロセス:**
  セルベース

- **設計環境:**
  Verilog-XL + DesignCompiler

- **リソース使用量:**
  約 25.6KG(@2NAND 面積換算) ＋メモリ (13312bit) 構成：[ランダムロジック＋乗算器× 2 ＋加算器] + [メモリ (13312bit)]

- **速度評価:**
  クロック 30MHz での演算速度 (シミュレーションにより測定)

EPOC-1		EPOC-2		EPOC-3	
暗号	640 ms	暗号	640 ms	暗号	640 ms
復号	960 ms	復号	960 ms	復号	320 ms

ただし、鍵サイズ 1152 bit とする。

3.2 ソフトウェアでの実装評価

- **評価プラットホーム:**
  CPU: Pentium with MMX 266MHz
  OS: Trubo Linux version 4.0

- **記述言語:**
  C言語 (gcc version 2.91.60)
  多倍長整数演算ライブラリとして gnu mp (gmp version 3.0.1) を使用

- **メモリ使用量 (コード量):**

EPOC-1		EPOC-2		EPOC-3	
暗号	44.814 Kbytes	暗号	49.371 Kbytes	暗号	50.838 Kbytes
復号	45.353 Kbytes	復号	50.957 Kbytes	復号	51.888 Kbytes

- **メモリ使用量 (ワークエリア):**

EPOC-1		EPOC-2		EPOC-3	
暗号	488 Kbytes	暗号	472 Kbytes	暗号	472 Kbytes
復号	484 Kbytes	復号	488 Kbytes	復号	488 Kbytes

- **処理速度:**

EPOC-1		EPOC-2		EPOC-3	
暗号	60.0 ms	暗号	53.3 ms	暗号	52.8 ms
復号	86.9 ms	復号	73.7 ms	復号	27.3 ms

  ただし、鍵サイズ 1152 bit とする。

- **データサイズ:**

n のサイズ	1152 bits
$hLen$	160 bits
$gLen$	160 bits
平文長	128 bits
公開鍵ファイルサイズ	694 bytes
秘密鍵ファイルサイズ	304 bytes
暗号文ファイルサイズ	291 bytes(EPOC-1) / 307 bytes(EPOC-2) / 332 bytes(EPOC-3)

- **最適化の有無:**

  コンパイル時の最適化は gcc -O3 を用いた。

  本評価は、サンプルプログラム (call-6) として添付したものを実行した結果である。

  EPOC の $g^R h^r$ の演算を、g^R と h^r を別々に行って求めている。一度に $g^R h^r$ を求める方法をによって、約 1.5 倍高速化することができるが、今回の評価ではこの高速化方法を用いていない。

  また、公開鍵パラメータの g は、ほとんどの場合 2 を使うことが出来る。これを利用してプログラムを $g = 2$ の場合に特化して高速化することが出来るが、今回の評価ではこの高速化方法を用いていない。

  mod n や mod p^2 の演算において、中国人の剰余定理を利用して高速に演算することが出来るが、今回の評価では、この高速化方法を用いていない。

  したがって、チューンを行えば本評価より高速な実装が可能である。

4 性能の机上評価と他代表的方式との比較

EPOC 暗号の性能を、基準となるサイズの剰余乗算の回数に換算して評価する。整数論的な演算に基づく公開鍵暗号の場合、このような方法による評価が方式の固有の性能を評価する最も客観的な方法と考えられる。つまり、どのようなハードウエアやソフトウエアを用いて実装しても、ここで得られた方式間の相対的な性能比はほぼ普遍であると考えられる。

ここでは、典型的な 2 種類のパラメータの場合について評価を行なう。一つは、性能を重視し、安全性の仮定を強めた場合である（定理 2.8, 定理 2.11）。もう一つは、定理 2.7, 定理 2.10 等の条件にあうようにパラメータを設定した場合である。

4.1 強い安全性仮定の下でのパラメータの場合

公開鍵暗号を利用する典型的な環境（公開鍵暗号は 128 ビット程度の鍵の配送のみに用いる）において、EPOC-1, EPOC-2 および EPOC-3 の典型的なパラメータ設定は以下となる（そのときの安全性は、定理 2.8, 定理 2.11で保証される）。EPOC-1 の場合、$mLen = 128, rLen = 80, hLen = 208$、EPOC-2（バーナム暗号利用）の場合、$rLen = 128$, $gLen = 128$, $hLen = 128$、また EPOC-3 （バーナム暗号利用）の場合、$RLen = 128$, $gLen = 128$, $rLen = hLen = 128$、とする。さらに、EPOC-1, EPOC-2, EPOC-3 においては、n のサイズを 1152 bits とし、比較のための OAEP-RSA の場合の n を 1152 bits、e を $2^{32} + 1$ とする。

また、各方式とも、ハッシュの処理量及び加算等の処理量は剰余乗算の処理量に比べほぼ無視できるため、ここでは剰余乗算の回数のみで比較を行なう。さらに、EPOC の暗号処理では、べき乗計算に拡張バイナリ法（2 基底のバイナリ法）を用いることとし、EPOC, RSA の復号計算では中国剰余定理[2]を用いるものとする。

このとき、EPOC-1, EPOC-2, EPOC-3 ならびに OAEP-RSA の処理量ならびにデータサイズを以下に示す。なお、以下で $\#M(1152)$ は 1152 bits の法の下での剰余乗算の回数を意味する。

表 2: 処理量等の比較（強い安全性仮定の下でのパラメータ）

方式	暗号化 ($\#M(1152)$)	復号化 ($\#M(1152)$)	鍵長 ($\|n\|$) (bits)	暗号文長 ($\|C\|$) (bits)
EPOC-1	364	266	1152	1152
EPOC-2(OTP)	224	188	1152	1280
EPOC-3(OTP)	224	64	1152	1408
OAEP-RSA	33	432	1152	1152

4.2 弱い安全性仮定の下でのパラメータの場合

公開鍵暗号を利用する典型的な環境（公開鍵暗号は 128 ビット程度の鍵の配送のみに用いる）において、定理 2.7, 定理 2.10 等の条件にあうように設定した EPOC-1, EPOC-2 および EPOC-3 の典型的なパラメータ設定は以下となる。EPOC-1 の場合、$mLen =$

[2]EPOC では、中国剰余定理の特殊形として、$\mathrm{mod} p^2$ の演算を $modp$ 演算に基づき計算する手法を用いる

128, $rLen = 80$, $hLen = 832$、EPOC-2（バーナム暗号利用）の場合、$rLen = 128$, $gLen = 128$, $hLen = 832$、また EPOC-3 （バーナム暗号利用）の場合、$RLen = 128$, $gLen = 128$, $rLen = 832$, $hLen = 128$、とする。さらに、EPOC-1, EPOC-2, EPOC-3 においては、n のサイズを 1152 bits とし、比較のための OAEP-RSA の場合の n を 1152 bits、e を $2^{32}+1$ とする。

その他の仮定は、前節と同じとする。

このとき、EPOC-1, EPOC-2, EPOC-3 ならびに OAEP-RSA の処理量ならびにデータサイズを以下に示す。

表 3: 処理量等の比較（弱い安全性仮定の下でのパラメータ）

方式	暗号化 ($\#M(1152)$)	復号化 ($\#M(1152)$)	鍵長 ($\|n\|$) (bits)	暗号文長 ($\|C\|$) (bits)
EPOC-1	1300	786	1152	1152
EPOC-2(OTP)	1280	775	1152	1280
EPOC-3(OTP)	1280	64	1152	1408
OAEP-RSA	33	432	1152	1152

参考文献

[1] Adleman, L.M. and McCurley, K.S.: Open Problems in Number Theoretic Complexity,II (open problems: C7, O7a and O7b), Proc. of ANTS-I, LNCS 877, Springer-Verlag, pp.291-322 (1995).

