Groups for DH Key Exchange

Consider multiplication modulo 23.

23 is a "safe prime" since 23 = 2*11 +1, where 11 is a prime.

Consider the following cyclic group generated by 2:

Actually, all of 2, 4, 8, 16, 9, 18, 13, 3, 6, 12 are generators and each of them raised to the 11 will be equal to 1 modulo 23.

$2^0 \ mod\ 23$	1
$2^1 \ mod\ 23$	2
$2^2 \ mod\ 23$	4
$2^3 \ mod\ 23$	8
$2^4 \ mod\ 23$	16
$2^5 \ mod\ 23$	$32 \rightarrow 9$
$2^6 \ mod\ 23$	18
$2^7 \ mod\ 23$	$36 \rightarrow 13$
$2^8 \ mod\ 23$	$26 \rightarrow 3$
$2^9 \ mod\ 23$	6
$2^{10} \ mod\ 23$	12
$2^{11} \ mod\ 23$	$24 \rightarrow 1$

Key Agreement

The key-exchange experiment $KE^{eav}{}_{A,\Pi}(n)$:

1. Two parties holding 1^n execute protocol Π. This results in a transcript $trans$ containing all the messages sent by the parties, and a key k output by each of the parties.
2. A uniform bit $b \in \{0,1\}$ is chosen. If $b = 0$ set $\hat{k} := k$, and if $b = 1$ then choose $\hat{k} \in \{0,1\}^n$ uniformly at random.
3. A is given $trans$ and $\hat{k}$, and outputs a bit b'.
4. The output of the experiment is defined to be 1 if $b' = b$ and 0 otherwise.

Definition: A key-exchange protocol Π is secure in the presence of an eavesdropper if for all ppt adversaries A there is a negligible function neg such that

$$\Pr\left[KE^{eav}{}_{A,\Pi}(n) = 1\right] \leq \frac{1}{2} + neg(n).$$

Diffie-Hellman Key Exchange

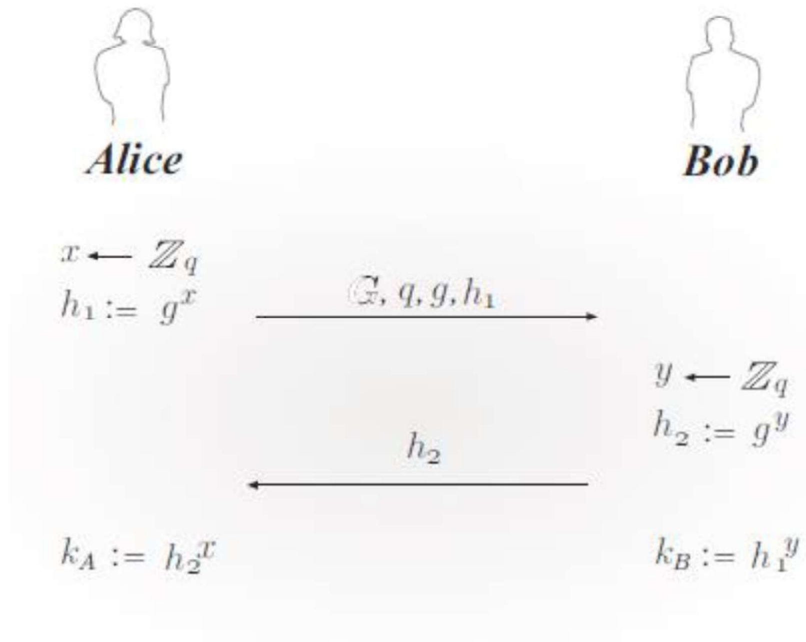


FIGURE 10.2: The Diffie-Hellman key-exchange protocol.

Example for the group we saw above with generator $g = 2$:

Alice:

$x \leftarrow \{0, \dots, 10\}$

Say $x = 8$

$2^8 \; mod \; 23 = 3$

$\xrightarrow{3}$

Bob:

