Ricordiamo alcune definizioni relative alla crittografia a chiave pubblica. Alice (A) e Bob (B) per scambiarsi informazioni in modo sicuro usano il seguente meccanismo, fasi (1) e (2):

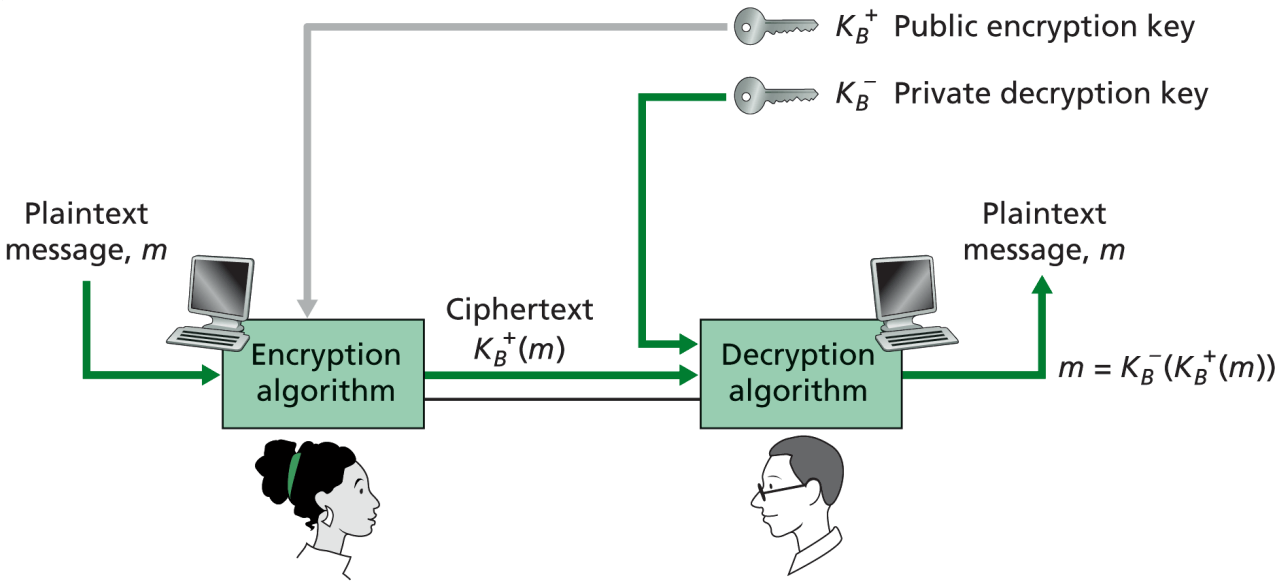


Figure 8.6 ♦ Public key cryptography

Per firmare un documento, Bob estrae, da un messaggio m di lunghezza qualsiasi, l'impronta di lunghezza fissa mediante una funzione di hashing H, ottenendo H(m)