[2] Bellare, M., Desai, A., Pointcheval, D., and Rogaway, P.: Relations Among Notions of Security for Public-Key Encryption Schemes, Proc. of Crypto'98, LNCS 1462, Springer-Verlag, pp. 26–45 (1998).

[3] Bellare, M. and Rogaway, P.: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols, Proc. of the First ACM Conference on Computer and Communications Security, pp.62–73 (1993).

[4] Bellare, M. and Rogaway, P. : Optimal Asymmetric Encryption, Proc. of Eurocrypt'94, LNCS 950, Springer-Verlag pp.92-111 (1995).

[5] Bleichenbacher, D.: Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1, Proc. of Crypto'98, LNCS 1462, Springer-Verlag, pp. 1–12 (1998).

[6] Canetti, R., Goldreich, O. and Halevi, S.: The Random Oracle Methodology, Revisited, Proc. of STOC, ACM Press, pp.209–218 (1998).

[7] Fujisaki, E. and Okamoto, T.: How to Enhance the Security of Public-Key Encryption at Minimum Cost, Proc. of PKC'99, Springer-Verlag, LNCS 1560, pp. 53–68 (1999).

[8] Fujisaki, E. and Okamoto, T.: Secure Integration of Asymmetric and Symmetric Encryption Schemes, Proc. of Crypto'99, Springer-Verlag, LNCS 1666, pp. 535–554 (1999).

[9] IEEE P1363 Draft, http://grouper.ieee.org/groups/1363/P1363/draft.html (1999).

[10] Joye, M., Quisquater, J.J., and Yung, M.: On the Power of Misbehaving Adversaries and Security Analysis of EPOC, Manuscript (February 2000).

[11] Okamoto, T. and Pointcheval, D.: OCAC: an Optimal Conversion for Asymmetric Cryptosystems, manuscript (2000).

[12] Okamoto, T. and Pointcheval, D.: EPOC–3: Efficient Probabilistic Public-Key Encryption – V3, submission to P1363a (2000).

[13] Okamoto, T. and Uchiyama, S.: A New Public-Key Cryptosystem as Secure as Factoring, Proc. of Eurocrypt'98, LNCS 1403, Springer-Verlag, pp. 308–318(1998).

[14] Peralta, R.: Bleichenbacher's improvement for factoring numbers of the form $N = PQ^2$ (private communication) (1997).

[15] Peralta, R. and Okamoto, E.: Faster Factoring of Integers of a Special Form, IEICE Trans. Fundamentals, E79-A, 4, pp.489-493 (1996).

[16] Pollard, J.L.: Manuscript (1997).

[17] Rivest, R., Shamir, A. and Adleman,L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM, Vol.21, No.2, pp.120-126 (1978).

Appendix

OCAC: an Optimal Conversion for Asymmetric Cryptosystems

Tatsuaki Okamoto David Pointcheval

要旨

Five years after the optimal asymmetric encryption padding (OAEP) which makes chosen-ciphertext secure encryption scheme from any trapdoor one-way permutation (but whose unique application is RSA), this paper presents OCAC, an optimal conversion which applies to any weakly secure cryptosystem: the overload is negligible, since it just consists, as with OAEP, of two hashings for both encryption and decryption. Furthermore, advantages of OCAC beyond OAEP are numerous:

1. it is more general than OAEP, since it can apply to any partially trapdoor one-way function (RSA and modular square, but also Diffie-Hellman, Higher Residues, etc);
2. it is possible to integrate symmetric encryption (block and stream ciphers) to reach very high speed rates;
3. it also provides a key distribution with session key encryption which achieves chosen-ciphertext security with an only semantically secure symmetric scheme.

Therefore, OCAC could become a new alternative to OAEP, and even reach security relative to factorization.

In addition, in order to clarify the security requirement of the underlying asymmetric encryption, this paper introduces a novel class of computational problems, the *gap problems*, which is considered to be dual to the class of the *decision problems*. We show the relationship among inverting problems (*e.g.*, computational-DH problem), decision problems (*e.g.*, decision-DH problem), and gap problems (*e.g.*, gap-DH problem).

1 Introduction

For a long time many conversions from a weakly secure encryption into a chosen-ciphertext secure cryptosystem have been attempted, with variable success. Such a

goal is of greatest interest since many one-way encryption schemes are known, with variable efficiency and various properties, whereas chosen-ciphertext secure schemes are very rare.

1.1 Chosen-Ciphertext Secure Cryptosystems

Until few years ago, the description of a cryptosystem, together with some heuristic arguments for security, were enough to convince and to make a scheme to be widely adopted. Formal semantic security [15] and further non-malleability [11] were just seen as theoretical properties. However, after multiple cryptanalyses of international standards [5, 8, 7], provable security has been realized to be important and even became a basic requirement for any new cryptographic protocol. Therefore, for the last two years, many cryptosystems have been proposed. Some furthermore introduced new problems [17, 21, 18, 23, 26], other are intricate constructions, over old schemes, to reach chosen-ciphertext security (from El Gamal [33, 32, 9, 1, 20], Okamoto-Uchiyama [22], D-RSA [25] or Paillier [24]), with specific security proofs.

Indeed, it is easy to describe a one-way cryptosystem from any trapdoor problem. Furthermore, such trapdoor problems are not so rare (Diffie-Hellman [10], factorization, RSA [29], elliptic curves, McEliece [16], etc). A very nice result would be a generic and *efficient* conversion from any such trapdoor problem into a chosen-ciphertext secure encryption scheme.

1.2 Related Work

In 1994, Bellare and Rogaway [3] suggested such a conversion, the so-called OAEP (Optimal Asymmetric Encryption Padding). However, its application domain was restricted to trapdoor *permutations*, which is a very rare object (RSA seems to be the only one application). Nevertheless, it provided the most efficient RSA-variant, the OAEP-RSA scheme, provably chosen-ciphertext secure, and became the new RSA standard – PKCS #1 [30].

At PKC '99, Fujisaki and Okamoto [13] proposed another conversion with further improvements [14, 27]. It therefore seemed that the expected goal was reached: a generic conversion from any one-way cryptosystem into a chosen-ciphertext secure encryption scheme. However, the resulting scheme is not optimal, from the compu-

tational point of view. Namely, the decryption phase is more heavy than one could expect, since it requires a re-encryption.

As a consequence, with those conversions, one cannot expect to obtain a scheme with an easy decryption phase (unless both encryption and decryption are easy, which is very unlikely). However, decryption is usually implemented on a smart card, hence efficient decryption process is a challenge with a practical impact.

1.3 Achievement: a New and Optimal Conversion

The present work provides a new conversion which is optimal in both the encryption and decryption phases. Indeed, the encryption needs an evaluation of the one-way function, and the decryption just makes one call to the inverting function. Further light computations are to be done, but just an XOR and two hashings. Moreover, many interesting features appear with integration of symmetric encryption schemes.

The aim of the new conversion is very natural: it roughly first encrypts a session key using the asymmetric scheme, and then encrypts the plaintext with any symmetric encryption scheme, which is *semantically-secure* under simple passive attacks (possibly the one-time pad), using the session key as secret key. Of course this simple and actually used scheme does not reach chosen-ciphertext security, but just making the session key more impredictable and adding a checksum, it can be made so:

$$C = \mathcal{E}_{\mathsf{pk}}^{\mathsf{asym}}(R) \quad (1)$$

$$K = G(R) \quad (2)$$

$$\mathcal{E}_{\mathsf{pk}}(m) = C||\mathcal{E}_K^{\mathsf{sym}}(m)||H(C,R,m), \quad (3)$$

where G and H are any hash functions.

