CHAPTER VI
SECURITY ANALYSIS
CHAPTER-VI

6.1 INTRODUCTION

This chapter provides security analysis of the cryptographic protocols, which have been presented in previous chapters. First we see that, different types of attacks on signature schemes, security parameters of already existed protocols and what makes the protocols to be strong in the sense of security. Then finally, we discuss the security of new signature schemes in the chapters 2,3,4, and 5 by using the parameters and structures on the underlying work structure.

6.2 TYPES OF ATTACKS ON SIGNATURE SCHEMES

The goal of an adversary is to forge signatures, i.e. produce signatures, which will be accepted as those of some other unauthorized entity. In this section, we discuss the different fundamental attacks on general digital signature schemes. Even though, we are taking precautions to withstand against security attacks, security of the signature schemes in the elementary level, depends on the following parameters.

(a). Security of the private key (Total break):

A digital signature scheme can only be secure, if the problem of constructing the secret signature key from the publicly available information. Information, in particular from the public verification is intractable. The signature schemes that are used today have this property. It is based on the intractability of certain computational problems from the number theory or group theory. However, there is no proof for the intractability of those problems.

(b). Selective forgery:

An attacker is able to create a valid signature for a particular message or class of messages chosen priori. Creating signature does not directly involve the legitimate signer.
(c). Universal forgery:

An adversary can forge the signature of any message and is called universal forgery.

(d). No message attack or Direct attack or existential forgery:

Finding the secret signature key is not only the possible goal of an attacker, he can also try to generate new valid signatures without the knowledge of the secret signature key and this is called existential forgery. To be more precise, the attacker proceeds as follows. (i). The attacker obtains Alice’s public verification key.

(ii). The attacker computes a message ‘x’ and a signature for ‘x’ and that can be verified with Alice’s verification key.

The attacker can compute the document ‘x’ as a function of a public verification key. In specific cases, an attack that also uses knowledge of valid signatures of other documents. Here the situation is simpler, no valid signatures of other documents are used. Therefore this attack is called a “No message attack”. Clearly, the attacker can simply guess a signature. Then this signature has a small probability of being valid. A signature scheme is called secure against a no-message attack, if there is no polynomial time algorithm that attacker can mount such an attack that is successful with non-negligible probability.

In 1984, Shafi Goldwasser, Silvio Micali and Ronald Rivest [GM88] became the first to rigorously define the security requirements of the digital signature schemes. They described a hierarchy of the attack models.

1. Key-only attack: In these attacks, an adversary or attacker knows only the signers public key.
2. Message Attacks: Here an adversary is able to examine signatures corresponding either to known or chosen messages. Message attacks can be further subdivided into three cases.

(a) Known-Message Attack: An adversary has signatures for a set of messages, which are known to the adversary, but not chosen by him.

(b) Chosen-Message Attack: It is not sufficient that a signature scheme is secure against 'no-message attacks'. It is possible that an attacker knows valid signatures and uses them to construct the new signatures. Such attacks, we will see in RSA and ElGamal signature schemes. It is even possible that an attacker obtain signature of his choice, before he generates a new signature.

i). The attacker obtains Alice’s public verification key.

ii). The attacker computes a message ‘x’ and a valid signature for x, that can be verified with Alice’s verification key. During the computation, the attacker always obtain signatures of documents of his choice.

The attack is called chosen message attack. A signature scheme is secure against chosen message attacks, if no polynomial time chosen message attack is possible i.e successful with non-negligible probability.

(c) Adaptive Chosen-Message Attack: An adversary is allowed to use the signer as an oracle. The adversary may request signatures of messages which depend on signer’s public key and he may request signatures of messages, which depend on previously received signatures or messages.

In principle, an adaptive chosen message attack is the most difficult type of attack to prevent. It is conceivable that given enough messages and corresponding signatures, an
adversary could deduce a pattern and then forge a signature of its voice. While an adaptive chosen-message attack may be infeasible to mount in practice, a well designed signature scheme should nonetheless be designed to protect against the possibility.