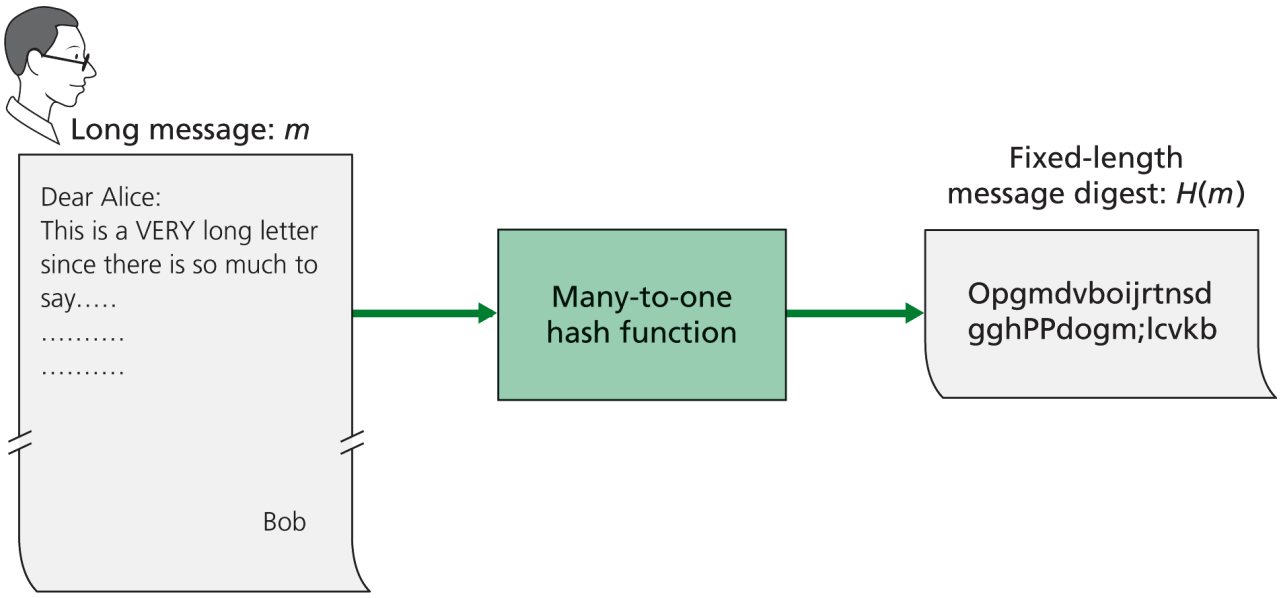


Figure 8.15 ♦ Hash functions are used to create message digests.

Bob applica la propria chiave privata all'impronta di lunghezza fissa H(m) ottenendo la "firma digitale" $K_B^-(H(m))$ e invia ad Alice la firma digitale e il testo in chiaro m, fase (3). La firma digitale gode delle 3 proprietà: deve essere verificabile, non falsificabile e non ripudiabile.

