2000

The zheng-seberry public key cryptosystem and signcryption

David J. Soldera
University of Wollongong

The Zheng-Seberry Public Key Cryptosystem and Signcryption

A thesis submitted in fulfilment of the requirements for the award of the degree

Honours Masters in Computer Science

From

University of Wollongong

By

David J Soldera, BE (Elec, Hons 1) University of Canterbury, Christchurch, New Zealand

School of Information Technology and Computer Science

2000

Abstract

In 1993 Zheng-Seberry presented a public key cryptosystem that was considered efficient and secure in the sense of indistinguishability of encryptions (IND) against an adaptively chosen ciphertext adversary (CCA2). This thesis shows the Zheng-Seberry scheme is not secure as a CCA2 adversary can break the scheme in the sense of IND. In 1998 Cramer-Shoup presented a scheme that was secure against an IND-CCA2 adversary and whose proof relied only on standard assumptions. This thesis modifies this proof and applies it to a modified version of the El-Gamal scheme. This resulted in a provably secure scheme relying on the Random Oracle (RO) model, which is more efficient than the original Cramer-Shoup scheme. Although the RO model assumption is needed for security of this new El-Gamal variant, it only relies on it in a minimal way.

In 1997 Zheng introduced a new notion called signcryption, a combination of signature and encryption. Zheng gave some details and properties of this new notion but did not include any formal definitions about the notions of security for a signcryption scheme. This thesis presents some formal notions of security for signcryption schemes, based on accepted notions of security for encryption and signature schemes.

Three new signcryption schemes are presented that are based on provably secure encryption schemes. The security of these new schemes is presented in terms of the new notions of security for signcryption schemes. Strong arguments are made for the security of these new schemes, formal proofs are not given as that would be a bold claim for schemes based on such new theory.

Some discussion is given about the possibilities of combining weak encryption with strong signatures to achieve stronger encryption. Intuitively the concept seems correct but proofs remain elusive. The idea of combining strong encryption and weak signatures is also discussed but appears less promising.

Also, some minor properties of elliptic curves are given. The theorems most probably are not original, but have been rediscovered by the author. Where it was discovered that a theorem was previously known, it is referenced. The theorems presented, whilst interesting, are not significant contributions to the field.

Declaration

I, David John Soldera, declare that this thesis, submitted in fulfilment of the requirements for the award of Honours Masters of Science (Computer Science), in the School of Information Technology and Computer Science, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualifications at any other academic institution.

David J. Soldera
December 2000

Acknowledgments

I wish to thank the following people:

- Dr David Pointcheval of ENS in France, although I only ever communicated with him via email, he made me appreciate the intricacies of proving security.
- Professor Jennifer Seberry and Associate Professor Josef Pieprzyk (my supervisors) for their help and guidance. Without them I would not have been led down the interesting paths that resulted in this thesis.
- And last but not least my family, their encouragement and support have helped me to an extent they will never appreciate.

Contents

CHAPTER 1 INTRODUCTION

The right to privacy is one of but a handful of rights that belong to every free individual. This right however, is becoming more and more difficult to maintain as information becomes the most prized commodity of the new millennium. This Information Age or Digital Age has brought about an unimaginable amount of information exchange, from one side of the world to the other, in the blink of an eye. So to where do we look to protect this vast and valuable commodity? We look to computer security and the theory of cryptography.

Public key cryptography is an important part in the theory of cryptography, it allows two people (let's assign them randomly chosen names, Alice and Bob), who have never meet each other, to communicate confidentially. Public key cryptography was essentially discovered by Whitfield Diffie and Martin Hellman in 1976 [18], when they discovered a way for Alice and Bob to create a shared secret key. From here actual public key cryptosystems were devised, such as one by Rivest, Shamir and Adleman [56] (RSA) and El-Gamal [21]. These two cryptosystems to this day underlie the majority of public key cryptosystems.

Security is the most important property of any public key cryptosystem, it is also the most difficult to show. The security of a scheme is always based on some assumptions. An adversary with access to unlimited resources, that wants to break a public key cryptosystem, always can. Hence assumptions are made that define the resources of an adversary and the way in which they are trying to break the cryptosystem.

Proving the security of a public key cryptosystem is a concept that has been around for almost as long as public key cryptography. In a scheme by Rabin [54] (1979) he presented a scheme that was as intractable as factoring, under certain (strong) assumptions. However, provable security is a very difficult goal and often comes at the cost of efficiency. Nevertheless, it is a goal that many public key cryptosystems are now achieving, and should be the goal of all such future cryptosystems.

Zheng-Seberry [70] presented a scheme in 1993 and gave a proof for its security. Their scheme represented one of the first schemes considered secure against the most

powerful attacker cryptographic theory defines, and yet at the same time was very practical. 0 of this thesis shows for the first time that their proof is not valid. This result should be credited equally to the author of this thesis and Assoc. Prof. Josef Pieprzyk as it was arrived at by this author during discussions with Assoc. Prof. Pieprzyk. Having broken the scheme, the focus is on repairing it.

The task of repairing Zheng-Seberry led the author to create an El-Gamal variant that's security involves adapting a proof from a scheme by Cramer-Shoup [15]. Although the variant and its proof are 100% the work of the author, Dr David Pointcheval helped verify the proof. The Cramer-Shoup scheme has the distinction of being the first provably secure public key cryptosystem that is practical, and whose proof relies only on standard assumptions. It makes sense then to try to adapt this proof to prove the security of other cryptosystems. The Cramer-Shoup proof cannot be directly applied to a new El-Gamal variant but aspects of its construction can be borrowed to develop a new proof.

The elliptic curve implementation of Zheng-Seberry is an important variant of the original. It turns out we can use the points on elliptic curves in much the same way we use counting numbers, but most importantly a discrete logarithm problem for elliptic curves can be defined. Interestingly, the elliptic curve discrete logarithm problem is currently a more difficult computational problem to solve than the standard discrete logarithm problem. This means cryptosystems based on the elliptic curve discrete logarithm problem can be implemented more efficiently. Hence the study of elliptic curves is a very important area.

Public key cryptography can achieve more than just confidentiality, it can also achieve authentication. Authentication is a process where if Bob receives a message from Alice, Bob can verify that Alice did indeed write the message. Alice provides this authentication by digitally signing the message she sends Bob. This digital signature is analogous in many ways to a hand written signature. As with encryption schemes, it is important for digital signature schemes to be secure and efficient.

Signcryption is a notion that was introduced by Zheng [68] and, as the name suggests, combines digital signatures and encryption. The combination is not trivial like concatenation, but must achieve the goal of having the 'cost' of the signcryption scheme less than the cost of the signature and encryption schemes taken independently. The goal of signcryption is to combine authentication and confidentiality in such a way as to get an advantage over doing them independently.

Although Zheng introduced the notion of signcryption, he did not include formal definitions and theories describing the security of signcryption schemes. Zheng did present a description of signcryption and many of its desirable properties. Also presented was a signcryption scheme and a sound argument for its security, yet without formal definitions of what it means for a signcryption scheme to be secure, we cannot be as confident about its security as we would like.

Definitions for the security of a signcryption scheme are solely the work of the author and are presented for the first time in Chapter 5. This task is made simpler by signcryption being a natural combination of digital signatures and encryption, both of which have well defined notions of security. This task is not simple though, as

intertwining two theories is fraught with pitfalls. This is the first such attempt at formally defining security for signcryption and undoubtedly as this new notion matures so must these definitions develop too.

With definitions for the security of signcryption, the development of new signcryption schemes can proceed with some confidence. In Chapter 6 this thesis presents three new signcryption schemes. The importance of provable security has been alluded too for confidentiality, and the same is true for signcryption schemes. Hence these new signcryption schemes all have underlying encryption schemes that are provably secure. Of course this does not guarantee the security of the signcryption scheme, but it is a good place to start. The arguments given for the security of these three new schemes are not formal proofs, since that is not a claim that could be made with confidence for such new theory.

The rest of this thesis is laid out in the following order. Chapter 2 describes the mathematical background required to understand public key cryptography and some formal definitions about the security of encryption schemes. Also, the theory of elliptic curves is reviewed and some minor theorems presented. Chapter 3 presents a review of previous schemes that are provably secure and presents the development of proving security for cryptosystems. Some relevant signature schemes are also reviewed. Chapter 4 describes in more detail the original Zheng-Seberry scheme and how it can be broken by an adversary. A modified version of the Cramer-Shoup proof is then used to show the security of an El-Gamal variant. Chapter 5 adds some substance to the theory of signcryption by formalising notions of security for it. Chapter 6 presents three new signcryption schemes and Chapter 7 poses some new, unanswered questions about the interplay between confidentiality and authentication. Chapter 8 offers some conclusions that can be drawn from this thesis and Chapter 9 gives the bibliography.

CHAPTER 2 MATHEMATICAL BACKGROUND

2.1 Groups, Rings and Fields

Algebraic structures are powerful mathematical constructs that immediately give information about their elements. They are important to appreciate in cryptography, as when working with an algebraic structure, for example a field, it is immediately known, for example, that every element has an inverse. Conversely, the inability to place elements of a cryptosystem in a well-defined algebraic structure can be important information in regards to the security of the cryptosystem.

Only the basic algebraic structures have been defined here, as these tend to underlie more complex structures.

2.1.1 Groups

A group G is a set with a binary operation $\circ$ defined on it, such that the following axioms are true.

1. The binary operation is associative.
$$(x \circ y) \circ z = x \circ (y \circ z) \qquad \forall x, y, z \in G$$
2. There is a unique identity element i.
$$i \circ x = x \circ i = x \qquad \forall x \in G$$
3. For each element x in G there is a unique inverse $\overline{x}$ in G.
$$x \circ \overline{x} = \overline{x} \circ x = i \qquad \forall x \in G$$

The number of elements in a group is called its *order*.

2.1.2 Rings

A ring R is a set with two binary operations + and $\circ$ defined on it, such that the following axioms are true for all elements x, y, z in R.

1. The + operation is commutative.
$$x + y = y + x$$
2. The + operation is associative.
$$(x + y) + z = x + (y + z)$$
3. There is a unique identity element i_+ for the + operation.

$$x + i_+ = i_+ + x = x$$

4. Every element has a unique inverse (-x) under the + operation.
$$x + (-x) = (-x) + x = i_+$$
5. The left distributive property.
$$x \circ (y + z) = x \circ y + x \circ z$$
6. The right distributive property.
$$(x + y) \circ z = x \circ z + y \circ z$$
7. The $\circ$ operation is associative.
$$(x \circ y) \circ z = x \circ (y \circ z)$$

2.1.3 Fields

A ring R can have additional algebraic properties defined on it.

1. A ring R is commutative (or abelian) if $x \circ y = y \circ x \quad \forall x, y \in R$
2. A ring R has a unique identity element $i_\circ$ for the $\circ$ operation if $x \circ i_\circ = i_\circ \circ x = x \quad \forall x \in R$
3. Every element of R has a unique inverse $\bar{x}$ under the $\circ$ operation.
$$x \circ \bar{x} = \bar{x} \circ x = i_\circ \qquad \forall x \in R$$

A field is a ring with all of the above three extra properties.

The size of a field can be infinite such as $\mathbb{Q}$, $\mathbb{R}$ or $\mathbb{C}$, they can also be finite such as $\mathbb{Z}_p$, the set of integers from 0 to $p - 1$ for some prime p.

2.2 Number Theory

Number theory is cryptographers 'bread and butter'; it is the foundations upon which everything is built. This section highlights the main theorems and definitions that will be relevant to this thesis. Proofs have been omitted, but [1, 12, 14, 23, 30, 37, 44, 50, 63] contain all relevant proofs and cover all the necessary background to understand the mathematics of public key cryptography.

This section will work with a finite set of integers, $\mathbb{Z}_n$, or the integers from 0 to n - 1. This set is a commutative ring with identity. This means that if the binary operations are addition and multiplication it can't be guaranteed that every element will have a multiplicative inverse.

2.2.1 Modulo Arithmetic

Cryptographers work with groups and fields and the simplest groups and fields are those associated with integers with operations based on modulo integer arithmetic. Fortunately computers are most suited to finite precision arithmetic, leading to efficient implementations. So a finite set of integers are used for public key cryptography, and any integers that are larger than the maximum element are replaced by an element in the set that is congruent to it. For example, if working with integers modulo 5, then 7 would be replaced with 2, or

$$7 \equiv 2 \bmod 5$$

It is said that 7 reduced modulo 5, is equivalent, or congruent to 2.

2.2.2 Euler's Theorem

Euler's theorem is a generalisation of Fermat's Little Theorem, so understanding Fermat's Little Theorem is essential. First let $p \mid a$ be read as "p divides a", meaning that p divides into a an integer amount of times. Similarly, let $p \nmid a$ be read as "p does not divide a".

Theorem 1 *(Fermat's Little Theorem) If p is a prime and $p \nmid a$, then $a^{p-1} \equiv 1 \bmod p$.*

This theorem is true when working modulo a prime, but Euler extended it to any number n. To understand Euler's theorem, Euler's phi-function (sometimes called the indicator or totient function) must first be introduced.

Definition 1 [12, pg. 136] For $n \geq 1$, let $\phi(n)$ denote the number of positive integers not exceeding n that are relatively prime to n.

Euler's phi-function can be best illustrated with an example. $\phi(30) = 8$, or there are 8 integers that are less than 30 and relatively prime (or co-prime) to 30, they are 1, 7, 11, 13, 17, 19, 23, 29. The phi-function is extremely important (eg in RSA) and is difficult to calculate for large n. However, when $n = p$, where p is prime, then $\phi(p) = p - 1$. Note also that $\phi(n)$ is a multiplicative function, for example $\phi(30) = \phi(2)\,\phi(15) = \phi(2)\,\phi(3)\,\phi(5)$. This means that $\phi(n)$ can be easily calculated if n can be factored into its prime factors.

Now Euler's theorem can be given.

Theorem 2 *(Euler) If n is a positive integer and gcd(a, n) = 1 then $a^{\phi(n)} \equiv 1 \bmod n$.*

This theorem is the foundation of much of public key cryptography.

2.2.3 Orders and Generators

When working in $\mathbb{Z}_n$ the order of an element is defined to be:

Definition 2 [12, pg. 156] Let $n < 1$ and $\gcd(a, n) = 1$. The *order of a modulo n* is the smallest positive integer k such that $a^k \equiv 1 \bmod n$.

If an element belongs to a group (or ring or field), then the order of the element must divide the order of the group.

An element's order allows us to define whether or not it is a primitive root.

Definition 3 [12, pg. 159] If $\gcd(a, n) = 1$ and a is of order $\phi(n)$ modulo n, then a is a *primitive root of n.*

The usefulness of primitive roots can be seen from the next theorem.

Theorem 3 *Let gcd(a, n) = 1 and let $a_1, a_2, \ldots, a_{\phi(n)}$ be the positive integers less than n and relatively prime to n. If a is a primitive root of n, then*

$$a, a^2, \ldots, a^{\phi(n)}$$

are congruent modulo n to $a_1, a_2, \ldots, a_{\phi(n)}$, *in some order.*

Specifically, when $n = p$, then the powers of a primitive root, g, will generate all the positive integers less than p, hence g is called a generator.

A useful result of the above theorem is the following corollary.

Corollary *If n has a primitive root, then it has exactly* $\phi(\phi(n))$ *of them.*

So when $n = p$, and $p - 1$ has a large prime factor, the number of generators can be easily found.

2.3 Discrete Logarithm Problem

The Discrete Logarithm Problem (DLP) is the basis for many cryptosystems, of these El-Gamal [21] is perhaps the best known, and it has numerous variants. The definition of the DLP is given in Definition 4.

Definition 4 (*Discrete Logarithm Problem*) Given a prime number p and two other numbers g and y (between 1 and $p - 1$), find a number x such that $y = g^x \bmod p$.

The definition has been given for a cyclic group of integers, but it can be generalised to any group (see [46, pg. 104]).

The algorithms that have been devised to solve the DLP can depend on the underlying group. There are algorithms that are independent of the group, eg exhaustive search, the baby-step giant-step algorithm [46, §3.6.2] and Pollard's rho algorithm [46, §3.6.3]. Another algorithm that works in arbitrary groups but is particularly efficient if the group has only small prime factors, is the Pohlig-Hellman algorithm [46, §3.6.4]. The fastest known algorithms for solving the DLP only work on certain groups and are variants of the index calculus algorithm [46, §3.6.5]. The running time of the index calculus method is sub-exponential, given as $L_q[\frac{1}{2}, c]$ where $L_q[\alpha, c] = O(\exp((c + o(1))(\ln q)^{\alpha}(\ln \ln q)^{1-\alpha}))$ with c a positive constant and α satisfying $0 < \alpha < 1$. Note, when $\alpha = 0$ the running time is polynomial and when $\alpha = 1$ the running time is fully exponential. The fastest known algorithm for the DLP is a variant of the index calculus method running at $L_q[1/3, c]$.

2.4 RSA and El-Gamal Cryptosystems

The RSA and El-Gamal cryptosystems are the basis for numerous cryptosystems. Since most cryptosystems discussed in this thesis are based on these systems, how they work will be outlined here.

RSA
Preliminaries
Generate two strong primes p and q. Calculate their product $N = pq$.
Key Generation
Generate a random number d as the public key and solve for e in $ed = 1 \bmod \phi(N)$ for the private key.
Encryption
Encrypt a message m. 1) $c = m^d \bmod N$ Ciphertext is c.
Decryption
1) $m = c^e \bmod N$

The security of RSA lies in the necessity to factor the product of two large primes. However, the security has not been shown to be equivalent to this.

El-Gamal
Preliminaries
Work over GF(p) with p a large prime and a generator g.
Key Generation
Choose a random x as a private key and use $y = g^x \bmod p$ as the public key.
Encryption
Encrypt message m 1) Choose random r 2) $a = g^r$ 3) $b = y^r m$ Ciphertext (a, b)
Decryption
1) $m = b/a^x$

The security of El-Gamal lies in the difficulty of solving an instance of the discrete logarithm problem, actually the (semantic) security of this most basic version of El-Gamal has been shown equivalent to the Decision Diffie-Hellman Problem [64].

Both RSA and El-Gamal are secure against a chosen message attacker but in their most basic forms are insecure against an adaptive adversary. The definitions of these adversaries are in section 2.7.

2.5 Elliptic Curves

Elliptic curves get their name as they were originally used to calculate the circumference of an ellipse. Elliptic curves are useful in cryptography because the points on an elliptic curve form an abelian group and this allows the creation of a cryptographic scheme based on the discrete logarithm problem.

Some good sources of information on elliptic curves are [20, 28, 34, 38, 61, 67] and there implementation [6, 33, 39, 42].

2.5.1 Definition

Plane curves are equations that satisfy $F(x, y) = 0$. The simplest plane curves are lines (of degree 1 in x and y) and conic sections (of degree 2 in x and y). The next simplest are cubic curves, which include elliptic curves.

The elliptic curves that will be considered are those in the Weierstrass equation form. The most general form, in affine coordinates, is given below.

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \qquad \text{Equation 1}$$

The Weierstrass equation can be expressed in projective coordinates involving X, Y and Z. There is exactly one point on the elliptic curve where $Z = 0$, namely (0, 1, 0) and so in affine coordinates (where $x = X/Z$ and $y = Y/Z$) this point is given a special value, $\mathcal{O}$, which is referred to as the point at infinity. The point at infinity is the identity element for the group.

Elliptic curves used in cryptography are defined over a finite field K and are required to be non-singular (their determinant $\neq 0$).

The equation for the elliptic curve can be simplified further if the characteristic of the underlying field K is restricted. There are two restrictions that are used in cryptography, when the underlying field has characteristic greater than 3 and when it has characteristic equal to 2.

For an elliptic curve E, defined over a field K with characteristic greater than 3, then by a simple change of variables the equation for E becomes:

$$E : y^2 = x^3 + ax + b \qquad a,b \in K \qquad \text{Equation 2}$$

Most often in cryptography this type of curve is used when the underlying field is GF(p) (or some extension field of this), with p prime. For this type of curve to be non-singular it is required that $4a^3 + 27b^2 \neq 0 \bmod p$.

