ICS 268: Cryptography and Communication Security 11/21/2004

Homework 5

Due **Tuesday**, *12/2/2004*

1 Insecure Variant of the Schnorr Signature Scheme

Consider the following variant of the Schnorr signature scheme (i.e. the discrete-log based signature scheme we discussed in class): p, q are prime, g is an element of order q in $\mathbb{Z}_p^*$, the private key is $x \in \mathbb{Z}_q$, public key is $y = g^x \bmod p$, and the signature on m is a pair (r, z) s.t. $r = g^k \bmod p$ for a random $k \in \mathbb{Z}_q$, and $z = k + x(m * r) \bmod q$.

(This is a simplification of the Schnorr scheme where hash $H(m, r)$ is replaced with just a product $m * r \bmod q$.)

1.1

What is the verification equation for this signature scheme?

1.2 Existential Forgery

Show that this signature is not secure against an existential forgery even if the adversary sees only the public key, i.e. the adversary does not need to see any message/signature pairs.

If you do not see how to do this, show that this signature is not secure against an existential forgery after the adversary sees only one valid (message,signature) pair.

1.3 Extension to Uniformly Generated Forgeries

Extend your attack to show an efficient adversary which can actually create (message,signature) pairs where the message is a uniformly distributed element in $\mathbb{Z}_q$.

If your attack creates signatures for uniformly distributed messages but only provided that $q = p - 1$ and g is a generator of $\mathbb{Z}_p^*$, that's good too.

(Again, your adversary can create such message/signature pairs either just from the public key or, in the easier scenario, after seeing one message/signature pair.)

2 Vulnerabilities of the "plain" RSA signatures

2.1 Existential forgery of plain RSA signatures

Suppose that the RSA cryptosysem is used as a signature scheme in the following plain way: The key generation algorithm picks (n, e) as a public key and d as the private key, where $ed = 1 \bmod \phi(n)$, and $n = pq$ where p, q are random big primes of equal size. The signature algorithm outputs $\sigma = SIG_d(m) = m^d \bmod n$, and the verification algorithm $Ver_{(n,e)}(m, \sigma)$ output 1 if $\sigma^e = m \bmod n$. (Assume that the signature scheme is used only for short messages, i.e. $m \in \mathbb{Z}_n^*$.)

Show that this "plain RSA" signature scheme is insecure in the sense of existential forgery.

Moreover, show how the attacker, seeing just the public key, can create valid (m, σ) pairs s.t. m is *uniformly distributed* in $\mathbb{Z}_n^*$.

2.2 Trying to pad plain RSA to avoid the above attack

Imagine instead that $SIG_d(m) = (m|p)^d \bmod n$ where p is a fixed k-bit pad, i.e. p is some constant k-bit string. (This modified signature scheme will work only for even shorter messages, as now m needs to be at most $n - 2^k$.) How long should the pad p be so that the above attack fails to produce a valid (message,signature) pair with significant probability and in a feasible running time? (Take anythig you think is realistic as "signficiant probability" and "feasible running time".)

Side Note: Such padding does fend off the above simple attack, but more sophisticated attacks show that an RSA signature scheme remains existentially forgeable even for much longer pads.

2.3 Total break of plain RSA signatures under the CMA attack

Show an adversary who can sign any message $m \in \mathbb{Z}_n^*$ if he can stage the CMA attack on the real signer in which he gets the signer to sign two messages m_1 and m_2, where $m_1 \neq m$ and $m_2 \neq m$. Show that one can do it so that both m_1 and m_2 are uniformly (although not independenty) distributed in $\mathbb{Z}_n^*$.

3 Insecurity of common modulus for RSA cryptosystems

3.1 "Insider attack": One user attacks another.

Show that if two users, $i = 1$ and $i = 2$, share the same RSA modulus n as part of their RSA public keys,[1] their public keys are (n, e_i) and private keys are d_i s.t. $e_i d_i = 1 \bmod \phi(n)$, then one user can compute the private key of the other one, and thus one user can totally break the cryptosystem of the other user.

3.2 "Outsider attack": Simmons' attack on plain RSA

The following attack on using RSA cryptosystems with common modulus is due to Simmons. Assume that the plain RSA permuation is used as encryption, i.e. that $ENC_{(n,e)}(m) = m^e \bmod n$ (assuming m is a short message, i.e. $m \in \mathbb{Z}_n^*$), and $DEC_{(d)}(c) = c^d \bmod n$. Assume also that two users share the same RSA modulus n and that their public exponents are e_1 and e_2 satisfy $gcd(e_1, e_2) = 1$. Show that if some sender encrypts the same message m to both users under their public keys (n, e_1) and (n, e_2) then anyone can compute m given these two ciphertexts.

The extended Euclidean algorithm is helpful here, i.e. use it first to find integers α_1, α_2 s.t. $\alpha_1 e_1 + \alpha_2 e_2 = 1$, and then show how to recover m.

3.3 Extension to any RSA-based *deterministic* cryptosystem.

Consider that $Enc_{(n,e)}(m) = (pad(m))^e \bmod n$ where $pad(\cdot)$ is any deterministic procedure which can be inverted, as it must be if there is a feasible decryption procedure which gets m back from $pad(m) = c^d \bmod n$. Show that Simmons' attack extends to any such scheme.

[1] This attack holds regardless whether the RSA keys are used for encryption or signatures, and regardless if they are used in a plain way or with paddings.