Moreover, if one uses a semantically secure symmetric encryption scheme against basic passive attacks (no known-plaintext attacks), the last part of the ciphertext, which is very fast since it only makes calls to a hash function and to a symmetric encryption, can be used more than once, with many messages. This makes a highly secure use of a session key, with symmetric encryption $\mathcal{E}^{\mathsf{sym}}$ which initially just meets a very weak security property:

$$C = \mathcal{E}_{\mathsf{pk}}^{\mathsf{asym}}(R)$$

$$K = G(R)$$

$$\mathcal{E}_{\mathsf{pk}}(m_i) \quad = \quad C||\mathcal{E}_K^{\mathsf{sym}}(m_i)||H(C,R,m_i) \text{ for } i = 1,\ldots.$$

1.4 Outline of the Paper

We first review, in Section 2, the security notions about encryption schemes (both symmetric and asymmetric) required in the rest of the paper, with namely the semantic security. Then, in the next section (Section 3), we describe a new attack scenario, we call the Plaintext-Checking Attack. In Section 4, we develop a novel class of problems, the Gap-Problems. Then in Section 5, we describe our new Optimal Conversion together with the security proofs, relative to the above gap-problems. The next section (Section 6) presents some interesting applications of this conversion. Then comes the conclusion.

2 Security Notions for Encryption Schemes

2.1 Asymmetric Encryption Schemes

In this part, we formally define public-key encryption schemes, together with the security notions.

Definition 2.1 (Asymmetric Encryption Schemes) *An asymmetric encryption scheme, on a message space $\mathcal{M}$, consists of 3 algorithms $(\mathcal{K}^{\mathsf{asym}}, \mathcal{E}^{\mathsf{asym}}, \mathcal{D}^{\mathsf{asym}})$:*

- *the key generation algorithm $\mathcal{K}^{\mathsf{asym}}(1^k)$ outputs a random pair of secret-public keys $(\mathsf{sk}, \mathsf{pk})$, relatively to the security parameter k;*
- *the encryption algorithm $\mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(m;r)$ outputs a ciphertext c corresponding to the plaintext $m \in \mathcal{M}$ (using the random coins $r \in \Omega$);*
- *the decryption algorithm $\mathcal{D}^{\mathsf{asym}}_{\mathsf{sk}}(c)$ outputs the plaintext m associated to the ciphertext c.*

Remark:

As written above, $\mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(m;r)$ denotes the encryption of a message $m \in \mathcal{M}$ using the random coins $r \in \Omega$. When the random coins are useless in the discussion, we simply note $\mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(m)$.

The basic security notion required from an encryption scheme is the *one-wayness*, which roughly means that, from the ciphertext, one cannot recover the whole plaintext.

Definition 2.2 (One-Way) *An asymmetric encryption scheme is said to be* one-way *if no polynomial-time attacker can recover the whole plaintext from a given ciphertext with non-negligible probability. More formally, an asymmetric encryption scheme is said* (t, ε)*-INV if for any adversary* $\mathcal{A}$ *with running time bounded by* t*, its inverting probability is less than* ε*:*

$$\mathsf{Succ}^{\mathsf{inv}} = \Pr[(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathcal{K}^{\mathsf{asym}}(1^k), m \stackrel{R}{\leftarrow} \mathcal{M}, r \stackrel{R}{\leftarrow} \Omega : \mathcal{A}(\mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(m; r)) = m] < \varepsilon.$$

A by now more and more required property is the *semantic security* [15] also known as *indistinguishability of encryptions* or *polynomial security* since it is the computational version of perfect security [31].

Definition 2.3 (Semantic Security) *An asymmetric encryption scheme is said to be* semantically secure *if no polynomial-time attacker can learn any bit of information about the plaintext from the ciphertext, excepted the length. More formally, an asymmetric encryption scheme is said* (t, ε, ℓ)*-IND if for any adversary* $\mathcal{A} = (A_1, A_2)$ *with running time bounded by* t*,*

$$\mathsf{Adv}^{\mathsf{ind}} = 2 \cdot \Pr\left[\begin{array}{ll} (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathcal{K}^{\mathsf{asym}}(1^k) & \\ (m_0, m_1, s) \leftarrow A_1(\mathsf{pk}), & : A_2(c, s) = b \\ b \stackrel{R}{\leftarrow} \{0,1\}, r \stackrel{R}{\leftarrow} \Omega, c \leftarrow \mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(m_b; r) & \end{array}\right] - 1 < \varepsilon,$$

where m_0 *and* m_1 *are both* ℓ*-bit long.*

Both notions are denoted INV and IND respectively in the following.

Another security notion has been defined, called *non-malleability* [11]. It roughly means that it is impossible to derive, from a given ciphertext, a new ciphertext such that the plaintexts are meaningfully related. But we won't detail it since this notion has been proven equivalent to semantic security against parallel attacks [4].

Indeed, the adversary considered above may obtain, in some situations, more informations that just the public key. With just the public key, we say that she plays a *chosen–plaintext attack* since she can encrypt any plaintext of her choice, thanks to the public key. It is denoted CPA. But she may, for some time, access a decryption oracle. She then plays a *chosen–ciphertext attack*, which is either *non-adaptive* [19]

if this access is limited in time, or *adaptive* [28] if this access is unlimited, and the adversary can therefore ask any query of her choice to the decryption oracle, but of course she is restricted not to use it on the challenge ciphertext.

It has already been proven [2] that under this latter attack, the adaptive chosen-ciphertext attacks, denoted CCA, the semantic security and the non-malleability notions are equivalent, and is the strongest security notion that one could expect. We therefore call this security level in this scenario the *chosen–ciphertext security.*

2.2 Symmetric Encryption Schemes

In this part, we briefly focus on symmetric encryption schemes.

Definition 2.4 (Symmetric Encryption Schemes) *A symmetric encryption scheme, on a message space $\mathcal{M}$, consists of 3 algorithms $(\mathcal{K}^{\mathsf{sym}}, \mathcal{E}^{\mathsf{sym}}, \mathcal{D}^{\mathsf{sym}})$:*

- *the key generation algorithm $\mathcal{K}^{\mathsf{sym}}(1^k)$ outputs a random key k, relatively to the security parameter k;*
- *the encryption algorithm $\mathcal{E}^{\mathsf{sym}}_{\mathsf{k}}(m)$ outputs a ciphertext c corresponding to the plaintext $m \in \mathcal{M}$, in a deterministic way;*
- *the decryption algorithm $\mathcal{D}^{\mathsf{sym}}_{\mathsf{k}}(c)$ gives back the plaintext m associated to the ciphertext c.*

As for asymmetric encryption, impossibility for any adversary to get back the whole plaintext just given the ciphertext is the basic requirement. However, we directly consider *semantic security.*

Definition 2.5 (Semantic Security) *A symmetric encryption scheme is said to be* semantically secure *if no polynomial-time attacker can learn any bit of information about the plaintext from the ciphertext, excepted the length. More formally, a symmetric encryption scheme is said (t, ε, ℓ)-IND if for any adversary $\mathcal{A} = (A_1, A_2)$ with running time bounded by t,*

$$\mathsf{Adv}^{\mathsf{ind}} = 2 \times \Pr\left[\begin{array}{ll} \mathsf{sk} \leftarrow \mathcal{K}^{\mathsf{sym}}(1^k) & \\ (m_0, m_1, s) \leftarrow A_1(k), & : A_2(c, s) = b \\ b \stackrel{R}{\leftarrow} \{0,1\}, c \leftarrow \mathcal{E}^{\mathsf{sym}}_{\mathsf{k}}(m_b) & \end{array}\right] - 1 < \varepsilon,$$

where m_0 and m_1 are both ℓ-bit long.

In the basic scenario, the adversary just sees some ciphertexts, but nothing else. However, many stronger scenarios can also be considered. The first which seemed natural for public-key cryptosystems are the known/chosen-plaintext attacks, where the adversary sees some plaintext-ciphertext pairs with the plaintext possibly chosen by herself. These attacks are not trivial in the symmetric encryption setting, since the adversary is unable to encrypt herself.