If the characteristic of K is equal to 2, then the equation for E becomes:

$$E : y^2 + xy = x^3 + ax^2 + b \qquad a,b \in K \qquad \text{Equation 3}$$

For cryptographic purposes this curve is most often used when the underlying field is GF(2^m). For this type of curve to be non-singular it is required that $b \neq 0$. Curves over GF(2^m) are used in most implementations of elliptic curve cryptography as they lend themselves to the most efficient implementation in computers.

2.5.2 Addition

The group addition for elliptic curves is defined by the chord and tangent rule. If E is defined over the real numbers then the addition rule can be seen graphically, in Figure 1.

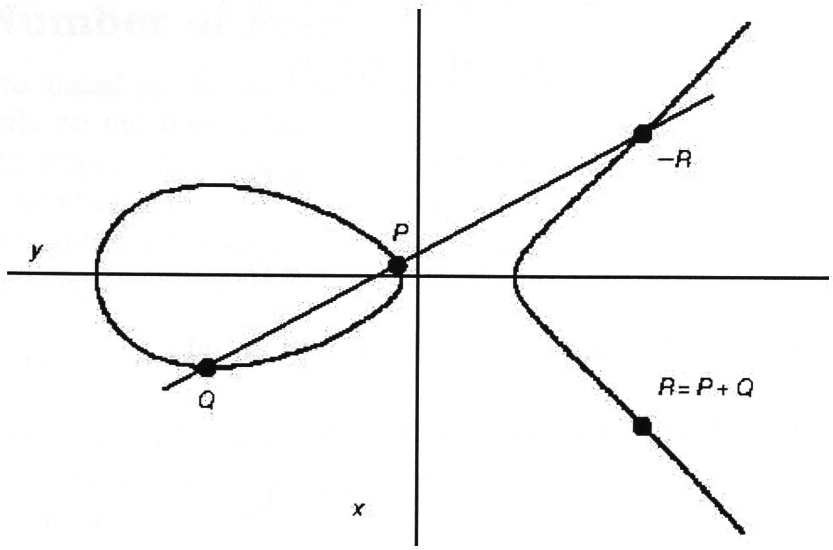


Figure 1 – A graphical representation of adding two points P and Q on an elliptic curve, to get a resultant point R.

To add two points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ on an elliptic curve, a line is drawn through them, and it can be shown that this line is guaranteed to intersect the curve at exactly one other point (or $\mathcal{O}$). The negative of the intersection point is then taken, and this is the result $R = (x_3, y_3)$. If $P = Q$ then the line taken is the tangent of the curve at that point. If $P = -Q$ then the result is $\mathcal{O}$.

Mathematically, addition is trying to find the third point of intersection between Equation 4 (a straight line) and Equation 5 (an elliptic curve).

$$y = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x - x_1) + y_1 = \lambda(x - x_1) + y_1 \qquad \text{Equation 4}$$

$$y^2 = x^3 + ax + b \qquad \text{Equation 5}$$

By eliminating y we arrive at a cubic, comparing the x^2 coefficient of this with that of

$$(x - x_1)(x - x_2)(x - x_3) = 0$$

gives Equation 6.

$$x_1 + x_2 + x_3 = \lambda^2$$

$$x_3 = \lambda^2 - x_1 - x_2 \qquad \text{Equation 6}$$

Now that x_3 can be found, Equation 4 can be used to find y_3, giving Equation 7 (remember the negative point is needed so the y coordinate is negated).

$$y_3 = \lambda(x_1 - x_3) - y_1 \qquad \text{Equation 7}$$

The addition formulas are slightly different when working over a field with characteristic 2, so a summary of both is given below.

When working over GF(p) with $P = (x_1, y_1)$ and Q = (x_2, y_2), and $P \neq -Q$, then $P + Q = (x_3, y_3)$ where

$$x_3 = \lambda^2 - x_1 - x_2$$
$$y_3 = \lambda(x_1 - x_3) - y_1$$

and

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & \text{if } P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1}, & \text{if } P = Q \end{cases}$$

If $P = -Q$ then $P + Q = \mathcal{O}$. Also if $P = (x, y)$ then $-P = (x, -y)$.

If working over GF(2^m) then the equations become

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & \text{if } P \neq Q \\ x_1 + \dfrac{y_1}{x_1}, & \text{if } P = Q \end{cases}$$

$$x_3 = \begin{cases} a + \lambda^2 + \lambda + x_1 + x_2, & \text{if } P \neq Q \\ a + \lambda^2 + \lambda, & \text{if } P = Q \end{cases}$$

$$y_3 = \lambda(x_1 + x_3) + x_3 + y_1$$

If $P = (x, y)$ then $-P = (x, x + y)$.

These are the methods used in the P1363 standard (draft 7-16-99) [29]. Subtraction for elliptic curves is achieved by adding the negative of a point.

2.5.3 Scalar Multiplication

The other group operation for elliptic curves is scalar multiplication. This is achieved by repeated addition, for example, consider multiplying the point P by 5.

$$5P = P + ((P + P) + (P + P))$$

In this case calculating $5P$ requires 3 additions. However there are faster ways to achieve scalar multiplication, the method used in the P1363 standard [29] is as follows.

P1363 A.10.3
Input: an integer n and an elliptic curve point P.

Output: the elliptic curve point nP.

1. If $n = 0$ then output $\mathcal{O}$ and stop.
2. If $n < 0$ the set $Q \leftarrow (-P)$ and $k \leftarrow (-n)$, else set $Q \leftarrow P$ and $k \leftarrow n$.
3. Let $h_l h_{l-1} ... h_1 h_0$ be the binary representation of $3k$, where the most significant bit h_l is 1.
4. Let $k_l k_{l-1} ... k_1 k_0$ be the binary representation of k.
5. Set $S \leftarrow Q$.
6. For i from $l - 1$ downto 1 do
   Set $S \leftarrow 2S$.
   If $h_i = 1$ and $k_i = 0$ then compute $S \leftarrow S + Q$ via A.10.1 or A.10.2.
   If $h_i = 0$ and $k_i = 1$ then compute $S \leftarrow S - Q$ via A.10.1 or A.10.2.
7. Output S.

This method is similar to better known square-and-multiply method normally associated with exponentiation.

2.5.4 Number of Points

Cryptosystems based on the discrete logarithm over finite fields have their security depend heavily on the size of the group - the same is true for elliptic curves. The problem with elliptic curves is given a particular curve it is not always trivial to calculate the number of points on the curve. What is known though is the bound on the number of points, given an elliptic curve E defined over a finite field F_q with $q = p^m$ (p is prime) then the number of points in $E(F_q)$, denoted by $\#E(F_q)$ is given by.

Theorem 4 *(Hasse) Let* $\#E(F_q) = q + 1 - t$. *Then* $|t| \leq 2\sqrt{q}$.

There is a polynomial time algorithm (due to Schoof [59]) that computes the number of points on an arbitrary elliptic curve, however it still is not very fast. This algorithm has been substantially sped up by Elkies [22] and Atkins [2], see also [31, 40, 43]. There are other ways to calculate the number of points, the complex multiplication method [32, 36, 48] lets you choose some parameters related to the number of points on the curve and then finds a curve with that number of points.

2.5.5 Elliptic Curve Discrete Logarithm

The Elliptic Curve Discrete Logarithm Problem (ECDLP) is defined in an analogous way to the DLP. The analogies between the two can be seen in Table 1.

Table 1 – Analogies between the DLP and the ECDLP.

	DLP	**ECDLP**
Setting	$GF(q)$	Curve E over $GF(q)$
Basic operation	Multiplication in $GF(q)$	Addition of points
Main operation	Exponentiation	Scalar multiplication
Base element	Generator g	base point G
Base element order	Prime r	Prime r
Private key	s (integer modulo r)	s (integer modulo r)
Public key	W (element of $GF(q)$)	W (point on E)

Hence the definition of the ECDL problem is as follows.

Definition 5 (*ECDL*) Given an ellptic curve E defined over a finite field F_q, and two points $P, Q \in E(F_q)$, find an integer t such that $P = tQ$, given that such an integer exists.

What makes the ECDL problem interesting, is that unlike the DL problem for which there exists a sub-exponential algorithm (see section 2.3) for solving, no such algorithm works for the ECDL problem. The fastest known algorithms are still fully exponential.

The reason for this is the sub-exponential algorithms based on the index calculus method require the use of a factor base, which when working in $\mathbb{Z}_n$ mean small prime numbers. However, to apply the same algorithm to elliptic curves, one would need 'small prime points', a concept that is just not defined for elliptic curves. There have been attempts to define these for elliptic curves but as yet they have not been very successful.

2.5.6 Supersingular and Anomalous Curves

Although at present the ECDL is more difficult to solve than the DL, there are some elliptic curves for which the ECDL reduces to the difficulty of the DL problem. These curves are either supersingular curves or anomalous curves.

Remember the bound on the number of points on an EC is given by $\#E(F_q) = p + 1 - t$, well supersingular curves are curves were p divides t. Since $|t| \leq 2\sqrt{q}$ this means that an elliptic curve defined over F_q is supersingular if $t^2 = 0$, q, $2q$, $3q$ or $4q$. This requirement on an elliptic curve is known as the MOV condition as Menezes, Okamoto and Vanstone [41] discovered this reduction for supersingular curves.

Anomalous curves are defined as curves where $\#E(F_q) = p$. Smart [62]-Satoh-Araki [57] showed that for such curves the ECDL could be reduced to the DL problem.

2.5.7 Isomorphic and Complimentary Curves

There is a simple theorem that defines whether or not two curves are isomorphic to each other.

Theorem 5 *Two elliptic curves, E_1: $y^2 = x^3 + ax + b$ and E_2: $y^2 = x^3 + \bar{a}x + \bar{b}$, over the field K are isomorphic over K if and only if there exists a $u \in K^*$ such that $u^4\bar{a} = a$ and $u^6\bar{b} = b$. The isomorphism is given by*

$$\phi : E_1 \to E_2, \quad \phi : (x, y) \mapsto (u^{-2}x, u^{-3}y)$$

or equivalently

$$\phi : E_2 \to E_1, \quad \phi : (x, y) \mapsto (u^2 x, u^3 y)$$ ∎

The advantage of knowing about isomorphic curves is that if you know the number of points on one curve then all curves isomorphic to that curve have the same number of points. The isomorphism equations can be extended to the general curve given in Equation 1.

It is easy to find an isomorphic curve if you know a and b in terms of some generator g, say $a = g^v$ and $b = g^w$, then an isomorphic curve is $\bar{a} = g^{v+4}$ and $\bar{b} = g^{w+6}$.

Another relation between curves is complimentary curves [17] (also known as the *twist* of a curve). EC points exists where $x^3 + ax + b$ is a quadratic residue, however complimentary curves ($\bar{E}$) have points where this value is a quadratic non-residue. The element y is now expressed as $y = u\sqrt{v}$ where v is a fixed quadratic non-residue. These points form an abelian group and have virtually identical rules for addition, differing only by the $\sqrt{v}$ which is associated with the y coordinate.

As given in Theorem 4 the number of points on an EC are $\#E = p + 1 - t$. The property of complimentary curves that makes them useful is the number of points on the complimentary curve of E is given by $\#\bar{E} = p + 1 + t$.

2.5.8 Some Minor Theorems

Here are some interesting theorems about some global properties of elliptic curves. The theorems presented are most likely not new work, but they were rediscovered by the author, and are presented with the hope that at least the proofs represent original work.

Theorem 6 shows how to transform an elliptic curve with $\#E_1 = p + 1 - t$ to another elliptic curve with $\#E_2 = p + 1 + t$. It is practically useful as if the number of points on one curve is known, it essentially doubles the number of elliptic curves where the number of points can be calculated for virtually no cost.

This is not a new result, it can be found in [7], however the author believes the proof provided to be original.

Theorem 6 *For an EC* E_1: $y^2 = x^3 + ax + b$ *modulo a prime* p *(with generator* g*), where* $a = g^v$ *and* $b = g^w$ *and* $\#E_1 = p + 1 - t$*, then the curve* $E_2 : y^2 = x^3 + \hat{a}x + \hat{b}$ *where* $\hat{a} = g^{v+2}$ *and* $\hat{b} = g^{w+3}$ *has* $\#E_2 = p + 1 + t$.

Proof.
The isomorphism equations from Theorem 5 are used to show how every y^2 value of E that is a quadratic residue becomes a quadratic non-residue.

Remember the isomorphic transform equations:

$$(x, y) \mapsto (u^2 x, u^3 y)$$

Let $u = g^{c/2}$, now when c is even it is the case of an isomorphic transform, here is considered the case when c is odd.

The equations now become

$$(x, y) \mapsto \left(g^c x, g^{\frac{3c}{2}} y \right) = \left(g^c x, y\left(\sqrt{g}\right)^{3c} \right) = \left(g^c x, y\left(\sqrt{g}\right)^{3c-1}\left(\sqrt{g}\right) \right) = \left(\hat{x}, \hat{y}\sqrt{g}\right)$$

With c odd the final expression is the form of a point on a complimentary EC (g is a generator and hence a quadratic non-residue), hence every point on the E_1 is changed to a point on $\bar{E}_2$, and so $\#\bar{E}_2 = p + 1 - t$. But $\bar{E}_2$ is the complimentary curve of E_2, meaning $\#E_2 = p + 1 + t$. ▌

This theorem basically doubles the number of curves where the number of points is known, all from one original curve, E_1. Now all curves isomorphic to E_1 and E_2 are known.

The next two theorems present some interesting, but not practically useful properties of elliptic curves.

Theorem 7 *Let E(a, b) be an elliptic curve over GF(p) with a,b ∈ GF(p). Then the following summations hold:*

(a) $\sum_{\substack{b=0 \\ a=const}}^{b=p-1} \#E(a,b) = p(p+1)$

That is the sum of the order of all elliptic curves with constant a, and b varying over GF(p) is equal to p(p+1).

(b) $\sum_{\substack{a=0 \\ b=0}}^{a=p-1} \#E(a,0) = p(p+1)$

That is the sum of the order of all elliptic curves with b = 0 and a varying over GF(p) is equal to p(p+1).

(c) $\sum_{\substack{a=0 \\ b=q.r.}}^{a=p-1} \#E(a,b) = p(p+2)$

That is the sum of the order of all elliptic curves with b a quadratic residue (q.r.) and a varying over GF(p) is equal to p(p+2).

(d) $\sum_{\substack{a=0 \\ b=q.n.r.}}^{a=p-1} \#E(a,b) = p^2$

That is the sum of the order of all elliptic curves with b a quadratic non-residue (q.n.r.) and a varying over GF(p) is equal to p^2.

Proof.

(a) The number of points on an elliptic curve can be calculated by $\#E(a,b) = 1 + \sum_{x=0}^{x=p-1} \left(1 + \left(\frac{x^3+ax+b}{p}\right)\right)$ where the expression in the big brackets $\left(\left(\frac{x^3+ax+b}{p}\right)\right)$ represents the Jacobi symbol. Summing this over all b with a fixed gives:

$$\sum_{\substack{b=0 \\ a=const}}^{b=p-1} \left[1 + \sum_{x=0}^{x=p-1} \left(1 + \left(\frac{x^3+ax+b}{p}\right)\right)\right] = \sum_{b=0}^{b=p-1} \left[1 + p + \sum_{x=0}^{x=p-1} \left(\frac{x^3+ax+b}{p}\right)\right]$$

$$= p(p+1) + \sum_{x=0}^{x=p-1} \sum_{b=0}^{b=p-1} \left(\frac{x^3+ax+b}{p}\right)$$

$$= p(p+1)$$

The double sum (in the second to last line) goes to zero as for a particular value of x all the Jacobi symbols of all elements of GF(p) are being summed. To see this consider the term $w = x^3+ax$ in the inner sum, it is constant. Now the sum of $\left(\frac{w+b}{p}\right)$ for all b's in GF(p) is zero, as $w+b$ will be a permutation of GF(p).

(b) Using the same argument as in (a), the equation becomes

$$\sum_{\substack{a=0\\b=0}}^{a=p-1}\left[1+\sum_{x=0}^{x=p-1}\left(1+\left(\frac{x^3+ax}{p}\right)\right)\right]=\sum_{a=0}^{a=p-1}\left[1+p+\sum_{x=0}^{x=p-1}\left(\frac{x^3+ax}{p}\right)\right]$$

$$= p(p+1)+\sum_{x=0}^{x=p-1}\sum_{a=0}^{a=p-1}\left(\frac{x^3+ax}{p}\right)$$

$$= p(p+1)$$

Again, the double sum goes to zero as x^3 is constant and with x constant and a co-prime to p then ax is a permutation on GF(p). So the sum of the Jacobi symbols will equate to zero. Note, when $x = 0$ the Jacobi symbol will also be zero.

(c) Using the exactly same argument as in (b), but adding a constant b term (where b is a q.r.). The only difference is when $x = 0$, to show this difference the double sum is evaluated below:

$$\sum_{x=0}^{x=p-1}\sum_{\substack{a=0\\b=q.r.}}^{a=p-1}\left(\frac{x^3+ax+b}{p}\right)=\sum_{\substack{a=0\\b=q.r.}}^{a=p-1}\left(\frac{b}{p}\right)+\sum_{x=1}^{x=p-1}\sum_{\substack{a=0\\b=q.r.}}^{a=p-1}\left(\frac{x^3+ax+b}{p}\right)$$

$$= p$$

Now putting the result of the double sum back into the same line of equations as (b), then we get $p(p+1) + p = p(p+2)$.

(d) Now b is a q.n.r, hence the double sum from (c) becomes $-p$, giving $p(p+1) - p = p^2$. ∎

Theorem 8 *Let E be an elliptic curve over GF(p), p a prime, then the values for the number of points on the curve #E have the following form when either a or b = 0 :*

(a) When $p \equiv 1$ *mod* 6 *and g is a generator for GF(p), then*

- *(i)* $\#E(0, g^c) = 6k$; *and* g^c *is a quadratic residue and* $g^c \equiv -g^{3d}$ *mod p for some d.*
- *(ii)* $\#E(0, g^{c+1}) = 6l + 1$
- *(iii)* $\#E(0, g^{c+2}) = 6m + 3$
- *(iv)* $\#E(0, g^{c+3}) = 6n + 4$
- *(v)* $\#E(0, g^{c+4}) = 6o + 3$
- *(vi)* $\#E(0, g^{c+5}) = 6r + 1$

and $k + l + m + n + o + r + 2 = p + 1$.

(b) When $p \equiv 5$ *mod* 6 *then* $\#E(0, b) = p+1$.

(c) When $p \equiv 1$ *mod* 4 *and g is a generator for GF(p) then*

- *(i)* $\#E(g^c, 0) = 4s$ *and* g^c *is a quadratic residue*
- *(ii)* $\#E(g^{c+1}, 0) = 4t + 2$
- *(iii)* $\#E(g^{c+2}, 0) = 4u$
- *(iv)* $\#E(g^{c+3}, 0) = 4v + 2$

and $s + t + u + v + 1 = p + 1$.

(d) When $p \equiv 3$ *mod* 4 *then* $\#E(a, 0) = p$.

Proof.

(a)(i) Consider the following:

- When $p \equiv 1 \bmod 6$ ($\equiv 1 \bmod 3$) then 3 real cube roots of x^3 exist, hence if $x^3 + g^c$ is a quadratic residue then the other roots of x^3 will also make $x^3 +$

g^c a qr, each of which contribute 2 points, giving a total of 6. There are k - 1 such triplets.

- When $x = 0$, since g^c is a qr this yields 2 points.
- Since $g^c \equiv -g^{3d} \bmod p$, meaning g^c has a cube root hence there are 3 values of x which make $x^3 - g^{3d} = 0 \bmod p$, yielding 3 points.