**Note 1. (Security considerations)** The level of security required in a digital signature scheme may vary according to the application. For example, in situations where an adversary is only capable of mounting a key only attack, it may suffice to design the scheme to prevent the adversary being successful at a selective forgery. In situations, where the adversary is capable of a message attack, it is likely necessary to guard against the possibility of existential forgery.

**Note 2. (Hash functions and digital signature processes** When a hash function ‘h’ is used in a digital signature scheme, h should be a fixed part of the signature process so that an adversary is unable to take a valid signature, replace h with a weak hash function, and then mount a selective forgery attack.

### 6.3 ATTACKS ON RSA SIGNATURE SCHEME

If the RSA signature is implemented as described in section 1.8.1, then there are number of possible attacks.

(a). If an adversary is able to factor the public RSA modulus ‘n’ of some entity A, then the adversary can compute ‘Φ’ and then using Euclidean algorithm, deduce the private key ‘d’ from ‘Φ’ and the public exponent ‘e’ by solving \( ed \equiv 1 \pmod{\Phi} \). This constitutes a total break of the system. To guard against this, A must select p and q, so that factoring ‘n’ is a computationally infeasible task.

(b). In order to verify a signature from Alice, Bob gets Alice’s public key. If the attacker “Oscar” is able to replace Alice’s public key with his own public key without Bob
noticing this, then he can sign in Alice's name. Therefore, it is important that Bob be able to convince himself that he has Alice's authentic public key. This is the reason for using a trust center.

(c) Another attack works as follows. Oscar, an attacker chooses an integer \( s \in \{0, 1, 2, 3, \ldots, n-1\} \). Then he claims that 's' is an RSA signature of Alice. Bob wants to verify this signature. He computes \( m = s^e \mod n \) and believes that Alice has signed 'm'. If 'm' is a meaningful text, then Oscar was able to fake a signature of Alice. This is a no-message attack. Also note that the following proposition holds on RSA signature.

**6.3(a) Proposition**: The RSA signature scheme is

(i) Existentially forgeable under a direct attack

(ii) Universally forgeable under a chosen message attack.

Proof can be found in [TW06]. A variant of the RSA signature scheme that is secure against chosen message attack under certain intractability assumptions can be found in [BR96].

### 6.4 ATTACKS ON ELGAMAL SIGNATURE SCHEME

If the Elgamal signature scheme is implemented as described in section 1.8.2, then we have the following points in the view of security.

(a). *The choice of \( p \)*: If the attacker 'Oscar' can compute discrete logarithms mod \( p \), he can determine Alice's secret key and can generate signatures in Alice's in name. This remains the only known general method of generating Elgamal signatures. Therefore \( p \) must be chosen such that computing discrete logarithms mod \( p \) is infeasible. Given the discrete logarithm algorithms known today, this means that \( p \) should be at least a 768-bit number.
Also, primes of special forms for which certain DL algorithms such as Pohlig-Hellman method are particularly efficient must be avoided. So that the best strategy is to use random primes. It is also dangerous if \( p \equiv 3 \pmod{4} \), the primitive root \( 'g' \) divides \( p-1 \) and computing discrete logarithms in the subgroup of \( Z_p^* \) of order \( 'g' \) is possible. Therefore \( g \) should not divide \( p-1 \).

(b). The choice of \( k \): we show that for every new signature, a new exponent \( k \) must be chosen. This is guaranteed if \( 'k' \) is a random number. Suppose that the signatures \( s_1 \) and \( s_2 \) of the documents \( x_1 \) and \( x_2 \) are generated with same value \( 'k' \), then \( r = g^k \mod p \) is same for both signatures. So that \( s_1 - s_2 = k^{-1}(h(x_1) - h(x_2)) \mod p - 1 \). From this congruence, \( 'k' \) can be determined if \( h(x_1) - h(x_2) \) is invertible modulo \( p-1 \). From \( k, s_1, r, h(x_1), \) Alice's secret key \( 'a' \) can be determined,

Since \( s_1 = k^{-1}(h(x_1) - ar) \mod p - 1 \) and hence \( a = r^{-1}(h(x_1) - ks_1) \mod p - 1 \)