The stronger scenario considers the adaptive chosen-plaintext/ciphertext attacks, where the adversary has access to both an encryption and a decryption oracle.

However, just the security against the basic no-plaintext/ciphertext attacks (a.k.a. passive attacks) is enough in our application. Therefore, one can remark that it is a very weak requirement. Indeed, if one considers AES candidates, cryptanalysts even fail in breaking efficiently semantic security using adaptive chosen plaintext/ciphertext attacks: with respect to pseudo-random permutations, semantic security is equivalent to say that the family $(\mathcal{E}_{\mathsf{k}}^{\mathsf{sym}})_{\mathsf{k}}$ is (t, ε)-indistinguishable from the uniform distribution on all the permutations over $\{0,1\}^{\ell}$, after just one query (*cf.* universal hash functions [6])!

Remark:

One should remark that the one-time pad provides a perfect semantically secure symmetric encryption: if $\mathcal{K}^{\mathsf{sym}}(1^k)$ outputs k-bit long secret key, then for any t it is $(t, 0, k)$-semantically secure.

3 The Plaintext-Checking Attacks

We have recalled above all the classical security notions together with the classical scenarios of attacks in the asymmetric setting. A new kind of attacks (parallel attacks) has been recently defined [4], which have no real practical meaning, but the goal was just to deal with non-malleability. In this paper, we define a new one, where the adversary can check whether a message-ciphertext pair (m, c) is valid: the *Plaintext-Checking Attack.*

Definition 3.1 (Plaintext-Checking Attack) *The attacker has access to a* Plaintext-Checking Oracle *which takes as input a plaintext m and a ciphertext c and outputs 1 or 0 whether c encrypts m or not.*

It is clear that such an oracle is less powerful than a decryption oracle. This scenario will be denoted by PCA, and will be always assumed to be fully adaptive: the attacker has always access to this oracle without any restriction: she can even include the challenge ciphertext in the query. Therefore, it is clear that semantic security under this attack cannot be reached. But we don't mind, since we just require a scheme to be *one-way* in this scenario. It is a very weak notion.

Remark:

One can remark that any deterministic IND-CPA asymmetric encryption scheme is clearly still IND-PCA. Namely, any trapdoor one-way permutation provides a IND-PCA-secure encryption scheme (*e.g.* RSA [29]).

4 Gap Problems

The attacking problem under the above-mentioned *Plaintext-Checking Attack* can be characterized by a novel class of computational problems, the *gap problems*.

We first define the gap problems as well as the related inverting and decision problems. Then we give some examples.

4.1 Definitions

Let $f : \{0,1\}^* \times \{0,1\}^* \mapsto \{0,1\}$ be any binary relation. The two classical problems are the following:

- the *inverting problem* of f is, given x, to compute any y such as $f(x,y) = 1$ if it exists, or to answer Fail.
- the *decision problem* (type 1) of f is, given a pair (x,y), to decide whether $f(x,y) = 1$ or not.
- the *decision problem* (type 2) of f is, given x, to decide whether there exists some y such that $f(x,y) = 1$ or not.

In this section, we define the *gap problems*.

Definition 4.1 (Gap Problem) *The gap problem (type 1 or 2) of f is to solve the inverting problem of f with the help of the oracle of f's decision problem (type 1 or 2, respectively).*

Let us also define some notations:

- a problem X is *tractable* if it can be solved with non-negligible probability by some probabilistic polynomial time Turing machine.
- a problem X is *strongly tractable* if it can be solved with overwhelming probability by some probabilistic polynomial time Turing machine.

Therefore, we have the negation:

- a problem X is *intractable* if it is not *tractable*
- a problem X is *weakly intractable* if it is not *strongly tractable*.

Finally, to compare the difficulty of problems, we use the notion of polynomial reductions:

- a problem X is *reducible* to problem Y if there exists a probabilistic polynomial time oracle Turing machine A^Y (with oracle of problem Y) to compute X with non-negligible probability.
- a problem X is *strongly reducible* to problem Y if there exists a probabilistic polynomial time oracle Turing machine A^Y (with oracle of problem Y) to compute X with overwhelming probability.

We can easily obtain the following proposition,

Proposition 4.2 *Let f be any binary relation.*

- *If the gap problem of f is tractable (resp. strongly tractable), the inverting problem of f is reducible (resp. strongly reducible) to the decision problem of f.*
- *Let us assume that all the defined problems, based on f, are uniformly easy or difficult. If the decision problem of f is strongly tractable, the inverting problem of f is reducible to the gap problem of f.*

Proof:

The first claim directly comes from the definition of the gap problem. Let us consider the second claim, with a probabilistic polynomial time Turing machine A that solves the decision problem of f, with overwhelming probability. Let us also assume that

we have a probabilistic polynomial time oracle Turing machine B^D that solves the inverting problem of f with the help of a decision oracle D. Since A solves the decision problem with overwhelming probability, it perfectly simulates the D oracle, after polynomially many queries, with non-negligible probability. In these cases, the machine B can invert. [QED] ¶

This proposition implies a duality between the gap and decision problems. In other words, the reasonability (or weakness) of the intractability assumptions of the gap and decision problems of f are comparable, unless one of them is shown to be tractable.

4.2 The Random Self-Reducible Problems

Definition 4.3 *A problem is said* random self-reducible *if any instance can be transformed in an other uniformly distributed instance whose solution helps in solving the initial instance.*

Such problems are clearly uniformly easy or difficult Problems. Furthermore, the weak intractability is equivalent to the classical intractability.

Corollary 4.4 *Let f be any random self-reducible binary relation.*

- *If the gap problem of f is tractable, the inverting problem of f is reducible to the decision problem of f.*
- *If the decision problem of f is tractable, the inverting problem of f is reducible to the gap problem of f.*

Remark:

Almost all the classical problems used in cryptography are *random self-reducible*.

4.3 Examples of Gap Problems

Let us review some of these classical problems, with their gap variations.

Definition 4.5 (The Diffie-Hellman Problems) *Let us consider any group $\mathcal{G}$ of order q together with a generator g. We define three problems as follows:*

- The Inverting Diffie-Hellman Problem *(a.k.a. the Computational Diffie-Hellman problem): given a pair* (g^a, g^b)*, find the element* $C = g^{ab}$*.*
- The Decision Diffie-Hellman Problem*: given a triple* (g^a, g^b, g^c)*, decide whether* $c = ab \bmod q$ *or not.*
- The Gap Diffie-Hellman Problem*: given a pair* (g^a, g^b)*, find the element* $C = g^{ab}$ *with the help of a Decision Diffie-Hellman Oracle (which answers whether a given triple is correct or not).*

Note that these decision and gap problems are of type 1, where

$$f((A,B),C) \stackrel{\text{def}}{=} \left(\log_g C \stackrel{?}{=} \log_g A \times \log_g B \bmod q\right),$$

which is *a priori* not a polynomially computable function.

Definition 4.6 (The Gap-DH Assumption) *For any probabilistic polynomial oracle Turing machine which has access to a Decision-DH oracle, the probability of, given* (g^a, g^b)*, finding* $C = g^{ab}$ *is negligible.*

Since no polynomial time reduction (even a probabilistic one) is known from the Computational-DH to the Decision-DH problems, the Gap-DH assumption seems as reasonable as the Decision-DH assumption due to the duality of these problems (Proposition 4.2). Note that, as for most of the problems in use in cryptography, the Inverting Problem is stronger than the Gap Problem (and the Decision Problem either). Therefore, the tractability of the Gap-DH problem would lead to an equivalence between Computational-DH and Decision-DH (they would be reducible to each other), which is very unlikely.

