Introduction to Cryptography
CS 355

Lecture 30

Digital Signatures
Announcements

• Wednesday’s lecture cancelled

• Friday will be guest lecture by Prof. Cristina Nita-Rotaru on Identification Schemes
  – This topic will be covered in the final exam
Lecture Outline

- Digital signatures
- Security requirements
- The RSA signatures
- The El Gamal signatures
- The DSA signatures
Where Does This Fit?

<table>
<thead>
<tr>
<th>Secret Key Setting</th>
<th>Public Key Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stream cipher</td>
<td>Public key encryption: RSA, El Gamal, etc.</td>
</tr>
<tr>
<td>Secrecy / Confidentiality</td>
<td>Block cipher + encryption modes</td>
</tr>
<tr>
<td>MAC</td>
<td>Digital Signatures</td>
</tr>
<tr>
<td>Authenticity / Integrity</td>
<td></td>
</tr>
</tbody>
</table>
Digital Signatures: The Problem

- Consider the real-life example where a person pays by credit card and signs a bill; the seller verifies that the signature on the bill is the same with the signature on the card.
- Contracts, they are valid if they are signed.
- Can we have a similar service in the electronic world?
Digital Signatures

- Digital Signature: a data string which associates a message with some originating entity.

- Digital Signature Scheme:
  - a signing algorithm: takes a message and a (private) signing key, outputs a signature
  - a verification algorithm: takes a (public) key verification key, a message, and a signature

- Provides:
  - Authentication
  - Data integrity
  - Non-Repudiation (MAC does not provide this.)
Adversarial Goals

- **Total break**: adversary is able to find the secret for signing, so he can forge then any signature on any message.
- **Selective forgery**: adversary is able to create valid signatures on a message chosen by someone else, with a significant probability.
- **Existential forgery**: adversary can create a pair (message, signature), s.t. the signature of the message is valid.

- A signature scheme can not be perfectly secure; it can only be computationally secure.
- Given enough time and adversary can always forge Alice’s signature on any message.
Attack Models for Digital Signatures

- **Key-only attack**: Adversary knows only the verification function (which is supposed to be public).
- **Known message attack**: Adversary knows a list of messages previously signed by Alice.
- **Chosen message attack**: Adversary can choose what messages wants Alice to sign, and he knows both the messages and the corresponding signatures.
Digital Signatures and Hash

- Very often digital signatures are used with hash functions, hash of a message is signed, instead of the message.
- Hash function must be:
  - Pre-image resistant
  - Weak collision resistant
  - Strong collision resistant
RSA Signatures

Key generation (as in RSA encryption):
- Select 2 large prime numbers of about the same size, p and q
- Compute $n = pq$, and $\Phi = (q - 1)(p - 1)$
- Select a random integer $e$, $1 < e < \Phi$, s.t. $\gcd(e, \Phi) = 1$
- Compute $d$, $1 < d < \Phi$ s.t. $ed \equiv 1 \mod \Phi$

Public key: $(e, n)$
Secret key: $d$
RSA Signatures (cont.)

**Signing message M**
- Verify $0 < M < n$
- Compute $S = M^d \mod n$

**Verifying signature S**
- Use public key $(e, n)$
- Compute $S^e \mod n = (M^d \mod n)^e \mod n = M$

Note: in practice, a hash of the message is signed and not the message itself.
RSA Signatures (cont.)

Example of forging

- Attack based on the multiplicative property of RSA.
  \[ y_1 = \text{sig}_K(x_1) \]
  \[ y_2 = \text{sig}_K(x_2), \text{ then} \]
  \[ \text{ver}_K(x_1x_2 \mod n, y_1y_2 \mod n) = \text{true} \]

- So adversary can create the valid signature \( y_1y_2 \mod n \) on the message \( x_1x_2 \mod n \)
- This is an existential forgery using a known message attack.
El Gamal Signature

Key Generation (as in ElGamal encryption)

- Generate a large random prime $p$ such that DLP is infeasible in $\mathbb{Z}_p$ and a generator $g$ of the multiplicative group $\mathbb{Z}_p$ of the integers modulo $p$.
- Select a random integer $a$, $1 \leq a \leq p-2$, and compute $y = g^a \mod p$.
- Public key is $(p; g; \beta = g^a)$.
- Private key is $a$.