Adding the above together gives $\#E(0, g^c) = 6(k\text{-}1) + 2 + 3 + 1$ point at infinity = 6k.
(ii) Now g^{c+1} is not a qr and there is no d such that $g^c \equiv -g^{3d}$, hence $\#E(0, g^c) = 6l + 1$.
(iii) Now g^{c+2} is a qr and there is no d such that $g^c \equiv -g^{3d}$, hence $\#E(0, g^c) = 6m + 3$.
(iv) Now g^{c+3} is not a qr and there is a d such that $g^c \equiv -g^{3d}$, hence $\#E(0, g^c) = 6n + 4$.
(v) Now g^{c+4} is a qr and there is no d such that $g^c \equiv -g^{3d}$, hence $\#E(0, g^c) = 6o + 3$.
(vi) Now g^{c+5} is a not qr and there is no d such that $g^c \equiv -g^{3d}$, hence $\#E(0, g^c) = 6r + 1$.

Summing these 6 orders gives $(6k) + (6l+1) + (6m+3) + (6n+4) + (6o+3) + (6r+1) = 6(k+l+m+n+o+r+2)$, and there are $(p\text{-}1)/6$ such groups (as $p \equiv 1 \bmod 6$), summing these gives $(p - 1)(k + l + m + n + o + r + 2)$. From Theorem 7 the sum of $\#E(0, b)$ for all b is $p(p+1)$. Hence

$$\begin{aligned} p(p+1) &= (p - 1)(k + l + m + n + o + r + 2) + \#E(0, 0) \\ &= (p - 1)(k + l + m + n + o + r + 2) + (p + 1) \\ \Rightarrow k + l + m + n + o + r + 2 &= p + 1. \end{aligned}$$

(b) Well known result, see [45]. When $p \equiv 5 \bmod 6$ ($\equiv 2 \bmod 3$) then x^3 is a permutation on GF(p) and so is $x^3 + b$, hence there will be $(p - 1)/2$ qr yielding $p - 1$ points, plus one point when $x^3 + b = 0$, and the point at infinity, totalling $p + 1$ points.

(b) (i) Consider the following:

- When $p \equiv 1 \bmod 4$, if x makes $x^3 + ax$ a qr then so does $-x$, each x yields 2 points so both give 4 points. There are s - 1 such pairs.
- Now g^c is a qr hence $-g^c$ is a qr then there are 2 x values such that $x^3 - g^c x = x(x^2 - g^c) = 0$, yielding 2 points.
- $x = 0$ will give 1 point.

Totalling $\#E(g^c, 0) = 4(s\text{-}1) + 2 + 1 + 1$ point at infinity = $4s$.
(ii) Now g^c is not a qr hence $\#E(g^c, 0) = 4t + 2$.
(iii) See (i).
(iv) See (ii).

Summing the four cases $4(s + t + u + v + 1)$, there are $(p - 1)/4$ such quartets, so again

$$\begin{aligned} p(p + 1) &= (p - 1)(s + t + u + v + 1) + \#E(0,0) \\ &= (p - 1)(s + t + u + v + 1) + (p + 1) \\ \Rightarrow s + t + u + v + 1 &= p+1. \end{aligned}$$

(d) Well known result, see [45]. When $p \equiv 3 \bmod 4$, if x makes $x^3 + ax$ a qr then $-x$ makes it a quadratic non-residue. Hence there are $(p - 1)/2$ qr each yielding 2 points for a total of $(p - 1)$ points, plus one point at $x = 0$, adding the point at infinity gives p+1 points. ▌

Corollary: *Using Theorem 8 and its notation, then a curve* $E: y^2 = x^3 + g^c x + g^d$ *with* $\#E = p + 1 - t$, *and so from Theorem 6 the curve* $\overline{E}: y^2 = x^3 + g^{c+2}x + g^{d+3}$ *has*

$\#\overline{E} = p+1+t$. *Applying this to Theorem 8 (a) and (c) the following relations can be derived:*

(a) *(i)* $k+n=\frac{p-1}{3}$

(ii) $l+o=\frac{p-1}{3}$

(iii) $m+r=\frac{p-1}{3}$

(b) *(i)* $s+u=\frac{p+1}{2}$

(ii) $v+t=\frac{p-1}{2}$

Proof.
(a) (i) Since the number of points on the curves from Theorem 8 (a)(i) and (a)(iv) can be related (from Theorem 6), then:

$$6k = p+1-t$$

$$6n+4 = p+1+t$$

Adding these two equations and simplifying yields:

$$6(k+n)+4 = 2p+2$$

$$k+n=\frac{p-1}{3}$$

(ii) Same as (i) using curves from Theorem 8 (a)(ii) and (a)(v).
(iii) Same as (i) using curves from Theorem 8 (a)(iii) and (a)(vi).
(b) (i) Same as (a)(i) using curves from Theorem 8 (c)(i) and (c)(iii).
(ii) Same as (a)(i) using curves from Theorem 8 (c)(ii) and (c)(iv). ▌

Conjecture: The k and n from Theorem 8 (a)(i) and (iv) are both even.

Progress. We know that $k + n$ is even as $(p - 1)/3$ is even, hence either k and n are even or they are odd. The conjecture arose from the observation (from about 30 consecutive cases) that both k and n were always divisible by 4. ▌

Conjecture: For $p \equiv 5 \bmod 6$ and $p \equiv 1 \bmod 6$, and some generator g:

$$\#E(0, g^d) + \#E(0, g^{d+2}) + \#E(0, g^{d+4}) = 3(p + 1)$$

Progress. Trivially true when $p \equiv 5 \bmod 6$. Otherwise saying:

$$3(p + 1) = (6k) + (6m + 3) + (6o + 3) = 6(k + m + o + 1)$$

$$\Rightarrow \frac{(p+1)}{2} = k+m+o+1$$

and

$$3(p + 1) = (6l + 1) + (6n + 4) + (6r + 1) = 6(l + n + r + 1)$$

$$\Rightarrow \frac{(p+1)}{2} = l+n+r+1$$ ▌

2.6 EC-RSA and EC-El-Gamal Cryptosystems

As with the original RSA, EC-RSA has its security based on the difficulty of factoring. This is achieved by carefully choosing primes such that when the EC over $\mathbb{Z}_N$ is used, the number of points on the curve is known.

EC-RSA
Preliminaries
Choose p and q so that both are congruent to 2 mod 3 and compute $N = pq$. This ensures that the number of points on the curve $\#E(0, b) = (p + 1)(q + 1)$, for any b.
Key Generation
Randomly select e such that $\gcd(e, (p + 1)(q + 1)) = 1$. Compute d such that $ed = 1 \bmod (p + 1)(q + 1)$.
Encryption
Encrypt message m 1) Convert m to an EC point (x, y) 2) Calculate $e \cdot (x, y) = (c_1, c_2)$ Ciphertext (c_1, c_2)
Decryption
1) Calculate $d \cdot (c_1, c_2) = (x, y)$

Supersingular curves are chosen in F_p and F_q as the number of points is already known for these curves and hence the number of points on the curve over $\mathbb{Z}_N$ can easily be determined. The security is based on factoring as factoring N breaks the cryptosystem. For more elliptic curve cryptosystems based on RSA see [17, 35, 47].

The EC El-Gamal cryptosystem is analogous to the original El-Gamal cryptosystem and hence its security is based on the ECDL problem.

EC El-Gamal
Preliminaries
Choose a prime p and an EC curve $E(a, b)$ over F_p. Find a generator point G on the EC that generates a large group of points.
Key Generation
Randomly select an integer a as the private key and calculate $a \cdot G = P$ as the public key.
Encryption
Encrypt message m 1) Convert m to an EC point (x, y) 2) Randomly select $k \in F_p$ 3) Compute $k \cdot P = (\bar{x}, \bar{y})$ 4) $c_1 = x\bar{x}, \quad c_2 = y\bar{y}$ Ciphertext $(k \cdot G, c_1, c_2)$
Decryption
1) Calculate $a \cdot k \cdot G = (\bar{x}, \bar{y})$ 2) $x = \frac{c_1}{\bar{x}}, \quad y = \frac{c_2}{\bar{y}}$

An actual implementation of this cryptosystem would be over the field F_{2^m} as this leads to faster computations on a computer. One of the main issues in using a scheme like this is determining the number of points. The security is based on the ECDL problem and the difficulty of this is based on the size of the group, which for elliptic curves corresponds to the number of points.

For both EC-RSA and EC- El-Gamal there are various methods to make the schemes more efficient and reduce overhead, see [29].

2.7 Notions of Security

This section presents the notions of security for an encryption scheme. First the notation used to formally present the notions is given. Then the types of attacks on an encryption scheme are given, followed by the goals of an attacker.

All definitions are taken from [3], it is the most complete treatise on this area and the reader is encouraged to be familiar with the paper.

2.7.1 Notation

If A is a probabilistic algorithm, then $A(x_1, x_2, \ldots; r)$ is the result of running A on inputs $x_1, x_2, \ldots$ and random number r. We let $y \leftarrow A(x_1, x_2, \ldots)$ denote the experiment of picking r at random and letting y be $A(x_1, x_2, \ldots; r)$. If S is a set then $x \leftarrow S$ is the operation of picking an element uniformly from S. If α is neither an algorithm nor a set then $x \leftarrow \alpha$ is a simple assignment statement. We say that y *can be output by* $A(x_1, x_2, \ldots)$ if there is some r such that $A(x_1, x_2, \ldots; r) = y$.

The notion describing an asymmetric scheme can be formally defined via a triple of algorithms, $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, where

- $\mathcal{K}$, the *key generation algorithm*, is a probabilistic algorithm that takes a security parameter $k \in \mathbb{N}$ (provided in unary) and returns a pair (pk, sk) of matching public and secret keys.
- $\mathcal{E}$, the *encryption algorithm*, is a probabilistic algorithm that takes a public key pk and a message $x \in \{0, 1\}^*$ to produce a ciphertext y.
- $\mathcal{D}$, the *decryption algorithm*, is a deterministic algorithm which takes a secret key sk and ciphertext y to produce either a message $x \in \{0, 1\}^*$ or a special symbol $\varnothing$ to indicate that the ciphertext was invalid.

Let 1^k be a string of k binary 1's. We require that for all (pk, sk) which can be output by $\mathcal{K}(1^k)$, for all $x \in \{0, 1\}^*$, and for all y that can be output by $\mathcal{E}_{pk}(x)$, we have that $\mathcal{D}_{sk}(y) = x$. We also require that $\mathcal{K}$, $\mathcal{E}$ and $\mathcal{D}$ can be computed in polynomial time. As the notation indicates, the keys are indicated as subscripts to the algorithms.

Recall that a function $\varepsilon: \mathbb{N} \rightarrow \mathbb{R}$ is *negligible* if for every constant $c \geq 0$ there exists an integer k_c such that $\varepsilon(k) \leq k^{-c}$ for all $k \geq k_c$.

2.7.2 Types of Attack

An adversary A is regarded as a pair of probabilistic algorithms, $A = (A_1, A_2)$. (A is polynomial time if both A_1 and A_2 are.) This corresponds to A running in two "stages". The basic idea is that in the first stage the adversary, given the public key, seeks and outputs some "test instance", and in the second stage the adversary is issued a challenge ciphertext y generated as a probabilistic function of the test instance. (In addition A_1 can output some state information s that will be passed to A_2)

Three types of attacks are considered under this setup.

In a *chosen-plaintext attack* (CPA) the adversary can encrypt plaintexts of her choosing. Of course a CPA is unavoidable in a public-key setting: knowing the public key, an adversary can, on her own, compute a ciphertext for any plaintext she desires.

The CPA attack is possible for every public-key cryptosystem. The original RSA and El-Gamal cryptosystems are only secure against this type of attack.

In a *non-adaptive chosen ciphertext attack* (CCA1) we give A_1 (the public key and) access to a decryption oracle, but we do not allow A_2 access to a decryption oracle. Intuitively, this is allowing an attacker to query the decryption oracle with ciphertexts before receiving the challenge ciphertext. Once the challenge is received the attacker is not allowed access to the decryption oracle.

The CCA1 attack could model a situation were an employee decrypts ciphertexts of her choice at her place of business when no one else is around, like at lunch-time or late at night, hence this attack is referred to as the *lunch-time* or *midnight* attack.

In an *adaptively chosen ciphertext attack* (CCA2) we continue to give A_1 (the public key and) access to a decryption oracle, but also give A_2 access to the same decryption oracle, with the only restriction that she cannot query the oracle on the challenge ciphertext y. This is an extremely strong attack model.

The CCA2 attack could model a situation where an attacker could send any ciphertext of their choice to an automated response server, which would return a message with the decrypted ciphertext. For example, an email automated response server, usually the received message (in this case the decrypted ciphertext) is also part of the reply email (with some check to see that the challenge ciphertext isn't responded too). Another situation could be one where a third party holds some encrypted key u and some credentials d, and anyone wanting to access the key needs to provide the correct credentials.

At one stage during the development of these attacks CCA1 was considered the strongest as some [16, 49] thought the notion of CCA2 was somewhat impractical, especially the condition that the adversary can ask the decryption any ciphertext apart from the challenge ciphertext. However, CCA2 is still a plausible attack and for a cryptosystem to be secure against this type of attacker it is indeed a strong condition.

It should be noted that these definitions are not attacks on a cryptosystem, but rather define the attacker we are considering. They basically define what the attacker has access to. Next we describe specific goals for the attacker to achieve.

2.7.3 Indistinguishability of Encryptions.

The classical goal of secure encryption is to preserve the privacy of messages: an adversary should not be able to learn from a ciphertext, information about its plaintext beyond the length of that plaintext. This notion is referred to as indistinguishability of encryptions (IND) and can be given a precise definition via the following simple experiment.

> "Algorithm A_1 is run on input the public key, *pk*. At the end of A_1's execution it outputs a triple (x_0, x_1, s), the first two components being messages which must be of the same length, and the last being state information (possible including *pk*) which the attacker wants to preserve. A random one of x_0 and x_1 is now selected, say, x_b. A "challenge" y is determined by encrypting x_b under *pk*. It is A_2's job to try to determine if y was selected as the encryption of x_0 or x_1, namely to determine the bit b. To make this determination A_2 is given the saved state s and the challenge ciphertext y." [3, pg. 32]

We simultaneously define IND with respect to CPA, CCA1, and CCA2. The only difference lies in whether or not A_1 and A_2 are given the decryption oracles. We let the string 'atk' be instantiated by any of the formal symbols 'cpa', 'cca1', and 'cca2', while ATK is then the corresponding formal symbol from CPA, CCA1, and CCA2. When we say $\mathcal{O}_i = \varepsilon$, where $i \in \{1, 2\}$, we mean $\mathcal{O}_i$ is the function which, on any input, returns the empty string, ε.

Definition 6 *[IND-CPA, IND-CCA1, IND-CCA2]* Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and let $A = (A_1, A_2)$ be an adversary. For atk $\in$ {cpa, cca1, cca2} and $k \in \mathbb{N}$ let

$$\mathrm{Adv}^{\mathrm{ind-atk}}_{A,\Pi}(k) = 2 \cdot \Pr\left[(pk, sk) \leftarrow \mathsf{K}(1^k); (x_0, x_1, s) \leftarrow A_1^{\mathcal{O}_1}(pk); b \leftarrow \{0,1\}; \right.$$
$$\left. y \leftarrow \mathsf{E}_{pk}(x_b): A_2^{\mathcal{O}_2}(x_0, x_1, s, y) = b\right] - 1$$

where

If atk = cpa then $\mathcal{O}_1(\cdot) = \varepsilon$	and $\mathcal{O}_2(\cdot) = \varepsilon$
If atk = cca1 then $\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$	and $\mathcal{O}_2(\cdot) = \varepsilon$
If atk = cca2 then $\mathcal{O}_1(\cdot) = \mathcal{D}_{sk}(\cdot)$	and $\mathcal{O}_2(\cdot) = \mathcal{D}_{sk}(\cdot)$

It is insisted that A_1 outputs x_0, x_1 with $|x_0| = |x_1|$. In the case of CCA2, it also insisted that A_2 does not ask its oracle to decrypt y. We say that Π is secure in the sense of IND-ATK if A being polynomial-time implies that $\mathrm{Adv}^{\mathrm{ind-atk}}_{A,\Pi}(\cdot)$ is negligible. ▌

2.7.4 Non-Malleability

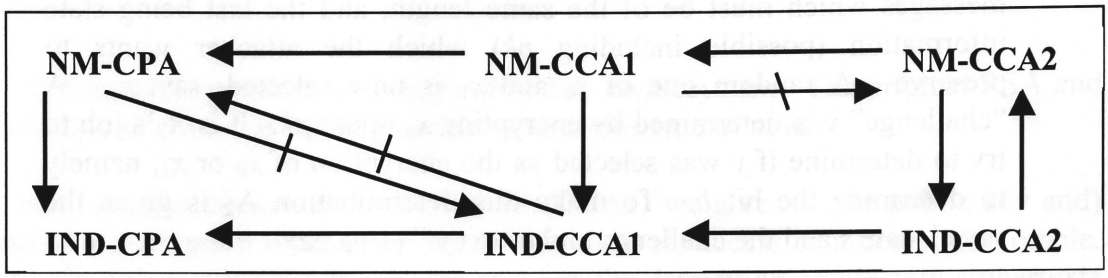
The goal non-malleability (NM) is not to ensure that no information about the message x can be recovered from the ciphertext (as in IND), but to ensure that an

adversary cannot create a ciphertext whose decryption is "meaningfully related" to the decrypted challenge ciphertext. The idea of "meaningfully related" is achieved through some mathematical relation $R(x, x')$.

We will not give the definition of [3] here, as this paper shows that NM is related to IND, and at least in the sense of CCA2 the two are equivalent. It should be noted that IND and NM are proved equivalent based on the definition of NM from [3], but also their definition implies a previous definitions from [19].

2.7.5 Relationships between Notions

Figure 2 shows the relationships between the notions of security as given in [3]. The arrows are implications and the hatched arrows represent separations which [3] actually prove. The reader is directed to their paper for the corresponding proofs.



These relations now allow us to only require a proof of security in the sense of IND-CCA2 (if that is our goal) and be sure that weaker, or equivalent, notions of security are achieved.

2.7.6 Notions of Security for Signature Schemes

As with encryption schemes there are notions of security for signature schemes. Again the notions of security are presented in terms of attacks and goals, however only informal definitions will be given as some of the notions are more difficult to capture than such simple tests such as IND. These definitions are from [25].

The following are a list of attacks on a signature scheme in order of increasing severity. Here A denotes the user whose signature method is being attacked.

- *Known-message attack (KMA).* The enemy is given access to a set of signatures for a set of messages $m_1, \ldots, m_t$. The messages are known to the enemy but are not chosen by him.
- *Generic chosen-message attack.* The enemy is allowed to obtain from A valid signatures for a chosen list of message $m_1, \ldots, m_t$ before he attempts to break A's signature scheme. The messages are chosen by the enemy before he sees A's public key and hence this attack is "generic" as it is independent of A's public key.
- *Directed chosen-message attack.* Similar to the generic chosen-message attack, but now the enemy is allowed access to A's public key before choosing a list of messages. This attack is still non-adaptive and is "directed" against a particular user, A.
- *Adaptive chosen-message attack (CMA).* The enemy is allowed to use A as an "oracle", to obtain signatures for messages that may depend on A's public key and on previously obtained signatures.

Next the goals of the attacker are given, these are the way in which the adversary would like to break the signature scheme. Goals are listed in order of decreasing severity.

- *A total break.* Compute A's secret trap-door information.
- *Universal forgery.* Find an efficient signing algorithm functionally equivalent to A's signing algorithm (based on possibly different but equivalent trap-door information).
- *Selective forgery.* Forge a signature for a particular message chosen a priori by the enemy.
- *Existential forgery.* Forge a signature for at least one message. The enemy has no control over the message whose signature he obtains, so it may be random or nonsensical. Consequentially this forgery may only be a minor nuisance to A.

The goal for any signature scheme is to be secure in the sense of existential forgery, that is the scheme should be 'not existentially forgeable' (NEF), against an adaptively chosen-message attack (NEF-CMA).

2.8 Plaintext Awareness

Plaintext Awareness (PA) was first defined in [4], but the definition was extended and developed in [3] so their definition shall be used. PA for an encryption scheme is a simple concept to fulfil: an adversary is unable to create a ciphertext without knowing the underlying plaintext. Currently the definition of PA only exists in the Random Oracle (RO) model.

