Cryptography, winter 2014/15
ElGamal encryption and RSA

Dr. Daniel Loebenberger
Cryptosystem. ElGamal encryption scheme.

General setup:

Input: $1^n$.

1. Choose a finite cyclic group $G = \langle g \rangle$ with $d$ elements, where $d$ is an $n$-bit integer, and a generating element $g$. These data, including a description of $G$, are made public.

Key generation:

Output: $(sk, pk)$.

2. Choose a secret key $sk = b \leftarrow \mathbb{Z}_d$ at random. The public key is $pk = B = g^b \in G$. 
Cryptography. ElGamal encryption scheme.

Encryption:

Input: Plaintext $x \in G$.
Output: Ciphertext $\text{enc}_{pk}(x) \in G^2$.

1. Choose a secret session key $a \leftarrow \mathbb{Z}_d$ at random.
2. Public session key $A \leftarrow g^a \in G$, and common session key $k \leftarrow B^a = g^{ab} = A^b$.
3. $y \leftarrow x \cdot k \in G$, Return $\text{enc}_{pk}(x) = (y, A)$.

Decryption:

Input: Arbitrary $(y, A) \in G^2$.
Output: Decryption $\text{dec}_{sk}(y, A) \in G$.

4. Common session key $k \leftarrow A^b$, inverse $k^{-1} \in G$ of the common key.
5. $z \leftarrow y \cdot k^{-1}$, Return $\text{dec}_{sk}(y, A) = z$. 
Let $G = \mathbb{Z}_{2579}^\times$ and $g = 2$ again. Bob has published his public key $B = g^b = 949$. Alice wants to encrypt the plaintext MY SECRET and writes it in the familiar numerical notation $(12, 24, 18, 4, 2, 17, 4, 19)$, where A corresponds to 0, etc. Spaces are often ignored in cryptography. Alice sends two letters $b_1, b_2$ as one unit by computing $26b_1 + b_2$. She uses different secret session keys to avoid statistical attacks, and sends $(y, A)$ as her message.

<table>
<thead>
<tr>
<th>plaintext letters</th>
<th>$x$</th>
<th>encryption</th>
<th>decryption</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$a$</td>
<td>$A = g^a$</td>
</tr>
<tr>
<td>MY</td>
<td>336</td>
<td>2111</td>
<td>1617</td>
</tr>
<tr>
<td>SE</td>
<td>472</td>
<td>367</td>
<td>734</td>
</tr>
<tr>
<td>CR</td>
<td>69</td>
<td>351</td>
<td>832</td>
</tr>
<tr>
<td>ET</td>
<td>123</td>
<td>161</td>
<td>1355</td>
</tr>
</tbody>
</table>
Theorem

For any group $G$, breaking the ElGamal cryptosystem and solving the Diffie–Hellman Problem can be reduced to each other in polynomial time. The reductions can be done using at most $O(n^2)$ bit operations, where $n$ is the bit size of $G$. 
**CRYPTOSYSTEM. RSA.**

Key Generation key gen.

Input: Security parameter $n$.
Output: secret key $sk$ and public key $pk$.

1. Choose two distinct primes $p$ and $q$ at random with $2^{(n-1)/2} < p, q < 2^{n/2}$.
2. $N \leftarrow p \cdot q$, $L \leftarrow (p - 1)(q - 1)$. [$N$ is an $n$-bit number, and $L = \phi(N)$ is the value of Euler’s $\phi$ function.]
3. Choose $e \leftarrow \{2, \ldots, L - 2\}$ at random, coprime to $L$.
4. Calculate the inverse $d$ of $e$ in $\mathbb{Z}_L$.
5. Publish the public key $pk = (N, e)$ and keep $sk = (N, d)$ as the secret key.
**Cryptosystem. RSA.**

Encryption enc.

Input: $x \in \mathbb{Z}_N$, $pk = (N, e)$.

Output: $\text{enc}_{pk}(x) \in \mathbb{Z}_N$.

1. $y \leftarrow x^e$ in $\mathbb{Z}_N$.
2. Return $\text{enc}_{pk}(x) = y$.

Decryption dec.

Input: $y \in \mathbb{Z}_N$, $sk = (N, d)$.

Output: $\text{dec}_{sk}(y) \in \mathbb{Z}_N$.

3. $x^* \leftarrow y^d$ in $\mathbb{Z}_N$.
4. Return $\text{dec}_{sk}(y) = x^*$. 
security parameter $n$,
distinct random primes $p$ and $q$ of $n/2$ bits,
$N = pq$ of $n$ bits,
$L = \phi(N) = (p - 1)(q - 1),$
e, $d \in \mathbb{Z}_L \setminus \pm 1$ with $ed = 1$ in $\mathbb{Z}_L$,
plaintext $x$, ciphertext $y$, decryption $x^*$, all in $\mathbb{Z}_N$,
y = $x^e$, $x^* = y^d$.

Figure: The RSA notation.
Chinese Remainder Theorem

Let \( N = q_1 \cdots q_r \) with pairwise coprime integers \( q_1, \ldots, q_r \). Then the ring homomorphism

\[
\mathbb{Z}_N \rightarrow \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_r},
\]

\[
x \mod N \mapsto (x \mod q_1, \ldots, x \mod q_r)
\]

is an isomorphism. In other words: given integers \( a_1, \ldots, a_r \), there exists an integer \( x \in \mathbb{Z} \) that solves the congruences

\[
x = a_i \text{ in } \mathbb{Z}_{q_i} \text{ for } 1 \leq i \leq r
\]

simultaneously, and two such solutions \( x \) and \( x' \) differ by a multiple of \( N \), so that \( x = x' \) in \( \mathbb{Z}_N \).
Theorem

RSA works correctly, that is, for every message $x$ the decrypted encrypted message $z$ equals $x$. 