Implementation and Comparison of Various Digital Signature Algorithms

-Nazia Sarang
Boise State University
What is a Digital Signature?

A digital signature is used as a tool to authenticate the information that is being sent from a person. Digital signatures are used in many fields like software distribution and financial transactions.
A Digital Signature scheme consists of three algorithms:

- A key generation algorithm.
- A signing algorithm.
- A signature verifying algorithm.
Why do you need Digital Signature?

- **Authentication**: Digital Signatures can be used to authenticate the source of messages.

- **Integrity**: If a message is digitally signed, any change in the message after signature will invalidate the signature. So both the sender and the receiver can be sure that the message is not being altered in the middle.
A few Digital Signature Algorithms

- Full Domain Hash
- SHA 1 with RSA
- Rabin Signature Algorithm
- Pointcheval-Stern Signature Algorithm
- BLS (Bonneh-Lynn-Shacham)
A simple Digital Signature Algorithm

At the sender's side:

- Generate the public and the private keys of the sender.
- Create the signature of the sender on a given message.
- Encrypt the message and send it across to the receiver.
At the receiver's side:

- The receiver receives the signed and encrypted message.
- He then decrypts the message with his private key.
- The resulting value must be equal to the signed message by the sender.
- This way the receiver gets confirmed about the authenticity of the sender.
Additional Security to the Messages

The messages can be first converted into some other cryptic statement and then signed by the sender. This way we are putting in extra robustness to the security issue. Such an implementation can be done with the help of some hash functions.
A hash function is an algorithm that takes a message of any length as input and produces a fixed-length string as output. Hash functions are sometimes known as “message digests” or “digital fingerprints”. MD5 and SHA-1 are currently the two most commonly used hash functions.
SHA1 Algorithm

INPUT: Bitstring x of bitlength $b \geq 0$. OUTPUT: 160-bit hash-code of x.

- Four 32 bit initial chaining values are taken.

- Pad x such that its bitlength is a multiple of 512, as follows. Append a single 1-bit, then append $r - 1$ ($\geq 0$) 0-bits for the smallest r resulting in a bitlength 64 less than a multiple of 512. Finally append the 64-bit representation of $b \mod 264$, as two 32-bit words with least significant word first.

- The final two 32-bit words specifying the bitlength b is appended with most significant word preceding least significant.
For each $i$ from 0 to $m - 1$, copy the $i$th block of sixteen 32-bit words into temporary storage: $X[j] \leftarrow x_{16i+j}$, $0 \leq j \leq 15$, and process them in four 20-step rounds before updating the chaining variables.

The final hash-value is: $H_1 \ || H_2 \ || H_3 \ || H_4 \ || H_5$. 
Full Domain Hash (FDH) is an RSA-based signature scheme that follows the hash-and-sign paradigm. Any of the hash functions can be used.
FDH can be implemented as a normal RSA Signature Algorithm except that the message is first converted into a message digest.

The message digest is then encrypted on the same lines as the RSA Cryptosystem.
The Rabin signature scheme is a variant of the RSA signature scheme. Verification is faster than signing, as with RSA signatures. So, the message to be signed must have a square root mod n.
contd...

Signature: \( s = m^{1/2} \mod n \) where \( s \) is the signature

Verification: \( m = s^2 \mod n \)
Pointcheval-Stern Signature Algorithm

- Pointcheval-Stern signature algorithm is a digital signature scheme based on the closely related ElGamal signature scheme.

- It changes the ElGamal scheme slightly to produce an algorithm which has been proven secure in a strong sense against adaptive chosen-plaintext attacks by implementing the forking lemma technique.
Boneh–Lynn–Shacham signature scheme allows a user to verify that a signer is authentic. The scheme uses a pairing function for verification and signatures are group elements in some elliptic curve. Working in an elliptic curve provides defense against index calculus attacks against allowing shorter signatures than FDH signatures.
Algorithm

Key generation:
The key generation algorithm selects a random integer $x$ in the interval $[0, r - 1]$. The private key is $x$. The holder of the private key publishes the public key, $g^x$.

Signing the message:
Given the private key $x$, and some message $m$, we compute the signature by hashing the bitstring $m$, as $h = H(m)$. We output the signature $\sigma = h^x$. 
Verification:
Given a signature $\sigma$ and a public key $g^x$, we verify that $e(\sigma,g) = e(H(m),g^x)$. 
Conclusion

I implemented the SHA1 Algorithm and a few Digital Signature Algorithms. Security is enhanced by incorporating the message digests.