Definition 4.7 (The Rabin Problems) *Let us consider* $n = pq$*. We define three problems as follows:*

- The Inverting Rabin Problem *(a.k.a. the Factoring Problem): given a pair* (n, y)*, find* $x = y^{1/2} \bmod n$ *if* x *exists.*
- The Decision Rabin Problem *(a.k.a the Quadratic Residuosity Problem): given a pair* (n, y)*, decide whether* x *exists or not.*

- The Gap Rabin Problem: *given a pair* (n,y), *find* $x = y^{1/2} \bmod n$ *if* x *exists, with the help of a Decision Rabin Oracle.*

Note that these decision and gap problems are of type 2, where

$$f(y,x) \stackrel{\text{def}}{=} \left(y \stackrel{?}{=} x^2 \bmod n\right),$$

which is a polynomially computable function.

Since no polynomial time reduction is known from the Factorization to the Quadratic-Residuosity problem, the Gap-Rabin assumption seems as reasonable as the Quadratic-Residuosity assumption.

Definition 4.8 (The RSA Problems) *Let us consider* $n = pq$ *and* e *relatively prime with* $\varphi(n)$. *We define three problems as follows:*

- The Inverting RSA Problem: *given a triple* (n,e,y), *find* $x = y^{1/e} \bmod n$.
- The Decision RSA Problem: *given a quadruple* (n,e,y,x), *decide whether* $x = y^{1/e} \bmod n$.
- The Gap RSA Problem: *given a triple* (n,e,y), *find* $x = y^{1/e} \bmod n$ *with the help of a Decision RSA Oracle.*

Note that these decision and gap problems are of type 1, where

$$f(y,x) \stackrel{\text{def}}{=} \left(y \stackrel{?}{=} x^e \bmod n\right),$$

which is a polynomially computable function. Therefore, it is a really different situation from the Diffie-Hellman problems. They are both type 1 problems, but in the current RSA situation, the function f is polynomially computable. Thus the Decision-problem is clearly strongly tractable (and even more than that since one can always answer correctly). As a consequence, the Gap and Inverting-RSA problems are equivalent.

Definition 4.9 (The Okamoto-Uchiyama Problems) *Let us consider* $n = p^2q$, $g \in \mathbb{Z}_n^\star$ *such that* $g_p^{p-1} \bmod p^2$ *is of order* p, *and* $h = g^n \bmod n$. *We define three problems as follows:*

- The Inverting-OU Problem *(a.k.a. the Factoring Problem): given a quadruple* (n, g, h, y)*, find* $x \in \mathbb{Z}_p^\star$ *such that* $y = g^x h^r \bmod n$*.*
- The Decision-OU Problem *(a.k.a. the High-Residuosity Problem): given a tuple* (n, g, h, y, x)*, decide whether* $y = g^x h^r \bmod n$ *for some* r*, or not.*
- The Gap-OU Problem *(thus called the Gap-High-Residuosity Problem): given a quadruple* (n, g, h, y)*, find* $x \in \mathbb{Z}_p^\star$ *such that* $y = g^x h^r \bmod n$ *with the help of a Decision-OU Oracle.*

Note that these decision and gap problems are of type 1, where f is a first order function:

$$f(y, x) \stackrel{\text{def}}{=} \left(\exists r, y \stackrel{?}{=} g^x h^r \bmod n\right),$$

which is *a priori* not a polynomially computable function.

Definition 4.10 (The Gap-High-Residuosity Assumption) *For any probabilistic polynomial oracle Turing machine, which has access to a High-Residuosity Oracle, the probability of success in factoring is negligible.*

Since no polynomial time reduction from Factorization to the High-Residuosity problem, the Gap-High Residuosity assumption seems as reasonable as the High-Residuosity assumption.

5 Description of the Conversion

5.1 The Basic Conversion

Let us consider $(\mathcal{K}^{\mathsf{asym}}, \mathcal{E}^{\mathsf{asym}}, \mathcal{D}^{\mathsf{asym}})$, any INV-PCA–secure asymmetric encryption scheme, as well as two given hash functions G and H which output k_1-bit strings and k_2-bit strings respectively. Then, the new scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ works as follows:

- Key generation algorithm $\mathcal{K}(1^k)$: it simply runs $\mathcal{K}^{\mathsf{asym}}(1^k)$ to get a pair of keys $(\mathsf{sk}, \mathsf{pk})$, and outputs it.
- Encryption algorithm $\mathcal{E}_{\mathsf{pk}}(m; R, r)$: it gets $c_1 = \mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(R; r)$, then it computes the session key $K = G(R)$, $c_2 = K \oplus m$ as well as $c_3 = H(c_1, R, m)$. The ciphertext consists of the triple $C = (c_1, c_2, c_3)$.

- Decryption algorithm $\mathcal{D}_{\mathsf{sk}}(C)$: from $C = (c_1, c_2, c_3)$, it first extracts R from c_1 by decrypting it: $R = \mathcal{D}^{\mathsf{asym}}_{\mathsf{sk}}(c_1)$. It can therefore recover the session key $K = G(R)$ and $m = K \oplus c_2$ which is output only if $c_3 = H(c_1, R, m)$. Otherwise, it outputs "Reject".

The overload is minimal. Indeed, if we consider the encryption phase, it just adds the computation of two hash values and an XOR. Concerning the decryption phase, which had been made heavy in previous conversions [13, 14, 27] with a re-encryption to check the validity, we also just add the computation of two hash values and an XOR, as in the encryption process.

5.2 The Hybrid Conversion

As it as already been done with some previous conversions [13, 14, 22, 25, 27], the "one-time pad" encryption can be generalized to any symmetric encryption scheme which is not perfectly secure, but semantically secure against passive attacks.

Let us consider two encryption schemes, $(\mathcal{K}^{\mathsf{asym}}, \mathcal{E}^{\mathsf{asym}}, \mathcal{D}^{\mathsf{asym}})$ is a INV-PCA–secure asymmetric scheme and $(\mathcal{K}^{\mathsf{sym}}, \mathcal{E}^{\mathsf{sym}}, \mathcal{D}^{\mathsf{sym}})$ is a IND–secure symmetric scheme which uses k_1-bit long keys, as well as two hash functions G and H which output k_1-bit numbers and k_2-bit numbers respectively. Then, the new scheme $(\mathcal{K}^{\mathsf{hyb}}, \mathcal{E}^{\mathsf{hyb}}, \mathcal{D}^{\mathsf{hyb}})$ works as follows:

- Key generation algorithm $\mathcal{K}^{\mathsf{hyb}}(1^k)$: it simply runs $\mathcal{K}^{\mathsf{asym}}(1^k)$ to get a pair of keys $(\mathsf{sk}, \mathsf{pk})$, and outputs it.

- Encryption algorithm $\mathcal{E}^{\mathsf{hyb}}_{\mathsf{pk}}(m; R, r)$: it gets $c_1 = \mathcal{E}_{\mathsf{pk}}(R; r)$ and a random session key $K = G(R)$. Then it computes $c_2 = \mathcal{E}^{\mathsf{sym}}_K(m)$ as well as the checking part $c_3 = H(c_1, R, m)$. The ciphertext consists of $C = (c_1, c_2, c_3)$.

- Decryption algorithm $\mathcal{D}^{\mathsf{hyb}}_{\mathsf{sk}}(C)$: from $C = (c_1, c_2, c_3)$, it first extracts R from c_1 by decrypting it: $R = \mathcal{D}^{\mathsf{asym}}_{\mathsf{sk}}(c_1)$. It can therefore recover the session key $K = G(R)$ as well as the plaintext $m = \mathcal{D}^{\mathsf{sym}}_K(c_2)$ which is output only if $c_3 = H(c_1, R, m)$. Otherwise, it outputs "Reject".