(c) Existential forgery: If no hash function is used in the Elgamal signature system, then the existential forgery is possible under direct attack. Without hash function, the verification congruence is \( A^r r^s \equiv g^x \mod p \), we show, how \( r, s, x \) can be chosen such that this congruence is satisfied. To mount the existential forgery, Oscar chooses two integers \( u, v \) with \( \gcd(v, p-1) = 1 \), then computes

\[
\begin{align*}
  r &= g^u A^v \mod p \\
  s &= -r v^{-1} \mod p - 1 \\
  x &= s \mod p - 1.
\end{align*}
\]

With these values for \( r \) and \( s \), the verification congruence

\[
A^r r^s \equiv A^r g^{su} A^v \equiv A^r g^{su} A^r \equiv g^x \mod p \quad \text{holds.}
\]

This procedure also works, if a collision resistant hash function is used. But since the hash function is a one way function, it is impossible for an attacker Oscar to find a
document 'x' such that the signature generated is the signature of x. Details can be found in [BUC01].

A variant of the Elgamal signature scheme, that is secure against Chosen message attacks under certain intractability assumptions can be found in [PS2K].

6.5 SECURITY OF THE DIGITAL SIGNATURE USING POLYNOMIALS OVERNON-COMMUTATIVE DIVISION SEMIRINGS

In this section, we will show the security of the digital signature scheme proposed in chapter-ii. For this, we follow the definitions in [GM88,BR96], where the concrete security analysis of digital signatures were performed.

Assume that the active eavesdropper "Eve" can obtain, remove, forge and retransmit any message, in which Alice sends to Bob. Any forged data d, we denote it by d_f. We study the security of the signature scheme for three main attacks. Data forgering on valid signature, signature repudiation on valid data, and existential forgering.

**Data forgering:**

Suppose Eve replaces the original message M, with forged one M_f. Then Bob receives the signature (u, s, α, β, v_1). Using forged data M_f or H(M_f), verifying the equation u^v_1 v_1 = s^{v_1} v_2 is impossible, because M_f or H(M_f) is completely involved in the signature generation, but not in the verification algorithm. Hence u^v_1 v_1 = s^{v_1} v_2 is true only for the original message. Data forgery without extracting signature is not possible. Another attempt is to try to find M_f, for valid H(M). But this is impossible, because we assumed that hash function H is cryptographically secure. So the invalid data can't be signed with a valid signature.
Signature Repudiation:

Assume Alice intends to refuse recognition of her signature on some valid data. Then it follows that valid signature \((u, s, \alpha, \beta, v_1)\) can be forged by Eve and she can sign the message \(M\), with the forged signature \((u_f, s_f, \alpha_f, \beta_f, v_1)\) instead. The verification procedure obtains that

\[
V_2 = \alpha_f \cdot y^1 \beta_f = \left[\frac{h(p)^m \cdot r \cdot (f(p)^m \cdot q^{-1} \cdot f(p)^m) \cdot [f(p)^m \cdot H(M) \cdot h(p)^n]}{r}\right]
\]

Since \([f(p)^n \cdot r \cdot [f(p)^m] \neq 1, [f(p)^m \cdot r \neq 1\), where \(I\) is the identity element in the multiplicative structure of the division semiring. Consequently \([u^1 \cdot v_1]_r \neq [s^1 \cdot v_2]_r\). So this signature scheme ensures that the non-repudiation property.

Existential Forgery:

Suppose Eve is trying to sign a forged message \(M_f\). Then she must forge the private key by replacing with some \([f(p)]_r\). Immediately, she faces a difficult with the public key, as we believe that PSD is intractable on non-commutative division semiring. Also note that all the structures in this signature scheme are constructed on non-commutative division semiring and based on PSD. Exact identification these structures are almost intractable as long as PSD is so hard on this underlying work structure. Consequently construction of new valid signatures, without proper knowledge of private key are impossible. So Eve is not able to calculate forged signatures.