The RO model assumes everyone has access to a public oracle that outputs perfectly random information. In reality, hash functions and PRNG are used. The RO model is discussed in [5] with a counter argument in [13].

The definition uses the following notation. By $(C, y) \leftarrow \text{run } B^{\mathsf{E}_{pk}}(pk)$ we mean the following. Run B on input pk and oracle $\mathcal{E}_{pk}$. Form into a list $C = (y_1, y_2, \ldots, y_{qE})$ the answers (ciphertexts) received as a result of $\mathcal{E}_{pk}$-queries. (The messages that formed the actual queries are *not* recorded.) Finally, record b's output, y.

Definition 7 – (*PA*) Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme, let B be an adversary, and let K be an algorithm (the "knowledge extractor"). For any $k \in \mathbb{N}$ let

$$\text{Succ}^{pa}_{K,B,\Pi}(k) \stackrel{def}{=}$$

$$\Pr[H \leftarrow \text{Hash}; (pk, sk) \leftarrow \mathsf{K}(1^k);$$

$$(hH, C, y) \leftarrow \text{run } B^{H, \mathsf{E}^H_{pk}}(pk) : K(hH, C, y, pk) = \mathsf{D}^H_{sk}(y)]$$

We insist $y \notin C$; that is, B never outputs a string y which coincides with the value returned from some $\mathcal{E}^{\mathrm{H}}_{\mathrm{pk}}$-query. We say that K is a $\lambda(k)$-extractor if K has running time polynomial in the length of its inputs and for every adversary B, $\text{Succ}^{pa}_{K,B,\Pi}(k) \geq \lambda(k)$. We say that Π is secure in the sense of PA if Π is secure in the sense of IND-CPA and there exists a $\lambda(k)$-extractor K where $1 - \lambda(k)$ is negligible. ∎

The purpose of the knowledge extractor is to simulate the decryption oracle, except without access to a private key. If the knowledge extractor succeeds at recovering the plaintext from a challenge ciphertext provided by the adversary, then this means the adversary must know the plaintext as it can be recovered from just examining oracle queries and the public key.

The main point of showing that an encryption scheme is PA is the following theorem.

Theorem 9 – *[PA $\Rightarrow$ IND-CCA2] If encryption scheme Π is secure in the sense of PA then it is secure in the RO sense of IND-CCA2.* ∎

This gives a very useful method for showing an encryption scheme is secure in the sense of IND-CCA2, it just needs to be shown it is secure in the sense of IND-CPA and define a knowledge extractor for it, thus complying with the definition of PA.

CHAPTER 3 REVIEW

When contributing to an area of knowledge it is essential to know and appreciate the work that has already been done. This chapter does this by highlighting important contributions, which lay out the development of provably secure cryptosystems.

3.1 Timeline

- 1979
  - Rabin [54] – Devised CPA scheme that is equivalent to factoring, but is insecure against CCA1 attack. This lead to the belief in a "paradox" – any scheme whose security broke down to finding the private key from the public key could not be secure against CCA1.
- 1984
  - Goldwasser, Micali, Rivest [26] – They showed the paradox false by creating the first signature scheme secure against a CMA attack. The result held for encryption schemes as well.
  - Goldwasser, Micali [27] – They define notions of probabilistic encryption, indistinguishability of encryptions (IND) and semantic security, and show the equivalence between them.
- 1988
  - Blum, Feldman, Micali [9] – Using non-interactive zero-knowledge proofs they show how security in the sense of CCA1 can be achieved.
- 1990
  - Naor, Yung [49] – The first cryptosystem secure against CCA1, but encryption is bit-by-bit and has massive ciphertext expansion.
- 1991
  - Damagard [16] – Designs a secure scheme against CCA1, and practical, but later discovered insecure against CCA2.
  - Rackoff, Simon [55] – Defined the adaptively chosen ciphertext attack (they didn't call it this though).
  - Dolev, Dwork, Naor [19] – Defined non-malleable (NM) cryptography.
- 1993
  - Zheng-Seberry [70] – Claimed security against chosen ciphertext attack (CCA2), but not in emerging model on security (IND).

- Bellare, Rogaway [5] – Use the random oracle model to create a secure scheme in CCA2 sense.

- 1995
  - Bellare, Rogaway [4] – They develop OAE, which is secure against CCA2 but uses the random oracle model. OAE can use any one-way trapdoor permutation (eg RSA). They develop the notion of plaintext awareness (PA)
- 1997
  - Zheng [68] – Creates the idea of Signcryption.
- 1998
  - Bellare, Desai, Pointcheval, Rogaway [3] – Seminal paper that specifies the relations among the notions of security. They show IND-CCA2 ≡ NM-CCA2 and many other implications. Results true in standard model and RO. They also refine the PA definition.
  - Cramer, Shoup [15] – The first practical and provably secure scheme in CCA2 sense under only standard assumptions.
- 2000
  - Pointcheval [52] – Develops method for creating a scheme provable secure in CCA2 sense from any trapdoor one-way function in RO model.

Cryptographic schemes that are provably secure are increasingly becoming the norm nowadays, hence the review given here will be of those schemes that are provably secure, at least under some assumptions. Zheng-Seberry has also been added to this list, even though it is not provably secure, however it was an integral stepping stone in this research. There are absences from this review, such as a scheme by Fujisaki and Okamoto [24], this has been done solely for reasons of conciseness.

3.1.1 Naor and Yung

A review of provably secure schemes could start as far back as Rabin's scheme in 1979, but a base security standard of CCA1 will be set. Naor and Yung were the first to devise a scheme that was secure against CCA1; their scheme was actually built upon any bit encryption scheme secure in the CPA sense.

Naor-Yung
Preliminaries
A PKC triple $(\mathcal{G}, \mathcal{E}, \mathcal{D})$, with $\mathcal{G}$ the key generator, $\mathcal{E}$ an encryption algorithm that encrypts bits and $\mathcal{D}$ a decryption algorithm. A non-interactive zero-knowledge proof (NIZKP) system triple $(\mathcal{P}, \mathcal{U}, \mathcal{V})$, where $\mathcal{P}$ and $\mathcal{V}$ are two parties, one to prove membership and one to verify, and $\mathcal{U}$ is a distribution.
Key Generation
$\mathcal{G}$ on input $n \in \mathbb{N}$ is run twice to produce two sets of public and private keys (x_{R1}, y_{R1}) and (x_{R2}, y_{R2}). Generate $R \in_R \mathcal{U}(n)$. Public key is (y_{R1}, y_{R2}, R) and private key is (x_{R1}, x_{R2}).
Encryption
To encrypt a message $m = b_1, b_2, \ldots, b_k$ For each $1 \leq i \leq k$ 1) Generate $r_{i_1}, r_{i_2} \in_R \{0,1\}^{p(n)}$ ($p(n)$ a polynomial in n)

2) Compute $c_i = E_{y_{R1}}(b_i, r_{i_1}), E_{y_{R2}}(b_i, r_{i_2})$ 3) Run $\mathcal{P}$ on c_i with witness (r_{i_1}, r_{i_2}) and string R to get p_I The encrypted message is $(c_1, p_1), (c_2, p_2), \ldots, (c_k, p_k)$.
Decryption
For each $1 \leq i \leq k$ 1) Verify that c_i is consistent by running the verifier $\mathcal{V}$ on c_i, p_i, R 2) If $\mathcal{V}$ accepts, then retrieve b_i by computing either $D_{x_{R1}}(E_{y_{R1}}(b_i, r_{i_1}))$ or $D_{x_{R2}}(E_{y_{R2}}(b_i, r_{i_2}))$. Otherwise the output is null.

This scheme, while theoretically sound, would never be used in practice as the ciphertext expansion is enormous. If the output of encryption scheme was of size q (say the underlying group) then the ciphertext would be of size $2kq + |p|$. Naor-Yung suggest an encryption scheme from [27] based on quadratic residues, now if the size of the underlying group was 512 bits, and we wanted to send a message of 1 kilobyte, then the ciphertext would be >1 Megabyte!

The proof for this scheme requires knowledge of NIZKP's and since this is the only scheme that requires such knowledge, the reader is referred to Naor-Yung's paper [49] for the proof.

3.1.2 Damagård

Damagard was the first to come up with an encryption scheme that was provably secure against CCA1 (under some reasonable assumptions) and was very practical. However, Zheng-Seberry was to later show that Damagard's scheme was completely insecure against CCA2.

Damagard actually devised two schemes, one deterministic and loosely based on Rabin's scheme and the other probabilistic and based on the El-Gamal/Diffie-Hellman scheme. Only the latter scheme is given here.

Damagård
Preliminaries
Working modulo a large prime p, with generator g.
Key Generation
Generate two secret keys $x_{R1}, x_{R2} \in \mathrm{GF}(p)$ Generate two public keys $y_{R1} = g^{x_{R1}} \bmod p, y_{R2} = g^{x_{R2}} \bmod p$
Encryption
Choose $r \in_R \mathrm{GF}(p)$, encrypt message m $E(m) = (w^r, g^r, m \oplus y^r)$
Decryption
$D(c_1, c_2, c_3) = \begin{cases} c_3 \oplus c_2^{x_{R1}} & \text{if } c_1 = c_2^{x_{R2}} \\ \varnothing & \text{otherwise} \end{cases}$

3.1.3 Zheng-Seberry

Like Damagard, Zheng-Seberry developed schemes that they considered not only secure but also practical. They extended Damagard's work by highlighting the point that although security against CCA1 seemed sound Damagard's scheme was completely insecure against CCA2. Zheng-Seberry showed that by taking the parameter $c_3 = m \oplus y^r$, a random message m' and calculating $c_3' = m' \oplus c_3$, then by passing (c_1, c_2, c_3') to the decryption oracle, it would return $m \oplus m'$ from which m can easily be found.

Zheng-Seberry then went on to describe a scheme they considered secure against CCA2, and presented a proof of security based on some reasonable assumptions. They actually presented three schemes, two very similar except one used a one-way hash function and the other universal one-way hash functions, whilst the third incorporated digital signatures. Present here is the scheme that used just a one-way hash function.

Zheng-Seberry
Preliminaries
Consider messages of length n, a one-way hash function H with output length k_0 and a PRNG G with output length $n + k_0$. Operations are modulo p and there is a generator g.
Key Generation
Private key is $x_R \in \mathrm{GF}(p)$ and public key is $y_R = g^{x_R} \bmod p$.
Encryption
Encrypt message m 1) $x \in_R [1, p\text{-}1]$ 2) $z = G\left(y_R^x\right)_{[1\ldots(n+k_0)]}$ 3) $t = \mathrm{H}(m)$ 4) $c_1 = g^x$ 5) $c_2 = z \oplus (m \| t)$ Ciphertext is (c_1, c_2)
Decryption
1) $z' = G\left(c_1^{x_R}\right)_{[1\ldots(n+k_0)]}$ 2) $w = z' \oplus c_2$ 3) $m = w_{[1\ldots n]}$ 4) $t' = w_{[(n+1)\ldots(n+k_0)]}$ 5) if $\mathrm{H}(m) = t'$ then output m else output $\varnothing$

Unfortunately the proof of security for this scheme doesn't use what has become the standard notions of security, security against IND or NM. The Zheng-Seberry paper is discussed in more detail in 0.

3.1.4 Bellare and Rogaway – OAE

Bellare and Rogaway develop Optimal Asymmetric Encryption (OAE) as the answer to finding a practical yet provably secure encryption scheme. However, the proof of security relies on the use of the Random Oracle (RO) model, which states that instead

of having real hash functions or PRNG, perfect random oracles exist. At first this may seem to provide no guarantees at all, but Bellare and Rogaway present convincing arguments as to the usefulness of using this model, not the least of which is that it yields extremely practical schemes with almost provable security.

The scheme they develop is based on the use of any trapdoor permutation (of which RSA is the best known), making their scheme very general. The scheme outlined here was to later become part of the PKCS #1 v2.1: RSA Cryptography Standard (1999). Although see [8] for an attack.

OAE (Bellare-Rogaway)
Preliminaries
A k-bit trapdoor permutation and its inverse (f, f^{-1}). Messages are of length n, and a random number of length k_0, so that $k = n + k_0$. Two random oracles, H taking strings of length n to strings of length k_0, and G taking strings of length k_0 to strings of length n.
Key Generation
As appropriate for the chosen permutation f.
Encryption
To encrypt a message m 1) Choose random r of length k_0 2) $s = m \oplus G(r)$ 3) $t = r \oplus H(s)$ 4) $w = s \parallel t$ 5) $y = f(w)$ Ciphertext is y
Decryption
1) $w = f^{-1}(y)$ 2) $s = w_{1..n}$ 3) $t = w_{n+1...n+k_0}$ 4) $r = t \oplus H(s)$ 5) $m = s \oplus G(r)$

Bellare and Rogaway also presented an even more ambitious scheme similar to the one above, but this time secure in the sense of PA. Their PA scheme is virtually the same as the one above except that $n = k - k_0$, becomes $n = k - k_0 - k_1$ and m is replaced with $m0^{k_1}$, where 0^{k_1} is a string of k_1 zeroes. Then during decryption $m0^{k_1}$ is recovered and if the last k_1 bits are checked to be the string of zeroes, the output is the message, otherwise $\varnothing$.

3.1.5 Zheng

As hinted at in the Zheng-Seberry paper a digital signature can be incorporated into an encryption scheme, Zheng formalises this idea by introducing the concept of 'signcryption'. The idea is basically to merge confidentiality and authentication in one scheme such that it performs better than when confidentiality and authentication are performed separately (see Chapter 5). Zheng's scheme involves the use of keyed hashing and a private key cipher, it is outlined below.

Zheng
Preliminaries
Use GF(p) with large prime factor q and generator g of order q. Need a keyed hashing algorithm KH and a private key encryption cipher E and decryption cipher D.
Key Generation
Private key of sender is $x_S \in$ GF(p) and public key is $y_S = g^{x_S} \bmod p$. Private key of receiver is $x_R \in$ GF(p) and public key is $y_R = g^{x_R} \bmod p$.
Signcryption
To signcrypt a message m 1) Choose $x \in_R [1...q]$ 2) $k = y_R{}^x \bmod p$ 3) Split k into k_1 and k_2 of appropriate length 4) $r = KH_{k_1}(m)$ 5) $s = x / (r + x_S) \bmod q$ (alternatively $s = x / (1 + rx_S) \bmod q$) 6) $c = E_{k_2}(m)$ Signcrypted-text is (c, r, s)
Unsigncryption
1) $k = \left(y_S \cdot g^r\right)^{s \cdot x_R}$ (alternatively $k = \left(g \cdot y_S^r\right)^{s \cdot x_R}$) 2) Split k into k_1 and k_2 3) $m = D_{k_2}(c)$ 4) Accept m if $r \equiv KH_{k_1}(m)$ else output $\varnothing$

Zheng does not provide a formal proof for this scheme, but the scheme is included in this review as it is the first paper to deal with this concept and hence very relevant to this thesis. In fact Zheng did not even base his arguments for security in terms of formal notions of security, one of the major goals of this thesis is to provide formal notions of security for signcryption (Chapter 5).

Note, if the output of KH is of size $|q|$, then the ciphertext has size $2|q| + |p|$, and overall there are 4 operations (not including symmetric key cipher calls).

3.1.6 Cramer-Shoup

Cramer-Shoup were the first to create a practical scheme that was provably secure in the sense of IND-CCA2 under standard assumptions (not using the RO model). The scheme is practical although not *as* practical as schemes based on the RO model, like OAE. However, the promise of provable security would out-way the benefits of better efficiency for many applications.

The proof of Cramer-Shoup is very strong, it assumes there is an adversary that can break the scheme, and then describes how to construct a simulator that uses the adversary to create a statistical test for the Diffie-Hellman decision problem. The

proof is strong as it holds true even if the adversary is powerful (not to be confused with the scheme being secure against a powerful adversary).

Cramer-Shoup
Preliminaries
Group G of prime order q, where q is large. Also need a universal one-way hash function H.
Key Generation
Choose randomly $x_1, x_2, y_1, y_2, z \in_R \mathbb{Z}_q$ which constitute the private key. Choose randomly two elements $g_1, g_2 \in_R G$, then calculate $c = g_1^{x_1} g_2^{x_2}$, $d = g_1^{y_1} g_2^{y_2}$, $h = g_1^z$, then public key is (g_1, g_2, c, d, h).
Encryption
Encrypt a message m 1) Randomly choose $r \in \mathbb{Z}_q$ 2) $u_1 = g_1^r$, $u_2 = g_2^r$ 3) $e = h^r m$ 4) $\alpha = \mathrm{H}(u_1, u_2, e)$ 5) $v = c^r d^{r\alpha}$ Ciphertext (u_1, u_2, e, v)
Decryption
1) $\alpha = \mathrm{H}(u_1, u_2, e)$ 2) If $u_1^{x_1+y_1\alpha} u_2^{x_2+y_2\alpha} \equiv v$ then output $m = \frac{e}{u_1^z}$ else output $\varnothing$

3.1.7 Pointcheval

OAE was a generic construction of an IND-CCA2 secure scheme from any one-way permutation, however there are just not that many one-way permutations. Pointcheval showed a generic construction for an IND-CCA2 secure scheme based on any partially trapdoor one-way function, of which many exist. As with OAE, the proofs are only valid in the RO model.

Pointcheval
Preliminaries
A partially trapdoor one-way function f, and another function that partially inverts it g. Work over a group appropriate for the function. A hash function H and PRNG G.
Key Generation
As appropriate for f.
Encryption
Encrypt a message m 1) Randomly choose r and s of appropriate size 2) $a = f(r, \mathrm{H}(m \| s)$ 3) $b = (m \| s) \oplus \mathrm{G}(r)$ Ciphertext is (a, b)

Decryption
1) $r = g(a)$ 2) $M = b \oplus G(r)$ 3) If $a = f(r, H(M))$ then $m = M_{[1...\|m\|]}$ else output $\varnothing$

3.2 Signature Schemes

Some signature schemes are presented here that are considered secure and have the advantage that the signature they generate is small.

3.2.1 Schnorr

Developed by Schnorr [58] in 1989 as a result of an identification protocol, and is more secure than the original El-Gamal signature scheme.

Schnorr
Preliminaries
Working over GF(p) with q being a large prime factor $p - 1$. Also need a generator g of order q. A hash function H.
Key Generation
Choose a random private key $x_S \in [1...q]$ and then calculate public key $y_S = g^{-x_S} \bmod p$.
Signature
Sign a message m 1) Randomly choose $r \in [1...q]$ 2) $x = g^r \bmod p$ 3) $e = H(x, m)$ 4) Calculate $y = r + x_S e \bmod q$ Signature is (e, y)
Verification
1) $x' = g^y y_S^e \bmod p$ 2) If $e = H(x', m)$ then output *true* else output *false*.

3.2.2 Digital Signature Standard (DSS)

Developed by NIST (National Institute of Standards and Technology) [51] in 1994 for use as a standard for digital signatures. Another variant of the El-Gamal signature scheme.

DSS
Preliminaries
A prime modulus, p, where $2^{L-1} < p < 2^L$ for $512 \le L \le 1024$ and L a multiple of 64. Also, q, a prime divisor of $p - 1$, where $2^{159} < q < 2^{160}$. A generator $g = h^{(p-1)/q} \bmod p$, where h is an integer $1 < h < p$ and $g > 1$. A hash function H (standard recommends SHA-1).

Key Generation
Randomly choose a secret key x_S where $0 < x_S < q$. Calculate the public key $y_S = g^{x_S} \bmod p$.
Signature
Sign message m 1) Choose random $k \in [1...q]$ 2) $r = (g^k \bmod p) \bmod q$ 3) $s = k^{-1}(\mathrm{H}(m) + x_S r) \bmod q$ Signature (r, s)
Verification
1) $w = s^{-1}$ 2) $u_1 = \mathrm{H}(m).w \bmod q$ 3) $u_2 = r.w \bmod q$ 4) $v = \left(y_S^{u_1} g^{u_2} \bmod q\right) \bmod p$ 5) If $v = r$ then output *true* else output *false*.