$y \leftarrow \{0, \dots, 10\}$

Say $y = 5$

$2^5 \; mod \; 23 = 9$

$\xleftarrow{9}$

Output: $9^8 \; mod \; 23$

$= 3^{16} \; mod \; 23$

$= 3^{11} \cdot 3^5 \; mod \; 23$

$= 1 \cdot 3^5 \; mod \; 23$

$= 27 \cdot 9 \; mod \; 23$

$= 4 \cdot 9 \; mod \; 23$

$= 36 \; mod \; 23 = \mathbf{13}$

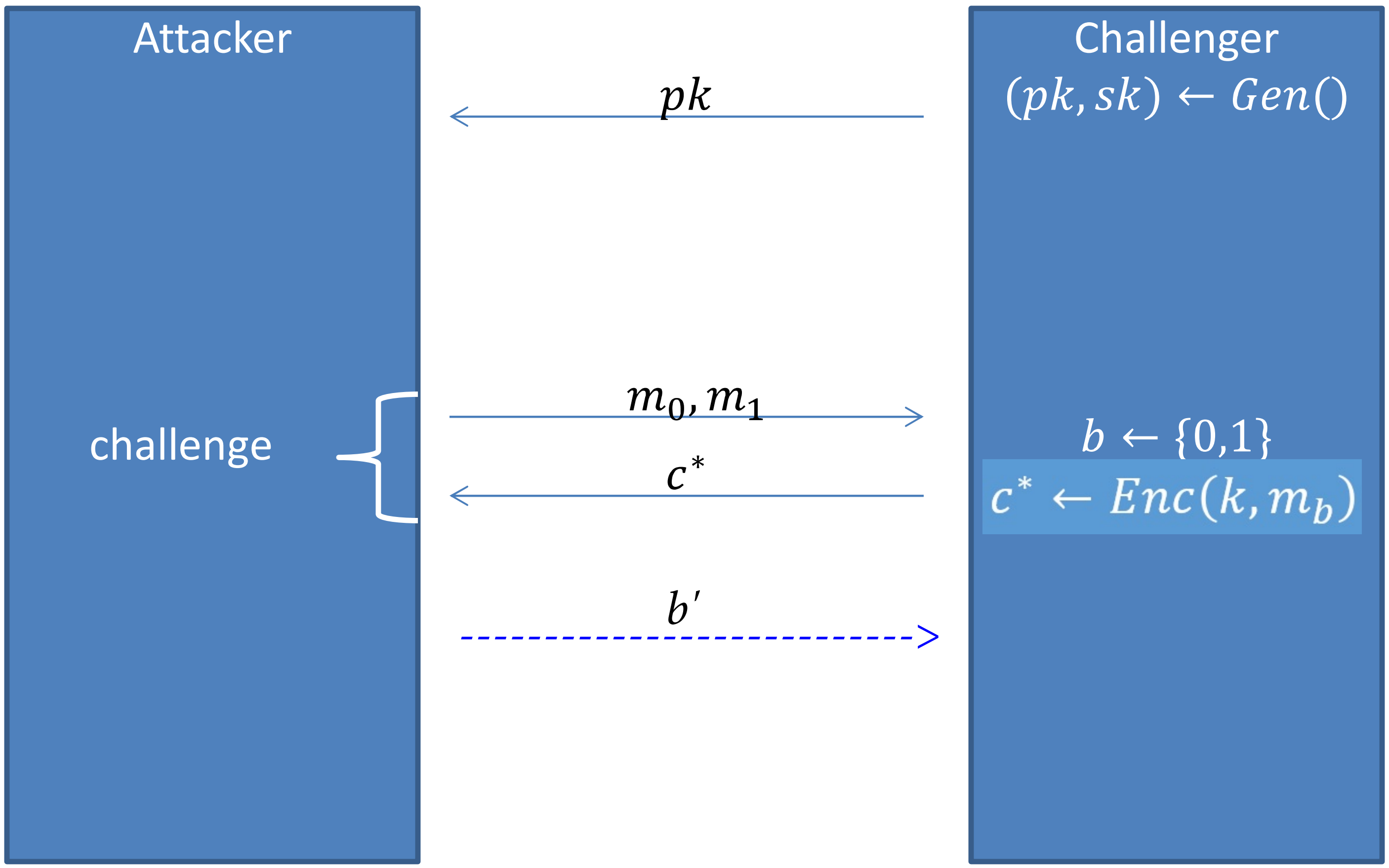
Output: $3^5 \; mod \; 23$

$= 27 \cdot 9 \; mod \; 23$

$= 4 \cdot 9 \; mod \; 23$

$= 36 \; mod \; 23 = \mathbf{13}$

CPA Security for PKE



Attacker "wins" if $b' = b$.

CPA Security: Any efficient attacker wins with probability at most $\frac{1}{2} + negligible$

RSA Encryption

CONSTRUCTION 11.25

Let GenRSA be as in the text. Define a public-key encryption scheme as follows:

- Gen: on input 1^n run GenRSA(1^n) to obtain N, e, and d. The public key is $\langle N, e\rangle$ and the private key is $\langle N, d\rangle$.
- Enc: on input a public key $pk = \langle N, e\rangle$ and a message $m \in \mathbb{Z}_N^*$, compute the ciphertext
$$c := [m^e \bmod N].$$
- Dec: on input a private key $sk = \langle N, d\rangle$ and a ciphertext $c \in \mathbb{Z}_N^*$, compute the message
$$m := [c^d \bmod N].$$

The plain RSA encryption scheme.

RSA Example

$$p = 3, q = 7, N = 21$$
$$\phi(N) = 12$$
$$e = 5$$
$$d = 5$$
$$Enc_{(21,5)}(4) = 4^5 \; mod\; 21 = 16 \; mod\; 21$$
$$Dec_{21,5}(16) = 16^5 \; mod\; 21 = 4^5 \cdot 4^5 mod\; 21$$
$$= 16 \cdot 16 \; mod\; 21 = 4$$

Is Plain-RSA Secure?

- It is deterministic so cannot be secure!
- There are also various additional attacks which we will not cover.

Padded RSA

CONSTRUCTION 11.29

Let GenRSA be as before, and let ℓ be a function with $\ell(n) \leq 2n - 4$ for all n. Define a public-key encryption scheme as follows:

- Gen: on input 1^n, run GenRSA(1^n) to obtain (N, e, d). Output the public key $pk = \langle N, e \rangle$, and the private key $sk = \langle N, d \rangle$.
- Enc: on input a public key $pk = \langle N, e \rangle$ and a message $m \in \{0,1\}^{\|N\| - \ell(n) - 2}$, choose a random string $r \leftarrow \{0,1\}^{\ell(n)}$ and interpret $\hat{m} := 1\|r\|m$ as an element of $\mathbb{Z}_N^*$. Output the ciphertext

$$c := [\hat{m}^e \bmod N].$$

- Dec: on input a private key $sk = \langle N, d \rangle$ and a ciphertext $c \in \mathbb{Z}_N^*$, compute

$$\hat{m} := [c^d \bmod N],$$

and output the $\|N\| - \ell(n) - 2$ least-significant bits of $\hat{m}$.

The padded RSA encryption scheme.