Soundness:

The key idea is that choosing a polynomial \(f(x)\) randomly, with semiring assignment and for any \(p \in S\), such that \(f(p) \neq 0\) \(\in (S, +, \cdot)\). A cheating prover \(P^*\) has no way to identify the polynomial \(f(x) \in Z_{>0}[x]\) such that \(f(p) \neq 0\) \(\in (S, +, \cdot)\).
even if he has infinite computational power. Let $n$ be the number of elements of $S$, $P^*$.

best strategy is to guess the value of $p$, and there are $n$ choices for $p$. Hence, even with infinite computing power, the cheating prover $P^*$ with a negligible probability to trace the exact private key $f(p) \in S$, so as to provide a valid response for an invalid signature. Hence this signature scheme is sound.

6.6 SECURITY OF THE DIGITAL SIGNATURE USING

POLYNOMIALS OVER NON-COMMUTATIVE GROUPS

The security of this signature scheme, which is presented in chapter iii, follows section 6.5, but on non-commutative group, with respect to multiplication is the underlying work structure. The security level of this scheme is much higher than the previous scheme, due to the monomorphism on non-commutative group.

6.7 SECURITY OF DIRECTED DIGITAL SIGNATURE USING

POLYNOMIALS OVER NON-COMMUTATIVE DIVISION SEMIRINGS

In this section, we will show the security of the Directed digital signature scheme, which is proposed in chapter-iv. Here we discuss several possible attacks. But we show that, none of these can successfully break our system. The security of the directed signature consists of two requirements, i.e. unforgeability property and verifiable directedness property. We say a directed signature scheme is secure if it satisfies the above two requirements.

6.7 (a) Definition: (Unforgeability) let $A$ be an adversary and Alice be a signer that involved in the following game.

1. $(P_{kA}, S_{kA}) \leftarrow KG(1^k)$ ; where $(P_{kA}, S_{kA})$ is the public and private key pair of Alice.

2. $A$ is given the public key $P_{kA}$ of Alice, a designated verifier Bob's public and private
key pair \((PkB, SkB) \leftarrow KG(1^k)\) and allowed to make signing oracle query to Alice adaptively.

3. Finally, A outputs a signature \(\sigma(m)\) A wins the game if \(\sigma(m)\) is accepted. We define the success probability of A as

\[
\text{Succ} \left( Uf_{Ds} \right)(A) = \text{pr} \left\{ (Pk_A, Sk_A) \leftarrow KG(1^K) \right. \\
\left. \sigma(m) \leftarrow A^\delta(Pk_A, PkB, SkB) \right. \\
\left. : DV(Pk_A, PkB, SkB, \sigma(m)) = \text{accept} \right\}
\]

We say a Directed signature scheme is unforgeable if the success probability of A is negligible in the game.

6.7 (b) Definition : (Verifiable Directedness)

We say a digital signature scheme is a directed signature scheme if

(i). Only the designated verifier Bob, can verify the authenticity of a purport signature issued to him.

(ii). A third party 'Carol' is able to verify the signature only with the help of the signer Alice (or) the designated verifier Bob.

Types of attacks and preventive measures:

(a). If the designated signer B, is dishonest, then he can cheat the original signer A and get her/his signature \(\{s_A, w_B, r_A, M\}\) on any message 'M' of her/his choice.

The solution of this problem is the existence of a trusted third party. The original signer 'A' may stress that all messages between two parties A and B, during the key generation algorithm to be authenticated. The third party keeps the records of original signer's orders and checks any case of designated signer disobeying original signer's order.
(b) Can any one, retrieve the secret keys:

This is as difficult as solving Polynomial symmetrical decomposition problem on non-commutative division semiring. No one get the secret key \( f(p) \), since \( f(x) \in \mathbb{Z}_0[x] \) is a randomly and secretly selected polynomial. On the other hand, by using the public keys \( A \) and \( B \) are \( y = f(p)^m q f(p)^m \), \( z = g(p)^m q g(p)^m \), no one get the secret keys \( f(p) \) and \( g(p) \), as we believe that polynomial symmetrical decomposition problem is not tractable on non-commutative division semiring.