3.2.3 Pointcheval-Stern Modified El-Gamal

Pointcheval and Stern [53] present a method for proving the security of signature schemes in the RO model. They apply their method to a modified version of El-Gamal and prove its security against existential forgery for an adaptively chosen message attacker, in the RO model.

Modified El-Gamal
Preliminaries
Work over GF(p) where p is a large prime with q a large prime factor of $p - 1$. Need a generator g of order q and hash function H.
Key Generation
Choose a random private key $x_S \in [1...q]$ and then calculate public key $y_S = g^{-x_S} \bmod p$.
Signature
Sign a message m 1) Choose random $k \in [1...q]$ 2) $r = g^k \bmod q$ 3) Solve $\mathrm{H}(m, r) = x_S r + ks \bmod q$ for s Signature $(r, \mathrm{H}(m, r), s)$
Verification
1) If $g^{H(m,r)} = y_S^r r^s \bmod q$ the output *true* else output *false*

CHAPTER 4 ZHENG-SEBERRY

The Zheng-Seberry (ZS) [70] encryption scheme was published in 1993 and was one of the first practical schemes that was considered secure against a CCA2 adversary. This chapter shows for the first time that a version of the ZS scheme is actually insecure against a CCA2 adversary. The ZS scheme is then modified to make it into a provably secure cryptosystem.

4.1 Review

The ZS paper presented three variants of an El-Gamal like cryptosystem. The three variants were described as 'immunising' the cryptosystem against a CCA2 adversary. The variants incorporated a one-way hash function (OWH), a universal hash function (UHF) and a digital signature (SIG). These variants are given below.

4.1.1 ZS-OWH

ZS-OWH
Preliminaries
Consider messages of length n, a one-way hash function H with output length k_0 and a PRNG G with output length $n + k_0$. Operations are modulo p and there is a generator g.
Key Generation
Private key is $x_R \in$ GF(p) and public key is $y_R = g^{x_R} \bmod p$.
Encryption
Encrypt message m 1) $x \in_R [1, p\text{-}1]$ 2) $z = G\left(y_R^x\right)_{[1\ldots(n+k_0)]}$ 3) $t = \mathrm{H}(m)$ 4) $c_1 = g^x$ 5) $c_2 = z \oplus (m \| t)$ Ciphertext is (c_1, c_2)
Decryption

1) $z' = G(c_1^{x_R})_{[1...(n+k_0)]}$
2) $w = z' \oplus c_2$
3) $m = w_{[1...n]}$
4) $t' = w_{[(n+1)...(n+k_0)]}$

If H(m) = t' then output m else output ∅

4.1.2 ZS-UHF

ZS-UHF
Preliminaries
Consider messages of length n, a (strong) universal hash function H_s that is indexed by a string of length k_0 and a PRNG G with output length $n + k_0$. Operations are modulo p and there is a generator g.
Key Generation
Private key is $x_R \in$ GF(p) and public key is $y_R = g^{x_R} \bmod p$.
Encryption
Encrypt message m 1) $x \in_R [1, p\text{-}1]$ 2) $r = y_R^x$ 3) $z = G(r)_{[1...n]}$ 4) $s = G(r)_{[(n+1)...(n+k_0)]}$ 5) $c_1 = g^x$ 6) $c_2 = H_s(m)$ 7) $c_3 = z \oplus m$ Ciphertext is (c_1, c_2, c_3)
Decryption
1) $r' = c_1^{x_R}$ 2) $z' = G(r')_{[1...n]}$ 3) $s' = G(r')_{[(n+1)...(n+k_0)]}$ 4) $m = z' \oplus c_3$ If $H_s(m) = c_2$ then output m else output ∅

4.1.3 ZS-SIG

ZS-SIG
Preliminaries
Consider messages of length n, a one-way hash function H with output length k_0 and a PRNG G with output length $n + k_0$. Operations are modulo p and there is a generator g.
Key Generation
Private key is $x_R \in$ GF(p) and public key is $y_R = g^{x_R} \bmod p$.
Encryption

Encrypt message m
1) $x \in_R [1, p\text{-}1]$
2) $k \in_R [1, p\text{-}1]$ such that $gcd(k, p-1) = 1$
3) $r = y_R^{x+k}$
4) $z = G(r)_{[1\ldots n]}$
5) $c_1 = g^x$
6) $c_2 = g^k$
7) $c_3 = (\mathrm{H}(m) - xr)/k \bmod (p-1)$
8) $c_4 = z \oplus m$
Ciphertext is (c_1, c_2, c_3, c_4)
Decryption
1) $r' = (c_1 c_2)^{x_R}$
2) $z' = G(r')_{[1\ldots n]}$
3) $m = z' \oplus c_4$
If $g^{H(m)} = c_1^{r'} c_2^{c_3}$ then output m else output $\varnothing$

4.1.4 Original Zheng-Seberry Proof

An economised version of the proof by ZS will be given, however the proof for ZS-SIG will be ignored, as this requires extra assumptions.

The proof relies on the intractability of the Diffie-Hellman Problem (DHP) [18] (or computational Diffie-Hellman problem (CDHP)), defined below.

Definition 8 – (*DHP*) Given y_1, y_2, g and p, where $y_1 = g^{x_1}$ and $y_2 = g^{x_2}$ for some x_1 and x_2 chosen randomly and independently from $[1 \ldots p-1]$, calculate $y = g^{x_1 x_2}$. ▌

The DHP is assumed computationally infeasible for any probabilistic polynomial time algorithm. The DHP is related to the DLP in that if one could solve the DLP then one could also solve the DHP, but the reverse result is unknown.

First the security against a CPA attacker will be shown. This is achieved by showing that no partial information about the message is leaked. For ZS-OWH consider that the message m is hidden by z and $z = g^{x \cdot x_R}$ where g^x and g^{x_R} are public. Then by an argument similar to [10], given z and z_1 (a truly random string), if the CDHP is intractable then no probabilistic polynomial time algorithm can distinguish z from z_1.

For ZS-UHF, consider that if z and s are truly random strings then neither $z \oplus m$ or $\mathrm{H}_s(m)$ reveal any partial information, and if z and s are independent then $z \oplus m$ and $\mathrm{H}_s(m)$ together, reveal no partial information. When z and s are the output of a pseudo-random number generator, to a probabilistic polynomial time algorithm they look like random strings and hence no partial information about m is leaked.

The ZS paper gives these informal proofs and states that they can easily be translated into formal proofs.

ZS then consider security against a CCA2 adversary. A new notion called "sole-samplable" is introduced, to describe the space induced by an encryption function. The idea is akin to the notion of a "knowledge extractor" from the definition of PA (see section 2.8). Intuitively, when a function induces a space that is sole-samplable, the only way to generate an element in that space is to start with an element of the functions pre-image. This notion is formally defined and involves a polynomial Turing machine $\mathcal{X}$ called the pre-image extractor, that with access to some relevant information, can find a pre-image with non-negligible probability.

The assumption is made that ZS-OWH and ZS-UHF induce a sole-samplable space. The proof shows that for an encryption system that is secure against a CPA adversary and induces a sole-samplable space, any CCA2 adversary can be completely simulated by a CPA adversary. But since by hypothesis the scheme is secure against a CPA adversary, it must also be secure against a CCA2 adversary.

4.2 Breaking ZS-OWH in IND-CCA2 Sense

> "Due to the involvement of $t = \mathrm{H}(m)$, the creation of the ciphertext is apparently impossible without the knowledge of x and m. ... This motivates us to introduce a notion called *sole-samplable space.*" [70, pg. 721]

If this author had to pick an assumption in the ZS paper that ultimately turned out to be incorrect, the above assumption would be an appropriate choice. As it turns out an adversary can create a new ciphertext from an existing ciphertext, if the message in the existing ciphertext is known. This credit for this attack should be shared equally between the author and Assoc. Prof. Josef Pieprzyk, as it was discovered by the author during discussions with Assoc. Prof. Pieprzyk.

To see how this is achieved consider the last part of the ciphertext, $c_2 = z \oplus (m \,||\, t) = z \oplus (m \,||\, \mathrm{H}(m))$. It just depends on the message, so if the message is known, this part of the ciphertext can be recreated. If the adversary wishes to replace the message m with another message m', this can be achieved via:

$$\begin{aligned} c_2' &= c_2 \oplus (m \,||\, \mathrm{H}(m)) \oplus (m' \,||\, \mathrm{H}(m')) \\ &= z \oplus (m \,||\, \mathrm{H}(m)) \oplus (m \,||\, \mathrm{H}(m)) \oplus (m' \,||\, \mathrm{H}(m')) \\ &= z \oplus [(m \,||\, \mathrm{H}(m)) \oplus (m \,||\, \mathrm{H}(m))] \oplus (m' \,||\, \mathrm{H}(m')) \qquad \text{(expression in [] is 0)} \\ &= z \oplus (m' \,||\, \mathrm{H}(m')) \end{aligned}$$

The new ciphertext is (c_1, c_2') and the adversary is successful in manipulating the cryptosystem.

This attack can be used by a CCA2 adversary to defeat IND and the adversary succeeds 100% of the time. In this situation the adversary does not know which of two messages, m_0 or m_1, has been encrypted, but they know one of them has been. Let the encrypted message be m_b where $b \in [0,1]$. The adversary uses the above attack by setting $m = m_0$ and $m' = m_1$ and creates a new cryptogram via:

$$\begin{aligned} c_2' &= c_2 \oplus (m_0 \,||\, \mathrm{H}(m_0)) \oplus (m_1 \,||\, \mathrm{H}(m_1)) \\ &= z \oplus (m_b \,||\, \mathrm{H}(m_b)) \oplus (m_0 \,||\, \mathrm{H}(m_0)) \oplus (m_1 \,||\, \mathrm{H}(m_1)) \end{aligned}$$

Then either (if b = 0)

$= z \oplus [(m_b \| H(m_b)) \oplus (m_0 \| H(m_0))] \oplus (m_1 \| H(m_1))$

Or (if b = 1)

$= z \oplus [(m_b \| H(m_b)) \oplus (m_1 \| H(m_1))] \oplus (m_0 \| H(m_0))$

Then

$= z \oplus (m_{\neg b} \| H(m_{\neg b}))$

Hence the adversary creates a new ciphertext (c_1, c_2'), which is a valid ciphertext for the message that was not encrypted in the challenge ciphertext. Since the adversary is a CCA2 adversary, and the new ciphertext is not the challenge ciphertext, they may query the decryption oracle with it. The decryption oracle will dutifully return the message that was not encrypted, $m_{\neg b}$, and the adversary makes their choice for b as corresponding to the message not returned by the decryption oracle.

The ZS-OWH scheme is largely of theoretical value to the cryptographic community, so while breaking the scheme does not have many practical implications, it is still of theoretical use. This break highlights the importance of adding probabilistic redundancy to the ciphertext, which can be verified on decryption. Also, as recently as EUROCRYPT 2000, a paper [60] made reference to the ZS paper with the implication being it was secure, under some assumptions. So this attack against ZS-OWH is indeed a new result.

This attack on ZS-OWH is not very complex, and as could be expected a minor change to the scheme thwarts the attack.

ZS-OWH
Preliminaries
Consider messages of length n, a one-way hash function H with output length k_0 and a PRNG G with output length $n + k_0$. Operations are modulo p and there is a generator g.
Key Generation
Private key is $x_R \in GF(p)$ and public key is $y_R = g^{x_R} \bmod p$.
Encryption
Encrypt message m 1) $x \in_R [1, p\text{-}1]$ 2) $r = y_R^x$ 3) $z = G(r)_{[1\ldots(n+k_0)]}$ 4) $t = H(m\|r)$ 5) $c_1 = g^x$ 6) $c_2 = z \oplus (m\|t)$ Ciphertext is (c_1, c_2)
Decryption
1) $r' = c_1^{x_R}$ 2) $z' = G(r')_{[1\ldots(n+k_0)]}$ 3) $w = z' \oplus c_2$ 4) $m' = w_{[1\ldots n]}$ 5) $t' = w_{[(n+1)\ldots(n+k_0)]}$ If $H(m'\|r') = t'$ then output m' else output $\varnothing$

The change incorporates some randomness into the hash calculation and thus defeats the above attack as the adversary can no longer create the concatenation of message and hash because the adversary does not know the source of randomness. This change defeats the above attack, but of course does not prove the security of the scheme.

This change was borrowed from a authenticated-encryption version of ZS-OWH by Zheng [69], however Zheng stresses that the changes made are only needed for the new scheme proposed and that the original scheme is secure.

4.3 Secure El-Gamal

The attack and the repair of the original ZS-OWH leaves a rather large question mark over its security. Securing the original ZS-OWH scheme led to a new El-Gamal variant. Great efforts were made to prove the security of this new variant using the CS proof and thus derive a scheme that was secure under some reasonable assumptions, but without using the RO model. Unfortunately, this goal was not realised, but encouragingly the proof does not heavily rely on the RO model.

4.3.1 Construction of Proof for Secure El-Gamal

The construction of the proof for Secure El-Gamal is very similar to the Cramer-Shoup proof. Knowledge of the Cramer-Shoup proof would help in understanding the construction of this proof, readers can see [15].

The proof relies on the difficulty of the Decision Diffie-Hellman Problem (DDHP), the definition of which, from Cramer-Shoup, is given below.

Definition 9 – [15, pg. 16] Let G be a group of large prime order q, and consider the following two distributions:

- the distribution **R** of random quadruples $(g_1, g_2, u_1, u_2) \in G^4$;
- the distribution **D** of quadruples $(g_1, g_2, u_1, u_2) \in G^4$, where g_1, g_2 are random, and $u_1 = g_1^r$ and $u_2 = g_2^r$ for random $r \in \mathbb{Z}_q$.

An algorithm that solves the DDHP is a statistical test that can effectively distinguish these two distributions. ▌

The construction of the proof is as follows. It is assumed an adversary that can break the cryptosystem in the IND-CCA2 sense exists, and then it is shown how this adversary can unwittingly be used to help solve what is considered a computationally unfeasible problem, in this case the DDHP. The construction of the proof can be seen in Figure 3.

The input to the proof are quadruples coming from either **D** or **R** (but not both). These go to a constructed simulator, which is responsible for, the creation of keys, simulation of an encryption oracle and simulation of a decryption oracle. The IND-CCA2 adversary receives all its information, including oracle queries, from the simulator.

The proof runs as follows. A quadruple is input, the simulator creates a valid secret key (once only), and creates the public key, which is passed to the IND-CCA2 adversary. The adversary runs its first stage A_1, and passes to the simulated encryption oracle two messages, m_0 and m_1, the simulated encryption oracle chooses a random bit $b \in [0, 1]$, encrypts m_b and passes the challenge ciphertext back to the adversary. The adversary cannot see the simulator's choice for b.

The adversary then runs its second stage, A_2, on the challenge ciphertext and outputs its guess, b', for the random bit. Both the simulator and the adversary pass b and b' respectively to a distinguisher that outputs 1 if $b = b'$ otherwise 0.

Consider the case when the input comes from **R**, the simulator is unable to create a valid ciphertext (as the relation that quadruples from **D** have, are not present in quadruples from **R**). This fact will be crucial in showing the adversary cannot succeed in guessing b with any advantage. Alternatively, when the input comes from **D**, then the simulator creates a perfectly valid ciphertext and the adversary can guess the bit b with an advantage.

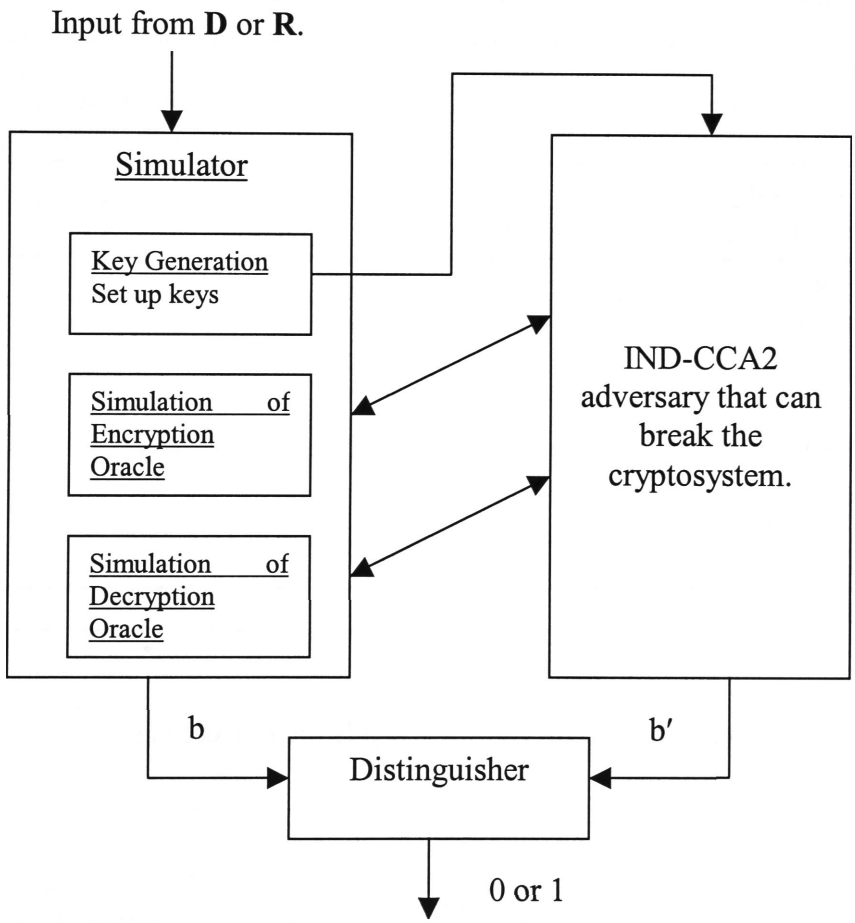


Figure 3 – Graphical representation for the construction of the Secure El-Gamal proof.

Hence by observing the distribution of 0's and 1's that are output by the distinguisher, it can be determined which distribution the quadruples are coming from. If the quadruples are coming from **R** then 1's will occur with probability 0.5 and 0's with probability 0.5. The adversary will only be correct half the time, as it has no advantage. If the quadruples come from **D** then the adversary has an advantage and

1's will occur with probability 0.5 + ε (where ε is the adversary's non-negligible advantage) and 0's with probability 0.5 - ε.

Hence, by observation of the output distribution, one has a statistical test for the DDHP.

The construction of the proof is relatively simple, however there are several properties that must hold for the proof to be valid.

- The simulator must create a valid ciphertext if the quadruple comes from **D** and an invalid ciphertext if the quadruple comes from **R**.
- When the quadruple comes from **D** the joint distribution of the adversary's view and the random bit b must be statistically indistinguishable from that in an actual attack
- When the quadruple comes from **R** the distribution of the random bit b must be (essentially) independent from the adversary's view.

4.3.2 Proof of Security for Secure El-Gamal

First the scheme is presented.

Secure El-Gamal
Preliminaries
Consider messages of length n - k_0, a random oracle H with output length k_0. Operations are modulo p where $p = 2q + 1$ (q is prime) and a generator g_1 of order q.
Key Generation
Private key is $x_R \in$ GF(p) and public key is $y_R = g^{x_R} \bmod p$.
Encryption
Encrypt message m 1) $x \in_R [1, p\text{-}1]$ 2) $r = y_R^x$ 3) $t = \mathrm{H}(m \parallel r)$ 4) $c_1 = g_1^x$ 5) $c_3 = r \cdot (m \| t)^2$ Ciphertext is (c_1, c_3)
Decryption
1) $r' = c_1^{x_R}$ 2) $w = \sqrt{\frac{c_3}{r'}}$ (choose the square root that yields the correct hash) 3) $m = w_{[1\ldots(n-k_0)]}$ 4) $t' = w_{[(n-k_0+1)\ldots n]}$ If $\mathrm{H}(m \parallel r') = t'$ then output m else output ∅

The differences between this and the original El-Gamal scheme is the addition of the hash appended to the message, and the squaring of the message and hash to convert them into a quadratic residue (this makes it an element of the quadratic residues of GF(p), the group of order q). Note that in step 2 of the decryption, if neither square root yields a correct hash then the output is ∅.

