**DIGITAL SIGNATURE**

**INTRODUCTION**

A digital signature of a message is a number dependent on some secret known only by a signer, and additionally on the content of the message being signed. The signatures must be verifiable; an unbiased third party should be able to resolve any matter occurring equitably without having to get access to the signer’s secret information (secret key).

The digital signatures have many applications in information security (authentication, data integrity, non-repudiation). Still, one of the most significant applications of digital signatures is the certification of public keys in large networks. For a TTP(TrustedThirdParty) this application is important, gives ability to bind the identity of a user to a public key, so later, other entities can authenticate a public key without assistance from a TTP.

One of the first method used is the RSA signature scheme, remaining today the most practical and versatile technique available, but others digital signature technique has been created since then, providing advantaged in terms of functionality and implementation.

**1. A FRAMEWORK FOR DIGITAL SIGNATURE MECHANISMS**

***Basic definitions***

Digital signature: data string associates a message(digital) with some originating entity.

Digital signature generation algorithm: method to produce a digital signature.

Digital signature verification algorithm: method to verify that a digital signature is authentic.

Digital signature scheme: consists of a signature generation algorithm and an associated verification algorithm.

Digital signature signing process: consists of a digital signature generation algorithm, with a method for formatting data into messages that can be signed.

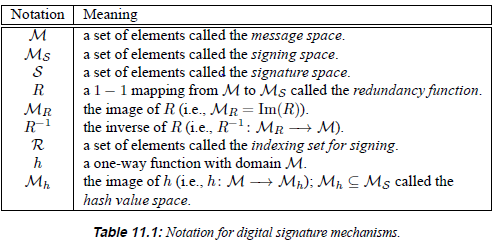
Digital signature verification process: consists of a verification algorithm with a method to recover the data from the message.

Message space: set of elements to which a signer can affix a digital signature

Signing space: set of elements to which the signature transformations are applied (not applied on the message)

Signature space: set of elements associated to the message, it is used to bind the signer to the message.

Indexing set: identify the specific signing transformation.



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

We have two types of digital signature schemes:

1. Appendix: require the original message as input to the verification algorithm
2. Message recovery: do not require the original message as input to the verification algorithm, the original message is recovered from the signature itself.

A randomized digital signature scheme is:

1. Randomized: if the absolute value of the indexing set is bigger than 1.

2. Otherwise: deterministic

* 1. **DIGITAL SIGNATURE SCHEMES WITH APPENDIX (DSA)**

This is the most commonly used in practice, they rely on cryptographic hash functions rather than customized redundancy functions, and less prone to existential forgery attacks. They are mostly applied to messages of arbitrary length. The following algorithm is selected to be collision-free hash function. A known alternative to hashing is to break the message into blocks of a fixed length which can be individually signed using signature scheme with message recovery. But since signature generation is slow and reordering multiples signed blocks presents a security risks, the hash is the preferred method.

* + 1. **Key generation**

SUMMARY: Each entity creates a private key for signing messages, and a corresponding public key to be used by other entities for verifying signatures.

1. Each entity A should select a private key defining,  ,of transformations. Each  is a 1-1 mapping from  to *S* and is called a signing transformation.
2. defines a corresponding mapping  from  to {true, false} such that :





for all  , here  for  .  is called a verification transformation and is constructed such that it may be computed without knowledge of a signer’s private key.

1. A’s public key is , A’s private key is the set .
   * 1. **Signature generation and verification**

SUMMARY: Entity A produces a signature  for a message  , which can be later verified by any entity B.

1. Signature generation [Entity A should do this]:
2. Select an element 
3. Compute , and 
4. A’s signature for . Both are available for verification.
5. Verification [Entity B should do this]:
6. Obtain A’s authentic public key  .
7. Compute  and  .
8. Accept the signature if and only if *u* = true.
   * 1. **Properties**
9. For each  should be efficient to compute.

(2) should be efficient to compute.

(3) Should be computationally infeasible for an entity other than A to find  and a  such that  , where .

* 1. **DIGITAL SIGNATURE SCHEMES WITH MESSAGE RECOVERY (RSA)**

This digital signature scheme has the feature that the message signed can be recovered from the signature itself. So it is used for short messages.

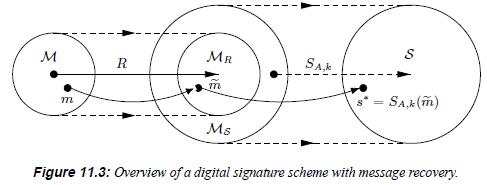
* + 1. **Key generation**

SUMMARY: Each entity creates a private key for signing messages, and a corresponding public key to be used by other entities for verifying signatures.

1. Each entity A should select a private key defining,  ,of transformations. Each  is a 1-1 mapping from  to *S* and is called a signing transformation.
2. defines a corresponding mapping  with the property that  is the identity map on  for all  .  is called a verification transformation and is constructed such that it may be computed without knowledge of a signer’s private key.
3. A’s public key is , A’s private key is the set .
   * 1. **Signature generation and verification**

SUMMARY: Entity A produces a signature  for a message  , which can be later verified by any entity B. The message *m* is recovered from *s.*

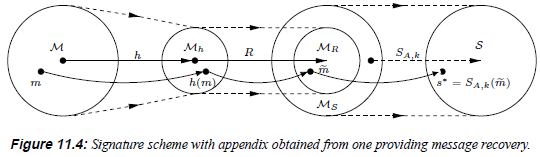
1. Signature generation [Entity A should do this]:
2. Select an element 
3. Compute  and  .
4. A’s signature is and available to entities to verify the signature and recover *m* from it.
5. Verification [Entity B should do this]:
6. Obtain A’s authentic public key 
7. Compute 
8. Verify that 
9. Recover *m* from  by computing .



* + 1. **Properties**

1. For each  should be efficient to compute.
2. should be efficient to compute.
3. Should be computationally infeasible for an entity other than A to find any  such that .
   1. **Remarks and note**

Any digital signature scheme with message recovery can be turned into a digital signature scheme with appendix by hashing the message and signing the hash value. And at this point the redundancy function R is no longer critical to the security of the signature scheme and can be any 1-1 function from  to .



* 1. **Types of attacks on signature schemes**

The goal of an adversary is to forge signatures, produce signatures that will be accepted as those of other entities.

Types of attacks:

1. Total break: adversary is able to compute the private key information of a signer or an efficient signing algorithm equivalent to the valid one.
2. Selective forgery: adversary is able to create valid signature for a particular message chosen.
3. Existential forgery: adversary able to forge signature for at least one message, but has little or no control over the message whose signature is obtained.

Two basic attack against public-key digital signature schemes:

1. Key-only attacks: adversary knows the signer’s public key
2. Message attacks: adversary able to examine signatures corresponding to known or chosen messages. Subdivided into 3 classes:
3. Known-Message attack: adversary has signature for a set of message known by him but not chosen.
4. Chosen-Message attack: adversary obtains valid signature from a chosen list of message before attempting to break the signature scheme (non-adaptive attack)
5. Adaptive chosen-message attack: adversary may request signatures of messages depending on the signer’s public key or on previously obtained signatures or messages.