- Recommended sizes: 1024 bits for $p$ and 160 bits for $a$.\[\]
ElGamal Signature (cont.)

Signing message M

- Select random \( k \), \( 1 \leq k \leq p-1 \), \( k \in \mathbb{Z}_{p-1}^* \)
- Compute
  
  \[
  r = g^k \mod p \\
  s = k^{-1}( h(M) - ar ) \mod (p-1)
  \]

- Signature is: \((r,s)\)
- Size of signature is double size of \( p \)

NOTE: \( h \) is a hash function
ElGamal Signature (cont.)

Verification

- Verify that \( r \) is in \( \mathbb{Z}_{p-1}^* \): \( 1 \leq r \leq p-1 \)
- Compute
  \[
  v_1 = \beta^r r^s \mod p \\
  v_2 = g^{h(M)} \mod p
  \]
- Accept iff \( v_1 = v_2 \)

Signature is: \((r, s)\)

\[
\begin{align*}
  r &= g^k \mod p \\
  s &= k^{-1} ( h(M) - ar ) \mod (p-1)
\end{align*}
\]
ElGamal Signature (cont.)

Security of ElGamal signature

- Weaker than DLP
- k must be unique for each message signed
- Hash function h must be used, otherwise easy for an existential forgery attack
  - without h, a signature on \( M \in \mathbb{Z}_p \), is \((r,s)\) s.t. \( \beta^r r^s = g^M \mod p \)
  - choose \( u,v \) s.t. \( \gcd(v,p-1)=1 \), then let \( r=g^u \beta^v \mod p \) \( p=g^{u+av} \mod p \), and let \( s=-rv^{-1} \mod (p-1) \)
  - then \( \beta^r r^s = g^{ar} (g^{u+av})^s = g^{ar} g^{avs} g^{us} = g^{ar} g^{av(-rv^{-1})} g^{us} = g^{ar} g^{-ar} g^{us} = g^{us} \)
  - i.e., \((r,s)\) is a signature of the message \( u \)
ElGamal Signature (Continued)

• $0 < r < p$ must be checked, otherwise easy to forge a signature on any message if an valid signature is available.
  – Given $M$, and $r = g^k$, $s = k^{-1}(h(M) - ar) \mod (p-1)$
  – For any message $M'$, let $u = h(M') / h(M) \mod (p-1)$
  – Computes $s' = su \mod (p-1)$ and $r'$ s.t.
    \[ r' \equiv ru \pmod{(p-1)} \quad \text{AND} \quad r' \equiv r \pmod{p}, \]
    then
    \[ \beta^{r'} r^{s'} = \beta^{ru} r^{su} = (\beta^r r^s)^u = (g^{h(M)})^u = g^{h(M')} \]
Digital Signature Algorithm (DSA)

Specified as FIPS 186

**Key generation**
- Select a prime q of 160-bits
- Choose 0 ≤ t ≤ 8
- Select \(2^{511+64t} < p < 2^{512+64t}\) with q \(\mid p-1\)
- Let \(\alpha\) be a generator of \(Z_p^*\), and set \(g = \alpha^{(p-1)/q} \mod p\)
- Select 1 ≤ a ≤ q-1
- Compute \(\beta = g^a \mod p\)

Public key: (p, q, g, \(\beta\))
Private key: a
Signing message $M$:

- Select a random integer $k$, $0 < k < q$
- Compute
  
  \[ r = (g^k \mod p) \mod q \]
  \[ s = k^{-1} ( h(M) + ar) \mod q \]
- Signature: $(r, s)$

Note: FIPS recommends the use of SHA-1 as hash function.
Verifying the DSA Signature:

- **Verification**
  - Verify $0 < r < q$ and $0 < s < q$, if not, invalid.
  - Compute
    
    \[
    u_1 = h(M)s^{-1} \mod q, \\
    u_2 = rs^{-1} \mod q
    \]
  - Valid if $r = (g^{u_1} \beta^{u_2} \mod p) \mod q$
    
    \[
    g^{u_1} \beta^{u_2} = g^{h(M)s^{-1}} g^{ars^{-1}} = g^{(h(M)+ar)s^{-1}} = g^k \mod p
    \]

Signature: $(r, s)$

\[
\begin{align*}
    r &= (g^k \mod p) \mod q \\
    s &= k^{-1} (h(M) + ar) \mod q
\end{align*}
\]
Coming Attractions …

- Security protocols