The following proof is 100% the work of the author, although rejection of previous version and verification of this version was done with significant help from Dr David Pointcheval.

Theorem 10 – *Secure El-Gamal is secure against adaptive chosen ciphertext attack in the Random Oracle model assuming that the Diffie-Hellman decision problem is hard in the group GF(p).*

The proof of security is for a scheme that is slight variant of the El-Gamal scheme described above, but the two schemes are interchangeable. The scheme used in the proof has an extra part to the ciphertext, c_2. A ciphertext from the El-Gamal scheme (above) can be transformed into one for this scheme (in the proof) by $(c_1, c_3) \rightarrow (c_1, c_2, c_3 \cdot c_2)$. The transformation back is obvious.

First the simulator is described. On input the quadruple (g_1, g_2, c_1, c_2) the simulator generates a random private key $x_R \in_R \mathrm{GF}(p)$ and outputs the public key as $y_R = g_1^{x_R} \bmod p$.

The simulator simulates the encryption oracle as follows. On input two messages m_0 and m_1 it selects a random bit $b \in [0, 1]$ and computes:

$$r = c_1^{x_R}$$

$$c_3 = (r \cdot c_2) \cdot (m_b \,\|\, \mathrm{H}(m_b \,\|\, r))^2$$

The simulated encryption oracle outputs (c_1, c_2, c_3), where c_1 and c_2 come from the input quadruple to the simulator.

The simulator simulates the decryption oracle as follows. On input (c_1, c_2, c_3) it computes:

$$r = c_1^{x_R}$$

$$w = \sqrt{(c_3 / (r \cdot c_2))}$$ (choose the square root that yields the correct hash)

$$m = w_{[1\ldots(n-k_0)]}$$

If $H(m \,\|\, r) = w_{[n-k_0+1\ldots n]}$ the simulated decryption oracle outputs m, else it outputs $\varnothing$.

The aim now is to show that when the input comes from **D** the simulator simulates the encryption and decryption oracles perfectly (probabilistically) and the advantage of the adversary is apparent at the distinguisher. Alternatively, if the input comes from **R** then the output of the simulated encryption oracle will not be a valid ciphertext in the sense that $\log_{g_1} c_1 \neq \log_{g_2} c_2$.

It is also important to note that since the DDHP is hard for the adversary they cannot even find out any partial information about the secret key that could be used to determine b.

The theorem follows from the following two lemmas.

Lemma 1 – *When the simulator's input comes from **D**, the joint distribution of the adversary's view and the hidden bit b is statistically indistinguishable from that in the actual attack.*

In this case it is clear the output of the simulated encryption oracle has the right form as $c_1^{x_R} c_2 = \left(g_1^x\right)^{x_R} g_2^x = \left(g_1^{x_R}\right)^x g_2^x = y_R^x g_2^x$ which is equivalent to the output of the actual encryption oracle. Similarly, the simulated decryption oracle will accept all valid ciphertexts.

It remains to be shown that all invalid ciphertexts are rejected with overwhelming probability. If an invalid ciphertext (in the sense that $\log_{g_1} c_1 \neq \log_{g_2} c_2$) is presented as a query to the decryption oracle it will be rejected as the resulting r will not be correct for recovering m from c_3. More importantly the invalid ciphertext will not pass the check involving the random oracle (H). By using a random oracle it is ensured that the hash is completely non-malleable and no partial information is leaked.

Lemma 2 – *When the simulator's input comes from* ***R****, the distribution of the hidden bit is (essentially) independent from the adversary's view.*

First it will be shown that no partial information about b is leaked from just the challenge ciphertext, this essentially is showing IND-CPA security. Then it will be shown that there is only a negligible chance that the simulated decryption oracle gives the adversary any information about b. Since an IND-CCA2 adversary that cannot gain any information from a decryption oracle is equivalent to an IND-CPA adversary, the lemma is proven.

It has been shown that assuming DDHP the El-Gamal cryptosystem is secure in the sense of IND-CPA [11, 64]. To show the IND-CPA security of this scheme it will be shown how to convert an El-Gamal challenge ciphertext into one for this scheme. First a second generator needs to be created, if p is of the form $p = 2q + 1$, then there are $q - 1$ generators. Hence by considering powers of g_1 a second generator of the form $g_2 = g_1^w$ can be found in polynomial time, with w known. So g_2^x can be calculated as $(g_1^x)^w$. So an El-Gamal challenge ciphertext can be transformed into a Secure El-Gamal challenge ciphertext as $(g_1^x, y_R^x.m_b) \rightarrow (g_1^x, g_2^x, (y_R^x\ g_2^x).m_b)$. It should be noted that the message is a different size to a message in an actual Secure El-Gamal challenge ciphertext. However this is not an issue, if p is an n bit prime, and the hash function outputs 128 bits, then the chances that two messages chosen at random do not differ in the first $n - 128$ bits is $1/2^{n-128}$, which is negligible for suitable large n. The absence of the appended hash is irrelevant since the use of a random oracle ensures no information about m is leaked to an IND-CPA adversary. Also, without access to a decryption oracle there is no need for a correct hash value to be present in the ciphertext.

The simulated decryption oracle still needs to reject all invalid ciphertexts, otherwise relevant information will be leaked. A valid ciphertext is (c_1, c_2, c_3), an invalid one is (c_1', c_2', c_3'). There are two cases to consider.

1) $(c_3) = (c_3')$. If this happens with non-negligible probability then the random oracle must not be one way since c_1' and c_2' will create a different r (as they are different from c_1 and c_2) and this will cause decryption to a different message and hash. If the hash check passes then the hash has been created without knowledge of the message.

2) $(c_1, c_2) = (c_1', c_2')$. With $c_3 \neq c_3'$, then the adversary has to replace the message and hash in c_3 to create c_3'. They can't just replace the message as if the hash check passes then a collision has been found. They can't replace the hash, or the message and hash, as without complete knowledge of r the correct hash cannot be calculated, and if it could then a collision could be found.

Using a random oracle means that one-wayness and collision-freeness cannot be defeated, in fact no partial information is leaked about the pre-image of the hash. Thus, the simulated decryption oracle will reject all invalid ciphertexts, except with negligible probability. ▌

Hence if the DDHP is a computationally unfeasible problem then an IND-CCA2 attacker for Secure El-Gamal cannot exist.

4.3.3 Comparison between CS and Secure El-Gamal

It is immediately obvious that the CS scheme is more secure than Secure El-Gamal (remember all security is based on assumption, so using a phrase like 'more secure' just refers to the confidence we have in the assumptions that allow for security). This is because the CS scheme relies only on standard assumptions, and the Secure El-Gamal scheme relies on the random oracle model, albeit in a minimal way.

So what are the other differences between the two schemes? Well, the CS proof is certainly stronger as it requires weaker assumptions, but the Secure El-Gamal scheme is far more efficient. There is a trade-off here between the strength of proof, and efficiency. In many practical applications of public key cryptography the Secure El-Gamal scheme would be the better choice due to its efficiency and good security. However it would not be the best choice, there exist schemes, like OAE [4], that are even more efficient.

However, circumstances could exist where an application used the original El-Gamal (or a variant) and it was decided to change to a more secure scheme. In this case, it would be more efficient to change to Secure El-Gamal as there would be less overhead than a change to OAE.

4.4 Elliptic Curve ZS

Most cryptosystems that are variants of El-Gamal can be transformed into systems based on the ECDL problem. Here, for the first time, the ZS schemes that have been examined in this chapter will be transformed into their elliptic curve equivalents.

Throughout each cryptosystem the two functions *compress()* and *expand()* are used to represent EC point compression and expansion. This is the process of sending one coordinate and 1 bit of the other coordinate, instead of sending both coordinates of an EC point. For a definition of *compress()* see [29, A.9.6] and for *expand()* see [29, A.12.9]. Also, for the sake of simplicity the dependence on the PRNG, G, has been removed from these cryptosystems.

The schemes here will be for an EC defined over a prime field; they can be easily modified for an EC over a binary extension field. Also, it is assumed that the choice of the coefficients a and b are such that the resulting EC is not supersingular or anomalous and that the number of points on the curve has a large prime factor (this ensures security against algorithms that solve the DLP in arbitrary groups).

4.4.1 ECZS-OWH

Here the repaired ZS-OWH is transformed to its EC equivalent.

ECZS-OWH
Preliminaries
For a large prime p, choose appropriate a and b to define an EC. Consider messages of length $n = \|p\| + 1 - k$, and a one-way hash function H with output length k. Find a point on the EC, G, that generates the large prime sub-group.
Key Generation
Randomly choose a secret key $x_R \in \mathrm{GF}(p)$ and calculate public key $Y_R = x_R G$.
Encryption
To encrypt a message m 1) $x \in_R [1, p\text{-}1]$ 2) $z = compress(xY_R)$ 3) $t = \mathrm{H}(m \,\|\, z)$ 4) $c_1 = compress(xG)$ 5) $c_2 = z \oplus (m \,\|\, t)$ Ciphertext is (c_1, c_2)
Decryption
1) $C = expand(c_1)$ 2) $z' = compress(x_R \cdot C)$ 3) $w = z' \oplus c_2$ 4) $m = w_{[1\ldots n]}$ 5) $t' = w_{[n+1\ldots \|p\|]}$ If $\mathrm{H}(m \,\|\, z') = t'$ then output m else output $\varnothing$.

4.4.2 ECZS-UHF

Here ZS-UHF is transformed to its EC equivalent.

ECZS-UHF
Preliminaries
For a large prime p, choose appropriate a and b to define an EC. Consider messages of length $n = \|p\| + 1 - k$, and a (strong) universal hash function H_s that is indexed by a string of length k. Find a point on the EC, G, that generates the large prime sub-group.
Key Generation
Randomly choose a secret key $x_R \in \mathrm{GF}(p)$ and calculate public key $Y_R = x_R G$.
Encryption
To encrypt a message m 1) $x \in_R [1, p\text{-}1]$. 2) $r = compress(xY_R)$ 3) $z = r_{[1\ldots \|p\|+1-k]}$

4) $s = r_{[|p|+2-k \ldots |p|+1]}$
5) $c_1 = compress(xG)$
6) $c_2 = H_s(m)$
7) $c_3 = z \oplus m$
Ciphertext is (c_1, c_2, c_3)

Decryption

1) $C = expand(c_1)$
2) $r' = compress(x_R \cdot C)$
3) $z' = r_{[1 \ldots p+1-k]}$
4) $s' = r_{[p+2-k \ldots |p|+1]}$
5) $m = z' \oplus c_3$
If $H_s(m) = c_2$ then output m else output $\varnothing$.

4.5 Implementation

Before the attack on the original ZS-OWH was discovered, the original ZS-OWH and its elliptic curve equivalent were implemented solely by the author. The schemes were implemented in the Java programming language using the Java Development Kit (JDK) version 1.3 beta [66] and a third party implementation (due to USA export laws) of the Java Cryptographic Environment (JCE) by the Australian Business Access Pty Ltd (ABA) [65]. Java was used as the programming language as it has extensive support for multiple precision arithmetic.

The JCE allows for the simplest creation of cryptographic algorithms and provides many standard algorithms too, like DES, SHA etc. The JCE lets programmers create an entire cryptographic suite, called a Provider. From the user's point of view, if they want to use an algorithm, they see which Providers implement their desired algorithm and choose the implementation they want. So a Provider was created that implemented the original ZS-OWH and ECZS-OWH.

The implementation of cryptographic algorithms is made simpler by the JCE. It lays out all the methods (Java's name for functions) that need to be written in order for the algorithm to work, and the programmer just writes the code for each method, appropriate for the algorithm being implemented. Of course anyone who has ever implemented cryptographic algorithms knows this tasks is not as easy as it sounds. The main advantage of using the JCE is that the user is provided with a consistent interface, the interface is the same regardless of Provider and algorithm type (cryptosystem, hash function, signature).

Implementing ECZS-OWH was more difficult than ZS-OWH as Java has no support for elliptic curves and so all the fundamentals of elliptic curves over binary extension fields needed to be implemented as well.

The code of this implementation can be found on a diskette in a pocket at the back of this thesis.

CHAPTER 5 SIGNCRYPTION

In 1997 Zheng [68] introduced a new notion that he termed 'signcryption' which combined digital signatures and encryption. Up until then the notions of confidentiality and authentication had been considered separately by most, or at best had been combined through simple concatenation. All the work in this chapter, not specifically referenced, is original work by the author.

5.1 Background

5.1.1 Principle

The principle behind signcryption (from Zheng) can be basically summed up in one inequality.

$$Cost(Signcryption) < Cost(Signature) + Cost(Encryption)$$

Signcryption is the process of achieving authentication and confidentiality at a 'cost' less than preforming authentication and confidentiality independently. Exactly how 'cost' is evaluated is not specified, as it is a relative term, depending on the application. Zheng used computational cost and communication cost. Computational cost is a measure of the number of operations it takes to implement a scheme, practically this means counting the number of dominant operations such as exponentiation or inversion. Communication cost is a measure of the amount of bits a scheme needs to send per ciphertext. Both computational and communication cost for a signcryption scheme can be compared with a signature-then-encryption scheme to judge the effectiveness of the signcryption scheme.

5.1.2 Definition

For the first time a signcryption scheme is given a formal definition. A signcryption scheme can be defined in much the same way as an encryption scheme.

A signcryption scheme can be formally defined via a triple of algorithms $\Pi = (\mathcal{K}, \mathcal{S}, \mathcal{U})$.

- $\mathcal{K}$, the *Key Generation Algorithm*. A probabilistic algorithm that takes a security parameter $k \in \mathbb{N}$ and returns a pair (x, y) of matching secret and public keys.
- $\mathcal{S}$, the *Signcryption Algorithm*. A probabilistic algorithm that takes the public key of the receiver y_R, the private key of the sender x_S and a message $m \in \{0, 1\}^*$ to produce a signcryptogram c.
- $\mathcal{U}$, the *Unsigncryption Algorithm*. A deterministic algorithm that takes the private key of the receiver x_R, the public key of the sender y_S and the signcryptogram c, to return the message m or a special symbol $\varnothing$ to indicate the signcryptogram was invalid.

We require that for all (x, y) which can be output by $\mathcal{K}(1^k)$, for all $m \in \{0, 1\}^*$, and for all c that can be output by $\mathsf{S}_{y_R, x_S}(m)$, we have that $\mathsf{U}_{x_R, y_S}(c) = m$. We also require that $\mathcal{K}$, $\mathcal{S}$ and $\mathcal{U}$ can be computed in polynomial time.

Part of the definition of signcryption could include the notion of 'cost' but it is a very relative term, not only as to how 'cost' is evaluated but also the encryption and signature schemes the 'cost' of the signcryption scheme is compared too. The goal of this thesis in formally defining signcryption is to allow for meaningful statements about the security of a signcryption scheme, not formally defining 'cost' will have no effect on evaluating security.

Zheng uses the phrase 'signcrypted text' to describe the output of the signcryption algorithm, this thesis uses the phrase 'signcryptogram', for no other reason than it is more convenient.

Formally defining a signcryption scheme becomes very useful when defining notions of security, as this in turn allows the security of cryptosystems based on this new notion (signcryption), to be formally expressed.

Although Zheng did not formally define signcryption, he did give some properties that he argued all signcryption schemes must fulfil.

1. *Unique unsigncryptability* – The unsigncryption algorithm unambiguously recovers the message m from the signcryptogram c.
2. *Security* – The signcryption scheme simultaneously fulfils the properties of a secure encryption scheme and a secure signature scheme.
3. *Efficiency* – The computational and communication cost of the signcryption scheme is smaller than that of the best currently known signature-then-encryption schemes with comparable parameters.

Zheng points out that conditions 2 and 3 justify the introduction of signcryption as a new concept as it clearly shows that signcryption is not the same as signature-then-encryption.

5.1.3 Signcryption versus Authenticated Encryption

Signcryption and authenticated encryption are two very similar ideas, it holds that signcryption achieves authenticated encryption whereas authenticated encryption does

not necessarily achieve signcryption. Hence all the definitions and results that are true for signcryption, are also true for authenticated encryption.

In the previous section, the properties of a signcryption scheme included efficiency and it is tempting to consider this the only difference between signcryption and authenticated encryption. However, authenticated encryption can always and easily be achieved by concatenation of independent and secure signature and encryption schemes (or vice-versa). It is important to note that it is *not* trivial to derive a signcryption scheme from secure signature and encryption schemes, indeed it is all too easy to destroy both authentication and confidentiality.

So although Zheng justifies signcryption as worthy of being a new concept in cryptography, to his argument should be added, that never before has a cryptographic notion focused on the interplay between authentication and confidentiality. Clever use of this interplay will allow schemes to be derived where authentication and confidentiality complement each other.

5.2 Notions of Security

The notions of security that exist for encryption allow for meaningful discussion about the security of encryption schemes, similarly for signature schemes. So it is only natural that for a combination of encryption and signature schemes there needs to be defined similar notions of security, so the security of signcryption schemes can be analysed. The notions presented here are extensions of the notions of security for encryption and signature schemes, but the extension is the original work of the author.

An extensive hierarchy of security levels is not given here. The reason is that in a practical signcryption scheme, the security should achieve a certain goal against a certain attack, if it does not then the signcryption scheme should not be used. Presented here are the significant goals and attacks.

5.2.1 Attacks against Confidentiality

The formal notation used, and the model of the adversary in this section is the same as those used in sections 2.7.1 and 2.7.2 respectively.

The weakest attack against a signcryption scheme is a *Known Signcryptogram Attack* (KSA). In this attack an adversary is not given access to either the sender or receiver's signcryption or unsigncryption oracles. The adversary only has access to a polynomially bounded number of signcryptograms. This models the situation when the adversary can only eavesdrop on communications.

In a *Chosen Plaintext Attack* (CPA) the adversary only has access to the signcryption oracle of the sender, this is the attack that is most available to an adversary (by making themselves the sender). The adversary can create a polynomially bounded number of signcryptograms thus obtaining a set of plaintext-signcryptogram pairs. The adversary then attempts to ascertain some information from these pairs.

The most powerful attack an adversary can have is a *Chosen Signcryptogram Attack* (CSA). The adversary has access to a signcryption oracle of the sender and an unsigncryption oracle of the receiver and may query both a polynomially bounded

number of times. However, the adversary may not query the unsigncryption oracle with the challenge signcryptogram.

CSA is a very strong attack, the adversary can not only create signcryptograms, whether valid or invalid, but can query the unsigncryption algorithm of the receiver (use it as an oracle), with the exception of the challenge signcryptogram. This is an adaptive attack since the adversary can query the unsigncryption oracle with signcryptograms related to the challenge signcryptogram.

5.2.2 Indistinguishability of Signcryptions

As with IND, indistinguishability of signcryptions (INDS) captures the notion that an adversary can gain no information about the message of a signcryptogram other than its length (but may be able to authenticate). This goal is achieved by giving the adversary one signcryptogram and asking the adversary to choose which of two plaintexts were signcrypted.

Exactly the same notation as section 2.7.1 is used.

Definition 10 [*INDS-CPA, INDS-CSA*] Let $\Pi = (\mathcal{K}, \mathcal{S}, \mathcal{U})$ be a signcryption scheme and let $A = (A_1, A_2)$ be an adversary. For atk $\in$ {cpa, csa} and $k \in \mathbb{N}$ let

$$\mathrm{Adv}^{\text{inds-atk}}_{A,\Pi}(k) = 2 \cdot \Pr\left[(pk_S, sk_S) \leftarrow \mathsf{K}(1^k); (pk_R, sk_R) \leftarrow \mathsf{K}(1^k); \right.$$
$$(x_0, x_1, s) \leftarrow A_1^{\mathcal{O}_1}(pk_R, sk_S); b \leftarrow \{0,1\};$$
$$\left. y \leftarrow \mathsf{S}_{pk_R, sk_S}(x_b) : A_2^{\mathcal{O}_2}(x_0, x_1, s, y) = b \right] - 1$$

where

If atk = cpa then $\mathcal{O}_1(\cdot) = \varepsilon$	and $\mathcal{O}_2(\cdot) = \varepsilon$
If atk = csa then $\mathcal{O}_1(\cdot) = \mathsf{U}_{sk_R, pk_S}(\cdot)$	and $\mathcal{O}_2(\cdot) = \mathsf{U}_{sk_R, pk_S}(\cdot)$

It is insisted that A_1 outputs x_0, x_1 with $|x_0| = |x_1|$. In the case of CSA, it also insisted that A_2 does not ask its oracle to unsigncrypt y. We say that Π is secure in the sense of INDS-ATK if A being polynomial-time implies that $\mathrm{Adv}^{\text{inds-atk}}_{A,\Pi}(\cdot)$ is negligible. ▌

5.2.3 Not Existentially Forgeable

Since signcryption covers both confidentiality and authentication, having defined how to evaluate the confidentiality of a signcryption scheme, the same must be done for authentication.