The overload is similar to the previous, but then, the plaintext can be longer. Such an hybrid transformation cannot be just considered as folklore since the OAEP conversion (which furthermore requires a trapdoor permutation) does not allow symmetric

encryption integration. Furthermore, the required property for the symmetric encryption is very weak. Indeed, as it will be seen during the security analysis in next section, it is just required that the symmetric encryption scheme is semantic security in the basic scenario (no plaintext/ciphertext attacks).

5.3 Chosen-Ciphertext Security

Theorem 5.1 *Let us assume that*

- *the asymmetric encryption scheme* $(\mathcal{K}^{\mathsf{asym}}, \mathcal{E}^{\mathsf{asym}}, \mathcal{D}^{\mathsf{asym}})$ *is* INV-PCA*–secure*[3]
- *and the symmetric encryption scheme* $(\mathcal{K}^{\mathsf{sym}}, \mathcal{E}^{\mathsf{sym}}, \mathcal{D}^{\mathsf{sym}})$ *is* IND*-secure,*

then the conversion $(\mathcal{K}^{\mathsf{hyb}}, \mathcal{E}^{\mathsf{hyb}}, \mathcal{D}^{\mathsf{hyb}})$ *is* IND-CCA *in the random oracle model.*

More precisely, one can claim the following exact security result.

Theorem 5.2 *Let us consider a* CCA*–adversary* $\mathcal{A}^{\mathsf{cca}}$ *against the "semantic security" of the conversion* $(\mathcal{K}^{\mathsf{hyb}}, \mathcal{E}^{\mathsf{hyb}}, \mathcal{D}^{\mathsf{hyb}})$*, between* ℓ*-bit messages, within a time bounded by* t*, with advantage* ε*, after* q_D*,* q_G *and* q_H *queries to the decryption oracle, and the hash functions* G *and* H *respectively. Then for any* $0 < \nu < \varepsilon$*, there either exist*

- *an adversary* $\mathcal{B}^{\mathsf{pca}}$ *against the* (t, φ)*-*INV-PCA*-security of the asymmetric encryption scheme* $(\mathcal{K}^{\mathsf{asym}}, \mathcal{E}^{\mathsf{asym}}, \mathcal{D}^{\mathsf{asym}})$*, after less than* $(g_G + q_H) \cdot (q_D + 1)$ *queries to the Plaintext-Checking Oracle, where*

$$\varphi = \frac{\varepsilon - \nu}{2} - \frac{q_D}{2^{k_2}}$$

- *or an adversary* $\mathcal{B}$ *against the* (t, ν, ℓ)*-*IND*–security of symmetric encryption scheme* $(\mathcal{K}^{\mathsf{sym}}, \mathcal{E}^{\mathsf{sym}}, \mathcal{D}^{\mathsf{sym}})$*.*

Proof:

More than semantically secure under chosen-ciphertext attacks, this converted scheme can be proven "plaintext–aware" [3, 2], which implies chosen-ciphertext security. To prove above Theorems, we first assume that the symmetric encryption scheme $(\mathcal{K}^{\mathsf{sym}}, \mathcal{E}^{\mathsf{sym}}, \mathcal{D}^{\mathsf{sym}})$ is (t, ν, ℓ)-IND–secure, for some probability $0 < \nu < \varepsilon$.

[3] In other words, "If the type 1 Gap problem is intractable (where $f(y, x) = 1$ iff $\mathcal{D}^{\mathsf{asym}}(y) = x$)"

Semantic Security. The semantic security of this scheme intuitively comes from the fact that for any adversary, in order to have any information about the encrypted message m, she at least has to have asked $(c_1, R, \star)$ to H (which is called "event 1" and denoted by E_1) or R to G (which is called "event 2" and denoted by E_2). Therefore, for a given $c_1 = \mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(R; r)$, R is in the list of queries asked to G or H. Then, for any candidate $\tilde{R}$, one asks to the Plaintext Checking Oracle whether c_1 encrypts $\tilde{R}$ or not. The accepted one is output as the inversion of $\mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}$ on the ciphertext c_1, which breaks the INV-PCA.

More precisely, let us consider $\mathcal{A} = (A_1, A_2)$, an adversary against the semantic security of the converted scheme, using an adaptive chosen-ciphertext attack. Within a time bound t, she asks q_D queries to the decryption oracle and q_G and q_H queries to the hash functions G and H respectively, and distinguishes the right plaintext with an advantage greater than ε. Actually, in the random oracle model, because of the randomness of G and H, if neither event 1 nor event 2 happen, she gets $c_2 = \mathcal{E}^{\mathsf{sym}}_K(m_b)$, for a totally random key K and then cannot gain any advantage greater than ν, since the running time is bounded by t and messages are ℓ-bit long. Then,

$$\Pr_b[A_2(\mathcal{E}^{\mathsf{hyb}}_{\mathsf{pk}}(m_b; r), s) = b \,|\, \neg(\mathsf{E}_1 \vee \mathsf{E}_2)] \leq \frac{1}{2} + \frac{\nu}{2}.$$

However,

$$\begin{aligned}
\frac{1}{2} + \frac{\varepsilon}{2} &\leq \Pr_b[A_2(\mathcal{E}^{\mathsf{hyb}}_{\mathsf{pk}}(m_b; r), s) = b] \\
&= \Pr_b[A_2 = b \wedge \neg(\mathsf{E}_1 \vee \mathsf{E}_2)] + \Pr_b[A_2 = b \wedge (\mathsf{E}_1 \vee \mathsf{E}_2)] \\
&= \Pr_b[A_2 = b \,|\, \neg(\mathsf{E}_1 \vee \mathsf{E}_2)] \times \Pr_b[\neg(\mathsf{E}_1 \vee \mathsf{E}_2)] + \Pr_b[A_2 = b \wedge (\mathsf{E}_1 \vee \mathsf{E}_2)] \\
&\leq \frac{1}{2} + \frac{\nu}{2} + \Pr_b[\mathsf{E}_1 \vee \mathsf{E}_2].
\end{aligned}$$

This leads to $\Pr[\mathsf{E}_1 \vee \mathsf{E}_2] \geq (\varepsilon - \nu)/2$. If E_1 or E_2 occurred, an $\tilde{R}$ will be accepted and returned after at most $(q_G + q_H)$ queries to the Plaintext Checking Oracle.

Plaintext–Extractor. Since we are in an adaptive chosen-ciphertext scenario, we have to simulate the decryption oracle, or to provide a plaintext-extractor. When the adversary asks a query (c_1, c_2, c_3), the simulator looks for the triples (m, R, K) in the table of the query/answer's previously got from the hash functions G and H, using c_1, which one both led to c_2 and c_3. For any correct one, it asks to the Plaintext-Checking Oracle whether c_1 encrypts the given R (therefore globally at most q_H). In

the positive case, it has found a triple (m, R, K) such that, $K = G(R)$ and for some r', $c_1 = \mathcal{E}^{\mathsf{asym}}_{\mathsf{pk}}(R; r')$, $c_2 = \mathcal{E}^{\mathsf{sym}}_{K}(m)$ and $c_3 = H(c_1, R, m)$. The corresponding plaintext is therefore m.

Some decryptions may be incorrect, but only refusing a valid ciphertext: a ciphertext is refused if the query R has not been directly asked to G by the attacker, or (c_1, R, m) not asked to H. This may happen in two situations:

- the attacker has guessed the right value for $H(c_1, R, m)$ without having asked for it, but only with probability $1/2^{k_2}$;
- the c_3 has been given directly by the encryption oracle, which means that it is a part of the challenge ciphertext. Because of c_1, R and m in the triple H-input, the decryption oracle query would either be exactly the challenge ciphertext, which is not allowed to the attacker, or a non-valid ciphertext.