(c) Can one impersonate the designated signer \( B \)

A forger may try to impersonate the designated signer \( B \), by choosing a random polynomial \( h_1(x) \in \mathbb{Z}_0[x] \) such that \( h_1(p) \in (\mathbb{R}, +, \cdot) \) and an element \( q_1 \in (\mathbb{R}, +, \cdot) \) then calculates

\[
\begin{align*}
    w_B &= h_1(p)^n q h_1(p)^m \\
    z_0 &= h_1(p)^n z h_1(p)^m \\
    r_A &= H(z_B, w_B, M) \quad \text{for some message } M
\end{align*}
\]

but without knowing the secret key \( f(p) \), it is highly difficult to generate a valid \( s_A = f(p)^m r_A f(p)^m \) and hence to satisfy verification equation \( r_A = H(z_B, w_B, M) \)

where \( z_0 = \lambda^{-1} \mu \lambda \cdot \) for \( \lambda = s_A^{-m} g(p)^m \) and \( \mu = s_A^{-m} w_B s_A^m \) so that signature is secure under existential forgery.

(d) Can one forge a signature \( \{ s_A, w_B, r_A, M \} \) using the equation \( \mu = s_A^{-m} w_B s_A^m \)

Computing the element \( s_A \in (\mathbb{R}, +, \cdot) \), a non-commutative division semiring, from the above equation is equivalent to solving symmetrical decomposition problem.

We believe that PSD is intractable on the designed underlying work structure.

Even then, if any forger selects \( s^* \) in some way, and sends the fake signature \( \{ s^*, w_B, r_A, M \} \) to \( B \), receiver \( B \) computes
\[ \mu^* = (s^*)^{-m} w_B (s^*)^m \quad \lambda^* = (s^*)^{-m} . g(p)^m \quad z_B^* = \left( \lambda^* \right)^{-1} \mu^* \lambda^* \]

Then concludes that \( r_A = H( z_B, w_B, M) \neq H(z_B^*, w_B, M) \) and detects the forgery.

(e) Data forgering:

Assume that a forger replaces the original message ‘M’ with a forgered one ‘Mf’, then he sends \( Mf \) to the designated signer B and taking the signature \( \{ s_A, w_B, r_A, Mf \} \) and verifies that \( r_A = H( z_B, w_B, Mf) \), but verification fails since

\[ r_A = H( z_B, w_B, M) \neq H( z_B, w_B, Mf) \]

Another attempt is try to find \( Mf \), for valid \( r_A \). But this is impossible in two ways, because we assume that hash function (h) is cryptographically secure in one side, and he is not able to find \( z_B \) as it is CSP based, so it is intractable in other side. So the invalid data, can’t be signed with a valid signature.

(f) Signature repudiation:

Assume original signer “A” intends to refuse recognition of his signature on some valid data. Then valid signature \( \{ s_A, w_B, r_A, M \} \) can be forgered by an attacker Eve and she can the message M with forgered signature \( \{ s_A^*, w_B^*, r_A^*, M^* \} \) instead. Then the designated verifier B computes

\[ \mu^* = (s^*)^{-m} w_B (s^*)^m \quad \lambda^* = (s^*)^{-m} . g(p)^m \quad z_B^* = \left( \lambda^* \right)^{-1} \mu^* \lambda^* \]

And then checks the validity of the signature by using the inequality

\[ r_A = H( z_B, w_B, M) \neq H(z_B^*, w_B^*, M^*) \] and detects the forgery.

There is a negligible probability to hold this equality, because this equation needs elements satisfying commutative property on non-commutative division semiring and
parameters based on PSD problem. So Eve is not able to find those elements on \((R, +, \cdot)\). Hence, this signature scheme ensures the non-repudiation property.

6.8 SECURITY OF ALLOCATION OF REGISTRATION NUMBER

The security of Allocation of registration number is as same as the security of the scheme given chapter-iv and analyzed in section 6.7, because allocation of registration number is an application of directed digital signature scheme on non-commutative division semiring. Thus by the construction facilities, the allocation of registration number in the electronic world with the following characteristics.

(i). Only the user can his/her registration number, due to the property of directed Signature scheme.

(ii). The problems of forgery can be easily identified and solved easily.

(iii). By using this system, we can minimize the possible misuse of the present system

(iv). The obvious advantage of our scheme over present system is that resulting registration number has no meaning to any third person.

(v). Since the relation between the signature and the signer secret key is not known to any one, but the designated receiver. Hence the security level is much higher than any scheme based on discrete logarithms.