Analogous to confidentiality, two types of attackers can be defined for authentication.

A *Known Message Attack* (KMA) is by an adversary that has access to a list of signcryptograms and their corresponding messages. Since this is supposed to correspond to the normal signature KMA, an adversary against a signcryption scheme needs access to the receiver's unsigncryption oracle. This is because for a normal signature scheme verification is public, but this is not necessarily true for signcryption. A signcryption scheme may or may not have its authentication publicly verifiable, and for consistency an adversary should either always be able to verify or

never be able to. To force an adversary to never be able to verify would mean changing the definition of signcryption so public verification is not allowed.

A *Chosen Signcryptogram Attack* (CSA) is the same as CSA defined for confidentiality. The adversary has access to both the sender's signcryption oracle and the receiver's unsigncryption oracle.

The strongest goal for any signature scheme is for it to be *Not Existentially Forgeable* (NEF), and this is the same for a signcryption scheme. NEF means an adversary is unable create a valid signcryptogram that contains a forgery of *any* message (whether known or not known by the adversary). NEF for a signcryption scheme means there is no message that can be forged (although messages can always be forged with a negligible probability).

5.2.4 Problems with combining Confidentiality and Authentication

The notions of security for signcryption are more difficult to appreciate than for encryption or signature schemes. An important issue is the oracles an adversary has access too. If the adversary is trying to forge a signature, then access to an unsigncryption oracle is of less use than a signcryption oracle, but the reverse is true for an adversary trying to break confidentiality. So a weak adversary against confidentiality is similar to a strong one against authentication, and vice versa.

This means that a signcryption scheme that has weak confidentiality security, like INDS-CPA, but strong authentication security, like NEF-CSA, does not really make sense. An NEF-CSA adversary has access to both oracles, but this means the attacker is also an INDS-CSA attacker against confidentiality. The potential is for an adversary against confidentiality to be used as one against authentication, or vice versa.

This inevitably leads to the need for a signcryption scheme to always have its confidentiality and authentication secure against the most powerful adversary, one with access to both oracles.

5.3 Plaintext Awareness

The notion of PA for signcryption is defined for the first time here and is exactly analogous to the notion of PA for confidentiality. PA for signcryption means that an adversary against a signcryption scheme should not be able to create a signcryptogram for which they do not know the plaintext. It is arguable that this notion is even more important for signcryption schemes as it obviously related to an adversary's ability to forge.

See section 2.8 for the notation and definitions of hH and C.

Definition 11 - (*PA*) Let $\Pi = (\mathcal{K}, \mathcal{S}, \mathcal{U})$ be a signcryption scheme, let B be an adversary, and let K be an algorithm (the "knowledge extractor"). For any $k \in \mathbb{N}$ let

$$\mathrm{Succ}^{pa}_{K,B,\Pi}(k) \stackrel{def}{=}$$

$$\Pr[H \leftarrow \mathrm{Hash}; (pk_S, sk_S) \leftarrow \mathsf{K}(1^k); (pk_R, sk_R) \leftarrow \mathsf{K}(1^k);$$

$$(hH, C, y) \leftarrow \mathrm{run}\, B^{H, \mathsf{S}^H_{pk_R, sk_S}}(pk_R, sk_S) : K(hH, C, y, pk_R, sk_S) = \mathsf{D}^H_{sk_R, pk_S}(y)]$$

We insist $y \notin C$; that is, B never outputs a string y which coincides with the value returned from some $\mathsf{S}^H_{pk_R, sk_S}$-query. We say that K is a $\lambda(k)$-extractor if K has running time polynomial in the length of its inputs and for every adversary B, $\mathrm{Succ}^{pa}_{K,B,\Pi}(k) \geq \lambda(k)$. We say that Π is secure in the sense of PA if Π is secure in the sense of INDS-CPA and there exists a $\lambda(k)$-extractor K where 1 - $\lambda(k)$ is negligible. ▌

As with the definition of PA for confidentiality, the definition for signcryption is restricted to the random oracle model.

PA is not a notion of security, it is rather a property of a cryptosystem that implies a notion of security. For confidentiality, a PA scheme implies the scheme is secure against an IND-CCA2 adversary [3]. It will be assumed that the same result holds for signcryption, PA $\Rightarrow$ INDS-CSA. This is a reasonable assumption as essentially the only difference is encryption and decryption oracles are replaced with signcryption and unsigncryption oracles and the addition of the sender's secret key. These changes are unlikely to invalidate the proof of the theorem.

CHAPTER 6 NEW SIGNCRYPTION SCHEMES

Presented in this chapter and for the first time are three new signcryption schemes all based on provably secure schemes (although all only so in the RO model). The three underlying schemes are Secure El-Gamal, Pointcheval [52] and OAE [4]. The signcryption schemes presented here will differ from the scheme presented by Zheng [68] as they will not use a private key (symmetric) cipher. The reason for starting from provably secure schemes is the hope that the changes made (to make them signcryption schemes) will not affect their confidentiality, in which case an argument about their authentication is all that remains.

For each new scheme its signcryption and unsigncryption algorithms will be given, along with other relevant information, then its security will be evaluated in terms of both confidentiality and authentication. Finally both computational and communication cost will be given, which in turn is a justification for each to be considered a signcryption scheme

6.1 Secure El-Gamal Signcryption

Secure El-Gamal Signcryption is derived from Secure El-Gamal and an El-Gamal style signature.

6.1.1 Scheme

Secure El-Gamal Signcryption
Preliminaries
Consider messages of length n, and random oracles H and G with output length k and output length $2n$, respectively. Operations are modulo p with a generator g_1.
Key Generation
The private key of receiver is $x_R \in$ GF(p) and public key is $y_R = g_1^{x_R} \bmod p$. The sender has private key $x_S \in$ GF(p) and public key $y_S = g_2^{x_S} \bmod p$.
Signcryption
Signcrypt message m 1) Randomly choose $x \in_R [1, p\text{-}1]$

2) $r = y_R^x$

3) $z = G(r)_{[1...2n]}$

4) $c_1 = g_1^{\,x}$

5) $t = \dfrac{x}{H(m) + x_S c_1} \bmod p - 1$

6) $c_3 = z \oplus (m \,\|\, t)$

Signcryptogram is (c_1, c_3)

Unsigncryption

1) $r' = c_1^{x_R}$

2) $z' = G(r')_{[1...2n]}$

3) $w = z' \oplus c_3$

4) $m = w_{[1...n]}$

5) $t' = w_{[(n+1)..2n]}$

If $(g_1^{\,t'H(m)} y_S^{\,t'c_1}) = c_1$ then output m else output $\varnothing$

The basic change between this scheme and Secure El-Gamal is the variable t has changed from a hash of the message to a signature of the message, signed with the sender's private key. In the unsigncryption algorithm the signature of the message is checked and if it is verified the message is accepted. Importantly though the output of a random oracle is used to hide all partial information about the message and signature, meaning this scheme relies heavily on the output of the random oracle being random.

This scheme can be made more efficient by calculating the signature in a large prime sup-group of $p - 1$. This would give minimal expansion in the number of bits sent compared to the original Secure El-Gamal.

6.1.2 Security

Confidentiality follows from the security of Secure El-Gamal.

Theorem 11 – *Secure El-Gamal Signcryption is secure against a chosen signcryptogram attack in the random oracle model assuming that the Diffie-Hellman decision problem is hard in the group GF(p).*

The proof is the same as from Theorem 10 with the following changes. The simulator is changed to incorporate the sender's keys. Instead of relying on the security of original El-Gamal to show no partial information is leaked, the output of a random oracle is used to mask all information. The difficulty of guessing the hash of the message concatenated with random information is replaced with the difficulty of guessing the sender's private key. Since H is a random oracle, there is no way that two messages have the same signature, so the signature can't be recreated without x_S. Also, an adversary without x_S cannot even create a new signature. If we assume that the signature is NEF-CSA then no argument needs to be made. ▌

Authentication has essentially already been shown. The form of the signature is almost identical to DSS, which is considered safe, but is even more secure as the signature is not in the adversary's view.

6.1.3 Cost

Computational cost is measured by counting the number of significant operations, these include exponentiation and inversion. For Secure El-Gamal signcryption this comes to 6 operations.

For communication overhead the most efficient implementation of Secure El-Gamal signcryption will be considered. This is when calculations are done in a large prime sub-group of $p - 1$, with order q. The total number of bits that need to be sent, for messages of size $|p|$, is $2|q| + |p|$.

6.2 Pointcheval Signcryption

Pointcheval Signcryption (PS) is derived from Pointcheval's modified El-Gamal [52] and one of three El-Gamal style signature schemes.

6.2.1 Scheme

Pointcheval Signcryption
Preliminaries
A partially trapdoor one-way function f, and another function that partially inverts it g. Work over group GF(p), with p - 1 having a large prime factor q, and a generator g of order q. A hash function, H, and PRNG, G.
Key Generation
The private key of receiver is $x_R \in$ GF(p) and public key is $y_R = g^{x_R} \bmod p$. The sender has private key $x_S \in$ GF(p) and public key $y_S = g^{x_S} \bmod p$.
Signcryption
Encrypt a message m 1) Randomly choose $r \in_R [1 \ldots q]$ 2) (a) Using modified El-Gamal signature Solve $H(m, g^r) = x_S g^r + rs \bmod q - 1$ for s (b) Using DSS Calculate $s = r^{-1}(m + x_S g^r) \bmod q - 1$ (c) Using modified Schnorr signature scheme Calculate $s = r - x_S H(m, g^r) \bmod q - 1$ 3) $a = f(g^r, H(m\|s)$ 4) $b = (m\|s) \oplus G(g^r)$ Signcryptogram is (a, b)
Unsigncryption
1) $u = g(a)$ 2) $M = b \oplus G(u)$ 3) $s' = M_{[\|m\| \ldots q-1]}$ 4) If $a = f(u, H(M))$ and (a) For modified El-Gamal Check $g^{H(m',u)} \equiv y_S^u u^{s'} \bmod p$ (b) For DSS

Calculate $\alpha = ms'^{-1} \bmod q, \beta = us'^{-1} \bmod q$ then check

$$u \equiv g^{\alpha} y_S^{\beta} \bmod p \bmod q$$

(c) For modified Schnorr

Check $u \equiv g^{s'} y_S^{H(m,u)} \bmod p$

Then

$m = M_{[1\ldots|m|]}$

else output $\varnothing$

This is a generic signcryption scheme. The only difference between this scheme and Pointcheval's original scheme is the variable s, which was random in the original, but is the signature in this scheme.

6.2.2 Security

The confidentiality of (PS) can be shown using a very similar proof as that used by Pointcheval for his original encryption scheme. The original proof first showed the scheme to be secure in the sense of IND-CPA, then defined a knowledge extractor for the scheme and used the definition of PA to show the scheme was IND-CCA2.

Theorem 12 – *PS is secure in the sense of INDS-CPA.*

Exactly the same proof as [52] will be used with a few minor changes. The original proof of security in the IND-CPA sense assumes that an IND-CPA attacker exists and then by observing its oracle queries the one-way function, f, can be inverted. The same is done here, except the security now needs to be in the INDS-CPA sense. This means the secret key of the sender is incorporated. Another minor change is r changes to g^r; this has no effect on the proof.

The only other change is s is no longer random, but a signature. However, s being random is not essential for the proof, but it actually is random from the adversary's point of view as they do not have access to an unsigncryption oracle and so couldn't distinguish a signature from a random string. ▌

Theorem 13 – *PS is secure in the sense of INDS-CSA.*

All that needs to be done is to define a knowledge extractor, K, for the scheme and then use the definition of PA. The first part of the knowledge extractor comes directly from Pointcheval's original proof:

> "The simulator S considers all the queries asked to G and H, (r, G_r) and (q, H_q), and checks if for some pair (r, q), both equalities $a = f(r, H_q)$ and $b = q \oplus G_r$ hold." [52, pg. 138]

Then K recovers the message m and signature s from q, and verifies s using (r, m). With a knowledge extractor defined and the scheme being INDS-CPA by Theorem 12, then the scheme fulfils PA, and so is INDS-CSA (as it was assumed PA $\Rightarrow$ INDS-CPA). ▌

Authentication is easy to show since the security of the signature schemes used is widely accepted (except for modified Schnorr, so this would need some security

justifications). So there is a negligible chance that an adversary could create a forged signature, and since the signature is hidden in the signcryptogram an adversary could not change a valid signature into a forged signature.

6.2.3 Cost

The computational cost of PS will be measured using the El-Gamal instantiation that Pointcheval derived in [52] and using the modified El-Gamal signature. This leads to a total of 8 operations (if modified Schnorr was used this would be 7, but there is a question mark over its security).

The communication cost, for a message of length $|p|$, is $3|q| + |p|$.

6.3 Optimal Asymmetric Signcryption

Optimal Asymmetric Signcryption (OAS) is derived from OAE [4] and a standard RSA signature scheme. Since OAS is derived from OAE it exists in a generic form, however the scheme presented here is instantiated with RSA.

6.3.1 Scheme

OAS
Preliminaries
Generate two strong primes p and q and calculate $N = pq$. Messages are of length n, and a random number of length k_0 is needed, so that $\|N\| = n + k_0 + k_1$. Also need two random oracles, H taking strings of length $n + k_1$ to strings of length k_0, and G taking strings of length k_0 to strings of length $n + k_1$.
Key Generation
Generate a random number x_R, such that $\gcd(x_R, N) = 1$, as the public key of the receiver and solve for y_R in $x_R y_R = 1 \bmod \phi(N)$ for the private key. Similarly for the sender, generate y_S, and solve $x_S y_S = 1 \bmod \phi(N)$ for x_S.
Signcryption
To signcrypt a message m 1) Choose random r of length k_0 2) $m' = m \,\|\| \, 0^{k_1}$ 3) $s = m' \oplus \mathrm{G}(r)$ 4) $t = r \oplus \mathrm{H}(s)$ 5) $w = s \,\|\, t$ 6) $y = \left(w^{y_R}\right)^{x_S}$ Signcryptogram is y
Unsigncryption
1) $w = \left(y^{y_S}\right)^{x_R}$ 2) $s = w_{1..n}$ 3) $t = w_{n+1\ldots n+k_0}$ 4) $r = t \oplus \mathrm{H}(s)$ 5) $m' = s \oplus \mathrm{G}(r)$ 6) If $m'_{[n+1\ldots n+k_1]} \equiv 0^{k_1}$ then $m = m'_{[1\ldots n]}$ else output $\varnothing$

Note, the notation '0^{k_1}' refers to a string of k_1 zeroes.

OAS is just OAE with the ciphertext signed using a standard RSA signature, so does it really qualify as a signcryption scheme? The argument for it being a signcryption scheme is that it is more efficient (in communication cost) than if OAE and the standard RSA signature were done independently. More importantly, it will be shown to be more secure, which means if the 'cost' is evaluated in terms of security then there is also a saving.

6.3.2 Security

The confidentiality of the scheme is already guaranteed by the underlying encryption, all that needs to be shown is that authentication is secure.

Theorem 14 – *OAS is secure against an INDS-CSA adversary.*

The reader is referred to the proof of OAE in [4]. It is trivial to show that if an IND-CSA adversary can break OAS then they could be used by an IND-CCA2 adversary to break OAE. This is true because the only difference between the schemes is the signing of the ciphertext; hence an IND-CCA2 adversary can change their challenge ciphertext into an INDS-CSA adversary's challenge ciphertext by simply signing it.

∎

The argument for the security of authentication of this scheme is based on the underlying OAE being PA. The authentication is secure in the sense of NEF-KMA as an adversary cannot create a signcryptogram for which they do not know the plaintext (definition of PA), so cannot create a signcryptogram for a plaintext other than one already in their list of message-signcryptogram pairs.

There are well known homomorphic attacks against RSA signatures for an adaptive adversary, but these will not work against this scheme. Hence this scheme is more secure than if OAE and a standard RSA signature were used independently. An adversary only has a negligible chance of forging as there is only a negligible chance that the new ciphertext that is signed (has a signature on it forged) will be valid. This is because OAE is non-malleable (equivalent to IND), so an adversary can't forge a signcryptogram for a message that is meaningfully related to any message they have signcryptograms for. Also, since OAE is PA they cannot forge a signcryptogram for a message they do not know. However, there is a slim chance that an adversary could forge a signature on a message that is known to them, but is unrelated to any message-signcryption pairs the adversary knows or creates.

6.3.3 Cost

The reason for using the phrase 'Optimal' in OAE was that encryption and decryption only needed 1 exponentiation each. For OAS signcryption and unsigncryption only need 2 exponentiations each, which is arguable optimal for a scheme incorporating confidentiality and authentication. This makes the computational cost 4 operations.

- Although OAS sends out $|N|$ bits, the message is shorter than this, so it is difficult to calculate the cost until real parameters are used.

6.4 Cost Comparison

Comparing the cost between the new schemes presented here and with possible signature-then-encryption schemes is difficult due to the differences in parameters. An important example of this is insisting the plaintext for two schemes be the same size, as two schemes with different sized plaintexts shouldn't be directly compared. Another important property is security assumptions, it isn't much use comparing a provably secure scheme to one with questionable security, since it is very likely the scheme with questionable security will have significantly less cost.

Presented here is a comparison between these new signcryption schemes and authenticated encryption schemes based on Pointcheval's El-Gamal and OAE. Pointchval's El-Gamal concatenated with Pointchval-Stern modified El-Gamal signature (both provably secure against IND-CCA2 and NEF-CMA respectively) has a computational cost of 8 operations and a communication cost of $5|q| + |p|$. OAE with standard RSA signature (OAE is provably secure the signature is not) has a computational cost of 4 operations and a communication cost dependent on parameters.

Recall computational cost refers to the number of dominant operations that a scheme needs to preform and communication cost refers to the number of bits a scheme needs to send. Savings in cost are measured via.

$$\frac{\textit{Cost(Authenticated Encryption)} - \textit{Cost(Signcryption)}}{\textit{Cost(Authenticated Encryption)}}$$

For example, a saving of 10% in computational cost would mean he signcryption scheme does 10% less dominant operations than the authenticated encryption scheme.

Table 2 presents the results. The different plaintext sizes have been taken into account, for an El-Gamal type schemes the message size is equal to $|p|$ and for the RSA type schemes, for $|p| = |N| = 1024$, $|m| = 768$ and for $|p| = |N| = 4096$, $|m| = 3072$. Hence the message size for the RSA type schemes are always ¾ that of the message size for the El-Gamal type schemes. The parameters were chosen to reflect appropriate sizes for use today and in 5-10 years time, depending on the progress of solving the DLP or factoring.

It should be noted that the OAE with standard RSA signature is rather inefficient since the message being signed is smaller than the modulus being used, but this is a practical issue since otherwise the moduli of the sender and receiver would be of vastly different size. Also, counting an RSA exponentiation as 1 operation is questionable, as there exist methods for speeding up this calculation, however, for the sake of simplicity it is left this way.

Table 2 shows that the new signcryption schemes presented here offer useful savings in the short term. However, the savings in communication cost for the El-Gamal type schemes decrease as more secure parameters are used. This is because the savings are associated with the sub-group parameter and not the main parameter p, and as this increases, it becomes the dominant effect on communication overhead. The computational cost savings are the same regardless of the parameters used.