Using this plaintext-extractor, we obtain,

$$\Pr[(\mathsf{E}_1 \vee \mathsf{E}_2) \wedge \text{ no incorrect decryption}] \geq \frac{\varepsilon - \nu}{2} - \frac{q_D}{2^{k_2}},$$

in which cases one solves the Inverting-problem, simply using the Decision-problem oracle to check which element, in the list of queries asked to G and H, is the solution. [QED] ¶

6 Some Examples

We now apply this conversion to many classical encryption schemes which are clearly INV-PCA under some well defined assumptions.

6.1 The RSA Encryption Scheme

6.1.1 Description of the Original Scheme.

In 1978, Rivest–Shamir–Adleman [29] defined the first asymmetric encryption based on the RSA–assumption. It works as follows:

- The user chooses two large primes p and q and publishes the product $n = pq$ together with any exponent e, relatively prime to $\varphi(n)$. He keeps p and q secret, or the invert exponent $d = e^{-1} \bmod \varphi(n)$.

- To encrypt a message $m \in \mathbb{Z}_n^\star$, one just has to compute $c = m^e \bmod n$.
- The recipient can recover the message thanks to d, $m = c^d \bmod n$.

The *one-wayness* of this scheme relies on the RSA assumption. Since this scheme is deterministic, it is still one-way, even against CPA, relative to the RSA assumption.

6.1.2 The Converted Scheme: OCAC–RSA.

Let us consider two hash functions G and H which output k_1-bit numbers and k_2-bit numbers respectively, and any semantically secure symmetric encryption scheme $(\mathcal{K}^{\mathsf{sym}}, \mathcal{E}^{\mathsf{sym}}, \mathcal{D}^{\mathsf{sym}})$.

- Key generation algorithm $\mathcal{K}(1^k)$: it chooses two large primes p and q greater than 2^k, computes the product $n = pq$. A key pair is composed by a random exponent e, relatively prime to $\varphi(n)$ and its inverse $d = e^{-1} \bmod \varphi(n)$.
- Encryption algorithm $\mathcal{E}_{e,n}(m; R)$: with $R \in \mathbb{Z}_n^\star$, it gets $c_1 = R^e \bmod n$, then it computes $K = G(R)$ and $c_2 = \mathcal{E}_K^{\mathsf{sym}}(m)$ as well as $c_3 = H(c_1, R, m)$. The ciphertext consists of the triple $C = (c_1, c_2, c_3)$.
- Decryption algorithm $\mathcal{D}_{d,n}(c_1, c_2, c_3)$, it first extracts $R = c_1^d \bmod n$. Then it recovers $K = G(R)$ and $m = \mathcal{D}_K^{\mathsf{sym}}(c_2)$ which is output if and only if $c_3 = H(c_1, R, m)$. Otherwise, it outputs "Reject".

Theorem 6.1 *The OCAC–RSA encryption scheme is IND-CCA in the random oracle model, under the RSA assumption (and the semantic security of the symmetric encryption scheme under the basic passive attack).*

This becomes the best alternative to OAEP–RSA [3, 30], since $\mathcal{E}^{\mathsf{sym}}$ can simply be the "one-time pad" but also any semantically secure encryption scheme to provide high-speed rates.

6.2 The El Gamal Encryption Scheme

6.2.1 Description of the Original Scheme.

In 1985, El Gamal [12] defined an asymmetric encryption scheme based on the Diffie-Hellman key distribution problem [10]. It works as follows:

- An authority chooses and publishes an Abelian group $\mathcal{G}$ of order q, denoted multiplicatively but it could be an elliptic curve, together with a generator g. Each user chooses a secret key x in $\mathbb{Z}_q^\star$ and publishes $y = g^x$.
- To encrypt a message m, one has to choose a random element k in $\mathbb{Z}_q^\star$ and sends the pair $(r = g^k \bmod p, s = m \times y^k)$ as the ciphertext.
- The recipient can recover the message from a pair (r, s) since $m = s/r^x$, where x is his secret key.

To reach semantic security, this scheme requires m to be encoded by an element in the group $\mathcal{G}$. Whereas the *one-wayness* of this scheme anyway relies on the Computational Diffie-Hellman problem.

Lemma 6.2 *The El Gamal encryption scheme is* INV-PCA *under the Gap-DH Assumption.*

Proof:

This lemma is clear since a Plaintext-Checking Oracle, for a given public key $y = g^x$ and a ciphertext $(r = g^k, s = m \times y^k)$, simply checks whether the triple $(y = g^x, r = g^k, s/m)$ is a DH-triple. It is exactly a Decision Diffie-Hellman Oracle. [QED] ¶

6.2.2 The Converted Scheme: OCAC–El Gamal.

Let us consider two hash functions G and H which output k_1-bit numbers and k_2-bit numbers respectively, and any semantically secure symmetric encryption scheme $(\mathcal{K}^{\mathsf{sym}}, \mathcal{E}^{\mathsf{sym}}, \mathcal{D}^{\mathsf{sym}})$.

- Key generation algorithm $\mathcal{K}(1^k)$: it chooses a large prime q, greater than 2^k, a subgroup $\mathcal{G}$ of order q of an Abelian group $\mathcal{G}'$ and a generator g of $\mathcal{G}$. A key pair is composed by a random element x in $\mathbb{Z}_q^\star$ and $y = g^x$.
- Encryption algorithm $\mathcal{E}_y(m; R, r)$: with $R \in \mathcal{G}'$ and $r \in \mathbb{Z}_q$, it gets $c_1 = g^r$ and $c_1' = R \times y^r$ in $\mathcal{G}'$, then it computes $K = G(R)$ and $c_2 = \mathcal{E}_K^{\mathsf{sym}}(m)$ as well as $c_3 = H(c_1, c_1', R, m)$. The ciphertext consists of the tuple $C = (c_1, c_1', c_2, c_3)$.

- Decryption algorithm $\mathcal{D}_x(c_1, c_1', c_2, c_3)$, it first extracts $R = c_1'/c_1^x$. Then it recovers $K = G(R)$ and $m = \mathcal{D}_K^{\mathsf{sym}}(c_2)$ which is output if and only if $c_3 = H(c_1, c_1', R, m)$. Otherwise, it outputs "Reject".

Theorem 6.3 *The OCAC–El Gamal encryption scheme is IND-CCA in the random oracle model, under the Gap-DH assumption (and the semantic security of the symmetric encryption scheme under the basic passive attack).*

6.3 The Okamoto-Uchiyama Encryption Scheme

6.3.1 Description of the Original Scheme.

Last year, Okamoto–Uchiyama [21] defined an asymmetric encryption based on a trapdoor discrete logarithm. It works as follows:

- Each user chooses two large primes p and q and computes $n = p^2 q$. He also chooses an element $g \in \mathbb{Z}_n^\star$ such that $g_p^{p-1} \bmod p^2$ is of order p and computes $h = g^n \bmod n$. The modulus n, and the elements g and h are made public while p and q are kept secret.

- To encrypt a message m, smaller than p, one has to choose a random element $r \in \mathbb{Z}_n$ and sends $c = g^m h^r \bmod n$ as the ciphertext.

- The recipient can recover the message m from c since $m = L(c_p)/L(g_p) \bmod p$, where $L(x) = (x-1)/p \bmod p$ for any $x = 1 \bmod p$, and $c_p = c^{p-1} \bmod p^2$.

The *semantic security* of this scheme relies on the p-subgroup assumption (a.k.a. p-residuosity or more generally high-residuosity), while the *one-wayness* relies on the factorization of the modulus n. The INV-PCA relies on the gap problem (Gap-High-Residuosity).

However, since the encryption process is public, the bound p is unknown. A public bound has to be defined, for example $n^{1/4}$ which is clearly smaller than p, or 2^k where $2^k < p, q < 2^{k+1}$.

Lemma 6.4 *The Okamoto-Uchiyama encryption scheme is INV-PCA under the Gap-High-Residuosity Assumption.*

Proof:

This lemma is clear since a Plaintext-Checking Oracle is exactly a high-residuosity oracle. [QED] ¶

6.3.2 The Converted Scheme: OCAC–Okamoto-Uchiyama

Let us consider two hash functions G and H which output k_1-bit numbers and k_2-bit numbers respectively, and any semantically secure symmetric encryption scheme $(\mathcal{K}^{\mathsf{sym}}, \mathcal{E}^{\mathsf{sym}}, \mathcal{D}^{\mathsf{sym}})$.

- Key generation algorithm $\mathcal{K}(1^k)$: it chooses two large primes p and q greater than 2^k, as well as g as described above. It then computes $n = p^2 q$ and $h = g^n \bmod n$.
- Encryption algorithm $\mathcal{E}_{n,g,h}(m; R, r)$: with $R < 2^k$ and $r < 2^{3k}$, it gets $c_1 = g^R h^r \bmod n$, then it computes $K = G(R)$ and $c_2 = \mathcal{E}_K^{\mathsf{sym}}(m)$ as well as $c_3 = H(c_1, R, m)$. The ciphertext consists of the triple $C = (c_1, c_2, c_3)$.
- Decryption algorithm $\mathcal{D}_p(c_1, c_2, c_3)$, it first extracts $R = L(c_{1p})/L(g_p)$. Then it recovers $K = G(R)$ and $m = \mathcal{D}_K^{\mathsf{sym}}(c_2)$ which is output if and only if $R < 2^k$ and $c_3 = H(c_1, R, m)$. Otherwise, it outputs "Reject".

Theorem 6.5 *The OCAC–Okamoto-Uchiyama encryption scheme is IND-CCA in the random oracle model, under the Gap-High-Residuosity assumption (and the semantic security of the symmetric encryption scheme under the basic passive attack).*

7 Conclusion

This paper presented OCAC, an optimal conversion which applies to any weakly secure cryptosystem: the overload is as negligible as OAEP, and advantages of OCAC beyond OAEP are numerous. Therefore, OCAC provides an optimal solution to realize a provably secure (in the strongest security sense) asymmetric or hybrid encryption schemes based on any practical asymmetric encryption primitive such as RSA, El Gamal, or Elliptic-Curve El Gamal. In addition, this paper introduced a novel class of computational problems, the *gap problems*, which is considered to be dual to the class of the *decision problems*.

参考文献

[1] M. Abdalla, M. Bellare, and P. Rogaway. DHAES: An Encryption Scheme Based on the Diffie-Hellman Problem. Submission to IEEE P1363a. September 1998. Available from `http://grouper.ieee.org/groups/1363/`.

[2] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among Notions of Security for Public-Key Encryption Schemes. In *Crypto '98*, LNCS 1462, pages 26–45. Springer-Verlag, Berlin, 1998.

[3] M. Bellare and P. Rogaway. Optimal Asymmetric Encryption – How to Encrypt with RSA. In *Eurocrypt '94*, LNCS 950, pages 92–111. Springer-Verlag, Berlin, 1995.

[4] M. Bellare and A. Sahai. Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization. In *Crypto '99*, LNCS 1666, pages 519–536. Springer-Verlag, Berlin, 1999.

[5] D. Bleichenbacher. A Chosen Ciphertext Attack against Protocols based on the RSA Encryption Standard PKCS #1. In *Crypto '98*, LNCS 1462, pages 1–12. Springer-Verlag, Berlin, 1998.

[6] L. Carter and M. Wegman. Universal Hash Functions. *Journal of Computer and System Sciences*, 18:143–154, 1979.

[7] D. Coppersmith, S. Halevi, and C. S. Jutla. ISO 9796 and the New Forgery Strategy. Working Draft presented at the Rump Session of Crypto '99, 1999.

[8] J. S. Coron, D. Naccache, and J. P. Stern. On the Security of RSA Padding. In *Crypto '99*, LNCS 1666, pages 1–18. Springer-Verlag, Berlin, 1999.

[9] R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In *Crypto '98*, LNCS 1462, pages 13–25. Springer-Verlag, Berlin, 1998.

[10] W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT–22(6):644–654, November 1976.

[11] D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. In *Proc. of the 23rd STOC*. ACM Press, New York, 1991.

[12] T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, IT–31(4):469–472, July 1985.

[13] E. Fujisaki and T. Okamoto. How to Enhance the Security of Public-Key Encryption at Minimum Cost. In *PKC '99*, LNCS 1560, pages 53–68. Springer-Verlag, Berlin, 1999.

[14] E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In *Crypto '99*, LNCS 1666, pages 537–554. Springer-Verlag, Berlin, 1999.

[15] S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.

[16] R. J. McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory. *DSN progress report*, 42-44:114–116, 1978. Jet Propulsion Laboratories, CALTECH.

[17] D. Naccache and J. Stern. A New Public-Key Cryptosystem. In *Eurocrypt '97*, LNCS 1233, pages 27–36. Springer-Verlag, Berlin, 1997.

[18] D. Naccache and J. Stern. A New Cryptosystem based on Higher Residues. In *Proc. of the 5th CCS*, pages 59–66. ACM Press, New York, 1998.

[19] M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *Proc. of the 22nd STOC*, pages 427–437. ACM Press, New York, 1990.

[20] T. Okamoto, E. Fujisaki, and H. Morita. PSEC: Provably Secure Elliptic Curve Encryption Scheme. Submission to IEEE P1363a. March 1999.
Available from `http://grouper.ieee.org/groups/1363/`.

[21] T. Okamoto and S. Uchiyama. A New Public Key Cryptosystem as Secure as Factoring. In *Eurocrypt '98*, LNCS 1403, pages 308–318. Springer-Verlag, Berlin, 1998.

[22] T. Okamoto, S. Uchiyama, and E. Fujisaki. EPOC: Efficient Probabilistic Public-Key Encryption. Submission to IEEE P1363a. November 1998.
Available from `http://grouper.ieee.org/groups/1363/`.

[23] P. Paillier. Public-Key Cryptosystems Based on Discrete Logarithms Residues. In *Eurocrypt '99*, LNCS 1592, pages 223–238. Springer-Verlag, Berlin, 1999.

[24] P. Paillier and D. Pointcheval. Efficient Public-Key Cryptosystems Provably Secure against Active Adversaries. In *Asiacrypt '99*, LNCS 1716, pages 165–179. Springer-Verlag, Berlin, 1999.

[25] D. Pointcheval. HD–RSA: Hybrid Dependent RSA – a New Public-Key Encryption Scheme. Submission to IEEE P1363a. October 1999.
Available from `http://grouper.ieee.org/groups/1363/`.

[26] D. Pointcheval. New Public Key Cryptosystems based on the Dependent-RSA Problems. In *Eurocrypt '99*, LNCS 1592, pages 239–254. Springer-Verlag, Berlin, 1999.

[27] D. Pointcheval. Chosen-Ciphertext Security for any One-Way Cryptosystem. In *PKC '2000*, LNCS 1751, pages 129–146. Springer-Verlag, Berlin, 2000.

[28] C. Rackoff and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto '91*, LNCS 576, pages 433–444. Springer-Verlag, Berlin, 1992.

[29] R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[30] RSA Data Security, Inc. Public Key Cryptography Standards – PKCS.
Available from `http://www.rsa.com/rsalabs/pubs/PKCS/`.

[31] C. E. Shannon. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, 28(4):656–715, 1949.

[32] V. Shoup and R. Gennaro. Securing Threshold Cryptosystems against Chosen Ciphertext Attack. In *Eurocrypt '98*, LNCS 1403, pages 1–16. Springer-Verlag, Berlin, 1998.

[33] Y. Tsiounis and M. Yung. On the Security of El Gamal based Encryption. In *PKC '98*, LNCS. Springer-Verlag, Berlin, 1998.