Table 2 – Savings in Cost of the new Signcryption schemes over corresponding authenticated encryption schemes

	Secure El-Gamal Signcryption		PS		OAS	
	Comp. Cost	Comm. Cost	Comp. Cost	Comm. Cost	Comp. Cost	Comm. Cost
El-Gamal Enc. And Sig. $\|p\|$ = 1024, $\|q\|$ = 160	25%	26.3%	0%	17.5%	33.3%**	25.1%**
El-Gamal Enc. And Sig. $\|p\|$ = 4096, $\|q\|$ = 256	25%	14.3%	0%	9.5%	33.3%**	-1.6%**
OAE and RSA Sig. $\|p\|$ = $\|N\|$ = 1024, $\|q\|$=160	-12.5%*	50.8%*	-50%*	44.9%*	0%	50%
OAE and RSA Sig. $\|p\|$ = $\|N\|$ = 4096, $\|q\|$=256	-12.5%*	57.8%*	-50%*	55.4%*	0%	50%

* Due to messages being different sizes

$$\text{savings} = \frac{\text{Cost(Auth. Enc.)} - 0.75\text{Cost(Signcryption)}}{\text{Cost(Auth. Enc.)}}$$

** Due to messages being different sizes

$$\text{savings} = \frac{\text{Cost(Auth. Enc.)} - (4/3)\text{Cost(Signcryption)}}{\text{Cost(Auth. Enc.)}}$$

Comparing cost is difficult in the sense of obtaining unbiased information from the comparison, this can be seen just by simply comparing El-Gamal type schemes with RSA type schemes. In Zheng's original paper on signcryption [68] the scheme he suggests uses a symmetric key cipher and a keyed hashing function, both of which make his scheme slightly more efficient than the schemes suggested here. However, adding these components mean more assumptions have to be made in order for the scheme to be secure. So Zheng's original scheme has not been compared to the new schemes presented here since computational or communication costs are biased by security assumptions.

The results of Table 2 suggest that Secure El-Gamal signcryption is a good trade off between security and efficiency, with PS not far behind. OAS is efficient, but it has a question over its security.

CHAPTER 7 COMBINING NOTIONS OF SECURITY

The security implications of signcryption were largely ignored by Zheng [68], but they are very relevant. Examining schemes that are secure in the sense of IND-CPA and IND-CCA2 reveal that the IND-CPA schemes are inevitable more efficient, that is they have lower computational and communication cost. The same result is apparent between signature schemes that are NEF-KMA and NEF-CMA. Now if a weak encryption (or signature) scheme could be combined with a strong signature (or encryption) scheme to achieve a better notion of security then it is probable that the cost of the scheme would be small.

This chapter asks some new questions about combining encryption and signature schemes and offers some original discussion towards their possible answer.

7.1 Does NEF-CMA + IND-CPA = INDS-CSA?

Consider starting with an encryption scheme secure in the sense of IND-CPA and concatenating to it (or creating a signcryption scheme with) a signature scheme secure in the sense of NEF-CMA, what would the security be?

Mentioned in section 5.3 was the apparent importance of a signature to determine if a scheme fulfilled the notion of PA, this importance is highlighted here. Consider an NEF-CMA + IND-CPA scheme, to fulfil the notion of PA, an adversary must not be able to create a signcryptogram for which they do not know the plaintext. This scheme achieves this, because if an adversary could create a signcryptogram, this involves signing, but since they do not know the message they achieve existential forgery, but by assumption the scheme is secure in the sense of NEF-CMA. This does imply though that the signature part of the scheme signs the message and not the ciphertext. Since, by assumption the scheme is IND-CPA (and hence INDS-CPA if scheme is a concatenation), the notion of PA is fulfilled and the scheme is INDS-CSA as PA $\Rightarrow$ INDS-CSA.

Unfortunately, what seems intuitively nice is not easily obtainable as a proof. The problem lies with definition of PA, it is only true in the RO model and requires the definition of a knowledge extractor. The knowledge extractor is part of the definition of PA so it can be shown the adversary knew the plaintext for any ciphertext (or signcryptogram) they create. However, a generic knowledge extractor cannot be defined in a proof for the intuitive argument above. It is tempting to use a knowledge extractor that just outputs a 'invalid ciphertext (signcryptogram)', no matter what the input, since it is known the adversary can't create a ciphertext (except from an encryption oracle). But the problem is the adversary can cheat and submit a ciphertext they do know the plaintext for, and this would defeat the proof.

If this conjecture was true, consider combining original El-Gamal (which is secure in the sense of IND-CPA) and DSS (which is secure in the sense of NEF-KMA), in the following scheme.

Original El-Gamal and DSS
Preliminaries
Work over group GF(p), with p - 1 having a large prime factor q, and a generator g of order q. A hash function H.
Key Generation
The private key of receiver is $x_R \in$ GF(p) and public key is $y_R = g^{x_R} \bmod p$. The sender has private key $x_S \in$ GF(p) and public key $y_S = g^{x_S} \bmod p$.
Signcryption
Encrypt a message m 1) Randomly choose $r \in_R [1...q]$ 2) Calculate $s = r^{-1}(m + x_S g^r) \bmod q - 1$ 3) $c_1 = g^r$ 4) $c_2 = y_R^r m$ Signcryptogram is (c_1, c_2, s)
Unsigncryption
1) $m = \dfrac{c_2}{c_1^{x_R}}$ 2) Calculate $\alpha = ms'^{-1} \bmod q, \beta = us'^{-1} \bmod q$ then check If $u \equiv g^{\alpha} y_S^{\beta} \bmod p \bmod q$ then output m else $\varnothing$

This scheme uses only 3 exponentiations and 3 inversions and outputs $2|q| + |p|$ bits (the same as Zheng's scheme [68]). Hence it is not difficult to see that the truth of this conjecture would be useful in creating efficient signcryption schemes.

7.2 Does NEF-KMA + IND-CCA2 = NEF-CSA?

Consider starting with an encryption scheme secure in the sense of IND-CCA2 and concatenating to it (or creating a signcryption scheme with) a signature scheme secure in the sense of NEF-KMA, what would the security be?

It seems that the combination of two schemes of this type would not necessarily yield a more secure scheme in any sense. An adaptive adversary against NEF-KMA can

forge signatures. The adversary then just needs pass the message (that had its signature forged) to the encryption algorithm and create a valid ciphertext. Combining signature and ciphertext would yield a valid, yet forged signcryptogram.

Although probably still not secure, if the NEF-KMA + IND-CCA2 scheme is forced to sign the ciphertext rather than the plaintext then this appears to be more secure than the general case. An adaptive adversary will find it difficult to use an existing signcryptogram to create a forgery. If the encryption scheme is PA this cannot be achieved because doing so would be to create a ciphertext outside of using the encryption oracle. The adversary could forge if they could just sign one ciphertext, but this is selective forgery, SEL-CMA. So there seems to be an argument that SEL-CMA + PA = NEF-CSA, if the ciphertext is signed.

CHAPTER 8 CONCLUSION

There are several new and interesting results to take from this thesis.

Some minor theorems about elliptic curves were presented. Although the theorems are not new work, the proofs provided for the theorems possible represent an original contribution.

The one-way hash variant of the original Zheng-Seberry cryptosystem (ZS-OWH) was shown insecure against an IND-CCA2 adversary. Although a minor change to the scheme could thwart this attack, the security would still be questionable. Hence, the goal of modifying ZS-OWH into a provably secure scheme was set. To achieve this, aspects of the proof from the provably secure scheme by Cramer-Shoup [15] (CS) were borrowed and applied to a variant of El-Gamal. This resulted in a new provably secure scheme called 'Secure El-Gamal', unfortunately its proof relies on the RO model, but only in a minimal way, as it was just required so the hash leaks no partial information.

The Secure El-Gamal scheme is far more efficient than the CS scheme. So as is often the case in cryptography there is a trade off between the assurance of security and efficiency. This trade off is best highlighted by the use of the RO model, where perfectly random hash functions are assumed to exist, and this results in very efficient schemes.

The notion of signcryption, which was introduced by Zheng [68], has been further developed to allow for more formal discussions on the security of a signcryption scheme. This has been achieved by defining notions of security for signcryption schemes that are based upon the notions of security for the underlying signature and encryption schemes. Prior to this thesis, no such formal definitions existed. However, this task is made difficult by the differences between an adversary against confidentiality and one against authentication. These new notions of security allow for an adversary against strong confidentiality to also be one against weak authentication and vice versa. This implies that it is always necessary for a signcryption scheme to have both strong confidentiality and authentication security.

This thesis also presented three new signcryption schemes. All three were based on provably secure (under some assumptions) encryption schemes. The new notions of security for signcryption schemes and the proofs of security for the underlying schemes allowed for informal proofs of confidentiality security for the signcryption schemes (under some assumptions). Strong arguments were made for the authentication security of Secure El-Gamal Signcryption and Pointcheval Signcryption.

Some ideas for future work were discussed regarding combining weak encryption with strong signatures to achieve stronger encryption. Intuitive arguments were made, but the result could not be formally proven. Similarly combining strong encryption with weak signatures was also addressed, but this seemed unlikely to achieve stronger signatures.

CHAPTER 9 BIBLIOGRAPHY

1. Adamson, I.T., *Introduction to Field Theory*. 1964: Oliver and Boyd.
2. Atkins, A., "The number of points on an elliptic curve modulo a prime". 1991(Draft).
3. Bellare, M., Desai, A., Pointcheval, D., and Rogaway, P. "Relations among notions of security for public-key encryption schemes" in *CRYPTO'98*. LNCS 1462, pg 26-45. Springer-Verlag, California, 1998.
4. Bellare, M. and Rogaway, P. "Optimal asymmetric encryption - how to encrypt with RSA" in *EUROCRYTP'94*. LNCS 950, pg 92-111. Springer-Verlag, 1994.
5. Bellare, M. and Rogaway, P. "Random Oracles are practical: a paradigm for designing efficient protocols" in *Proceedings of the 1st Annual Conference on Computing and Communications Security*. 1993.
6. Bender, A. and Castagnoli, G. "On the implementation of elliptic curve cryptosystems" in *CRYPTO'89*. LNCS 435, pg 186-192. Springer-Verlag, 1989.
7. Blake, I., Seroussi, G., and Smart, N., "Elliptic curves in cryptography". *London Mathematical Society Lecture Note Series*, 1999. 265.
8. Bleichenbacher, D. "Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS#1" in *CRYPTO'98*. LNCS 1462, pg 1-12. Springer-Verlag, California, 1998.
9. Blum, Feldman, and Micali. "Non-interactive zero-knowledge and its applications" in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*. 1988.
10. Blum, M. and Goldwasser, S. "An efficient probabilistic public key encryption scheme which hides all partial information" in *CRYTPO'84*. LNCS 196, pg 289-299. Springer-Verlag, 1984.
11. Boneh, D. "The decision Diffie-Hellman problem" in *Third Algorithmic Number Theory Symposium (ANTS)*. LNCS 1423, Springer-Verlag, 1998.
12. Burton, D.M., *Elementary Number Theory*. 1980, Boston: Allyn and Bacon.
13. Canetti, R., Goldreich, O., and Halevi, S., "The Random Oracle methodolgy, revisited". 1998, available at http://theory.lcs.mit.edu/~oded/rom.html.
14. Cohen, H., *A Course in Computational Algebraic Number Theory*. Graduate Texts in Mathematics. Vol. 138. 1996, New York: Springer-Verlag.

15. Cramer, R. and Shoup, V. "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack" in *CRYPTO'98*. LNCS 1462, pg 13-25. Springer-Verlag, California, 1998.
16. Damagard, I. "Towards practical public key systems secure against chosen ciphertext attacks" in *CRYPTO'91*. LNCS 576, pg 433-444. Springer-Verlag, 1991.
17. Demytko, N. "A new elliptic curve based analogue of RSA" in *EUROCRYPT'93*. LNCS 765, pg 40-49. Springer-Verlag, 1993.
18. Diffie, W. and Hellman, M., "New directions in cryptography". *IEEE Transactions on Information Theory*, 1976. IT-22: p. 644-654.
19. Dolev, D., Dwork, C., and Naor, N. "Non-malleable cryptography" in *Proceedings of the 23rd Annual Symposium on Theory of Computing*. pg 542-552. 1991.
20. ds.dial.pipex.com/george.barwood/ec_faq.txt
21. El-Gamal, T., "A public key cryptosystem and a signature scheme based on discrete logarithms". *IEEE Transactions on Information Theory*, 1985. IT-31(4): p. 460-472.
22. Elkies, N.D., "Explicit isogenies". 1991(Draft).
23. Ellis, G., *Rings and Fields*. 1992: Oxford University Press.
24. Fujisaki, E. and Okamoto, T. "How to enhance the security of public-key encryption at minimum cost" in *PKC'99*. LNCS 1560, pg 53-68. Springer-Verlag, 1999.
25. Goldwasser, Macali, and Rivest, "A digital signature scheme secure against adaptive chosen-message attacks". *Siam Journal of Computing*, 1988. 17(2).
26. Goldwasser, Micali, and Rivest. "A 'paradoxical' solution to the signature problem" in *FOCS'84*. 1984.
27. Goldwasser, S. and Micali, S., "Probabilistic encryption". *Journal of Computer and System Sciences*, 1984. 28: p. 270-299.
28. http://www.certicom.com
29. IEEE, "P1363 - Standard specifications for public key cryptography". , http://grouper.ieee.org/groups/1363/index.html.
30. Ireland, K. and Rosen, M., *A Classical Introduction to Modern Number Theory*. Graduate Texts in Mathematics. Vol. 84. 1980, New York: Springer-Verlag.
31. Izu, T., Kogure, J., Noro, M., and K, Y. "Efficient implementation of Schoof's algorithm" in *ASIACRYPT'98*. LNCS 1514, pg 66-79. Springer-Verlag, 1998.
32. Koblitz, N. "CM-curves with good cryptographic properties" in *CRYPTO'91*. LNCS 576, pg 279-287. Spinger-Verlag, 1991.
33. Koblitz, N. "Constructing elliptic curve cryptosystems in characteristic 2" in *CRYPTO'90*. LNCS 537, pg 155-168. Springer-Verlag, 1990.
34. Koblitz, N., "Elliptic curve cryptosystems". *Mathematics of Computation*, 1987. 48: p. 203-209.
35. Koyama, K., Maurer, U., Okamoto, T., and Vanstone, S. "New public-key schemes based on elliptic curves over the ring Z_n" in *CRYPTO'91*. LNCS 576, pg 252-266. Springer-Verlag, 1991.
36. Lay, G. and Zimmer, H. "Constructing elliptic curves with given group order over large finite fields" in *Algebraic Number Theory: First International Symposium*. LNCS 877, pg 250-263. Springer-Verlag, 1994.
37. Ledermann, W., *Introduction to Group Theory*. 1973, Edinburgh: Oliver and Boyd.

38. Lenstra, A. and Lenstra, H., Algorithms in Number Theory, in *Handbook of Theoretical Computer Science*, J. Leeuwen, Editor. 1990, Eisevier Science Publishers. p. 677-681.

39. Lercier, R. "Finding good random elliptic curves for cryptosystems defined over F_{2^n}" in *EUORCRYPT'97*. LNCS 1233, pg 379-392. Springer-Verlag, 1997.

40. Lercier, R. and Morain, F. "Counting the number of points on elliptic curves over finite fields: strategies and performances" in *EUROCRYPT'95*. pg 79-94. Springer-Verlag, 1995.

41. Menezes, A., Okamota, T., and Vanstone, S. "Reducing elltipic curve logarithms to logarithms in a fintite field" in *Proceedings of the 23rd Annual ACM Symposium on the Theory of Computing (STOC'91)*. pg 80-89. 1991.

42. Menezes, A. and Vanstone, S. "The implementation of elliptic curve cryptosystems" in *AUSCRYPT'90*. LNCS 453, pg 2-13. Springer-Verlag, 1990.

43. Menezes, A., Vanstone, S., and Zuccerato, R., "Counting points on elliptic curves over F_{2^m}". *Mathematics of Computation*, 1993. 60(201): p. 407-420.

44. Menezes, A.J., *Applications of Finite Fields*. 1993, Boston: Kluwer Academic Publishers.

45. Menezes, A.J., *Elliptic curve public key cryptosystems*. 1993, Boston: Kluwer Academic publishers.

46. Menezes, A.J., van Oorschot, P.C., and Vanstone, S.A., *Handbook of Applied Cryptography*. Discrete Mathematics and its Applications, ed. K.H. Rosen. 1997: CRC Press.

47. Meyer, B. and Muller, V. "A public key cryptosystem based on elliptic curves over Z/nZ equivalent to factoring" in *EUROCRYPT'96*. LNCS 1146, pg 49-59. Springer-Verlag, 1996.

48. Morain, F. "Building cyclic elliptic curves modulo large primes" in *EUROCRYPT'91*. LNCS 547, pg 328-336. Springer-Verlag, 1991.

49. Naor, M. and Yung, M. "Public-key cryptosystems provably secure aginast chosen ciphertext attacks" in *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing*. pg 427-437. 1990.

50. Niederreiter, H. and Lidl, R., *Finite Fields*. 1983: Addison-Wesley Pub. Co.

51. NIST, "Digital Signature Standard (DSS)", FIPS PUB 186, *US, Department of Commerce*, 1994.

52. Pointcheval, D. "Chosen-ciphertext security for any one-way cryptosystem" in *PKC 2000*. LNCS 1751, pg 129-145. Springer-Verlag, 2000.

53. Pointcheval, D. and Stern, J. "Security proofs for signature schemes" in *EUROCRYPT'96*. LNCS 1070, pg 387-398. Springer-Verlag, 1996.

54. Rabin, M., "Digital signatures and public key encryption as intractable as factorization". Technical Report, MIT/LCS/TR-212, *M.I.T.*, 1978.

55. Rackoff and Simon. "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack" in *CRYPTO'91*. 1991.

56. Rivest, R., Shamir, A., and Adleman, L., "A Method for obtaining digital signatures and public key cryptosystems". *Comm. ACM*, 1978. 21(2).

57. Satoh, T. and Arako, K., "Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves". 1997. preprint.

58. Schnorr, C. "Efficient identification and signatures for smart cards" in *CRYPTO'89*. LNCS 435, pg 235-251. Springer-Verlag, 1989.

59. Schoof, R., "Elliptic curves over finite fields and the computation of square roots mod p". *Mathematics of Computation*, 1985. 44(170): p. 483-494.
60. Shoup, V. "Using hash functions as a hedge against chosen ciphertext attack" in *EUROCRYPT'00*. LNCS 1807, pg 275-288. Springer-Verlag, 2000.
61. Silverman, J.H., *The Arithmetic of Elliptic Curves*. 1986, New York: Springer-Verlag.
62. Smart, N., "An attack on the ECDLP for anomalous elliptic curves". 1997, Announcement in the NMBRTHRY-List.
63. Stewart, I.N. and Tall, D.O., *Algebraic Number Theory*. 1979, New York: John Wiley and Sons.
64. Tsiounis, Y. and Yung, M. "On the security of El-Gamal based encryption" in *PKC'98*. LNCS 1431, Spinger-Verlag, Japan, 1998.
65. www.aba.net.au
66. www.java.sun.com
67. www.sbox.tu-graz.ac.at/home/j/jonny/projects/crypto/asymmetr/content.htm
68. Zheng, Y. "Digital signcryption or how to achieve Cost(Signature & Encryption) << Cost(Signature) + Cost(Encryption)" in *CRYPTO'97*. LNCS 1294, pg 165-179. Springer-Verlag, California, 1997.
69. Zheng, Y., "Improved public key cryptosystems secure against chosen ciphertext attacks", Technical Report 94-1, *University of Wollongong*, 1994.
70. Zheng, Y. and Seberry, J., "Immunizing public key cryptosystems against chosen ciphertext attacks". *IEEE Journal on Selected Areas in Communications*, 1993. 11(5): p. 715-724.

ALFRED HARRIS (Bristol) LTD
5 KINGSDOWN PARADE
BRISTOL BS6 5UD

Phone 0117 9292375
Fax 0117 9273836