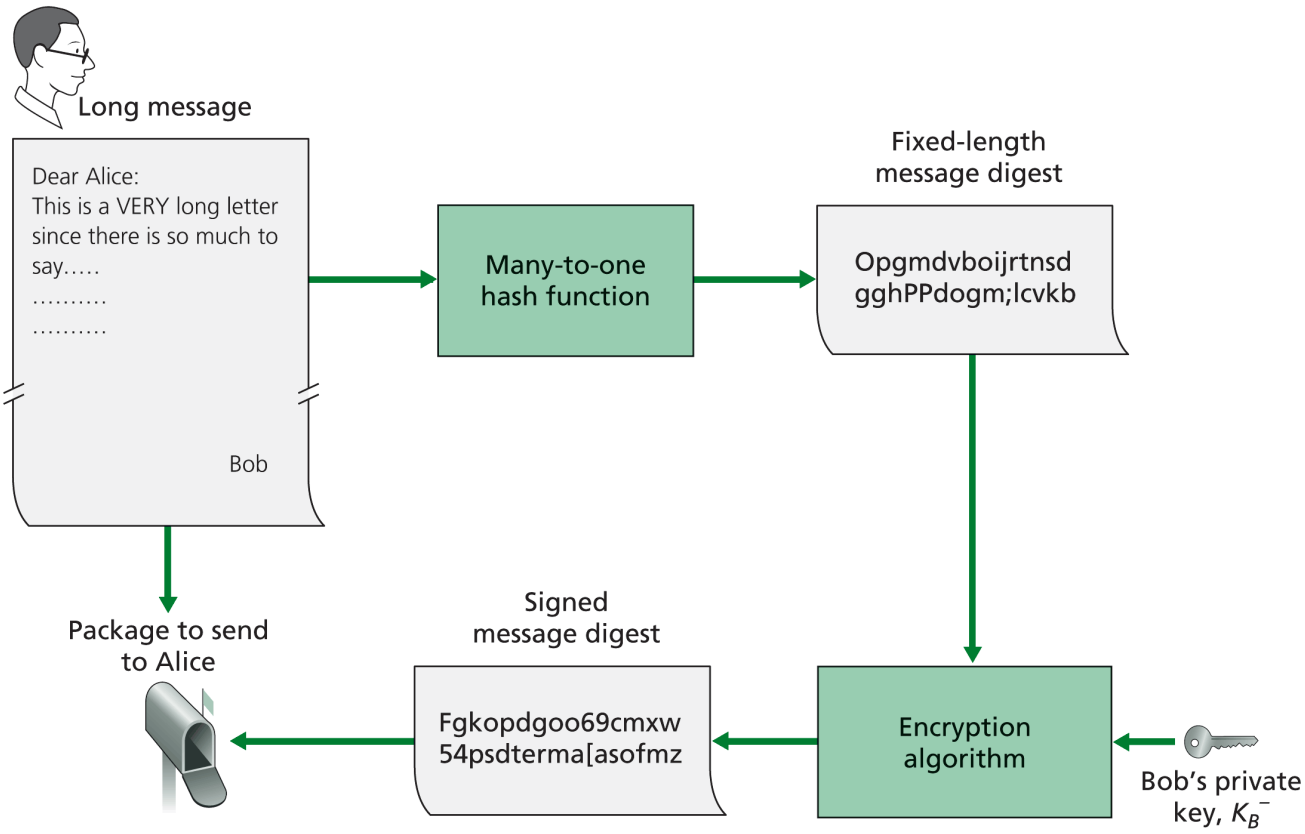


Figure 8.16 ♦ Sending a digitally signed message

Alice ricalcola partendo dalla firma digitale $K_B^-(H(m))$ e applicando la chiave pubblica di Bob K_B^+ l'impronta di lunghezza fissa. Poi ricalcola partendo da m e usando la funzione di hashing H una nuova impronta di lunghezza fissa. Se le due impronte coincidono, Alice ha verificato l'integrità del messaggio inviato da Bob, fase (4).

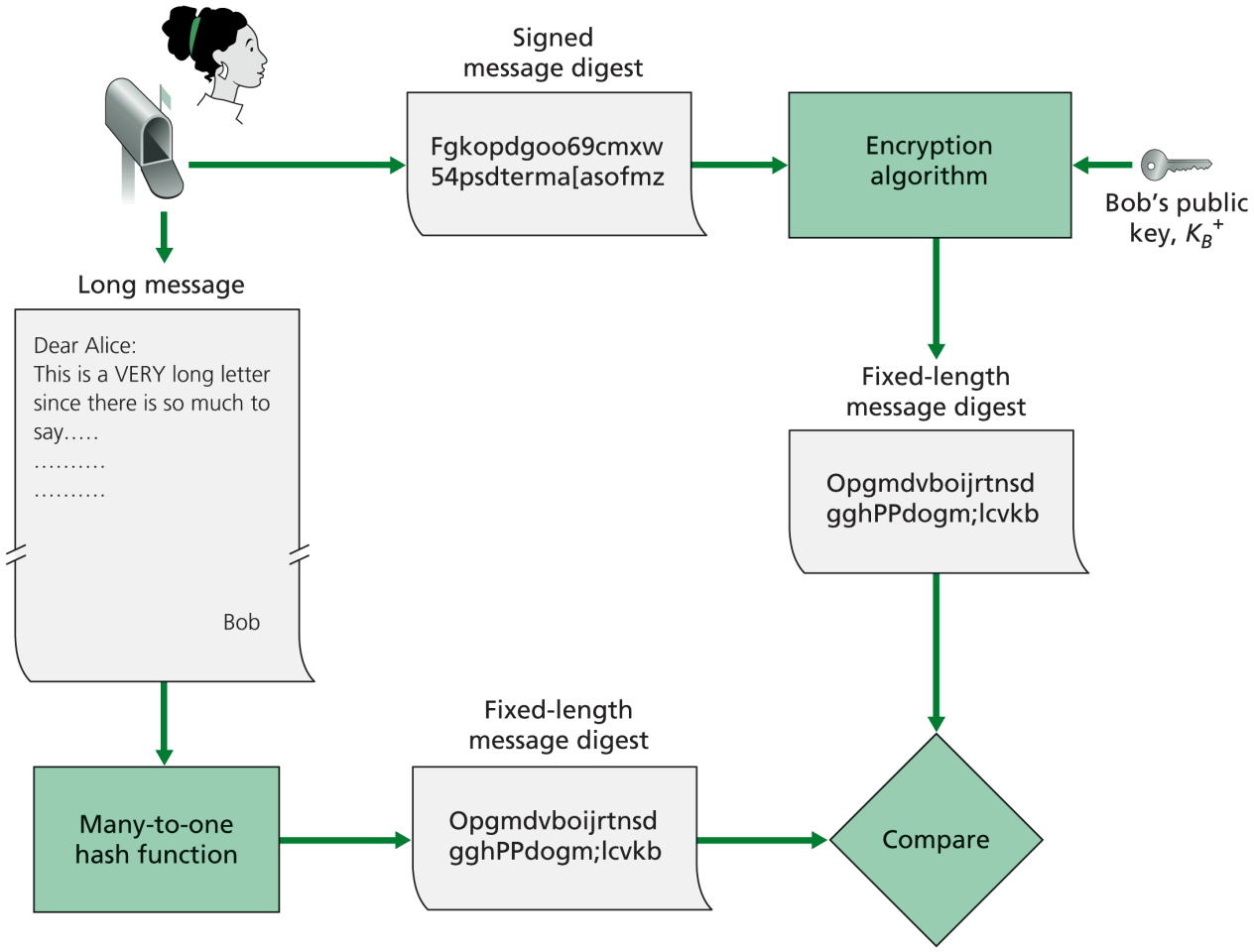
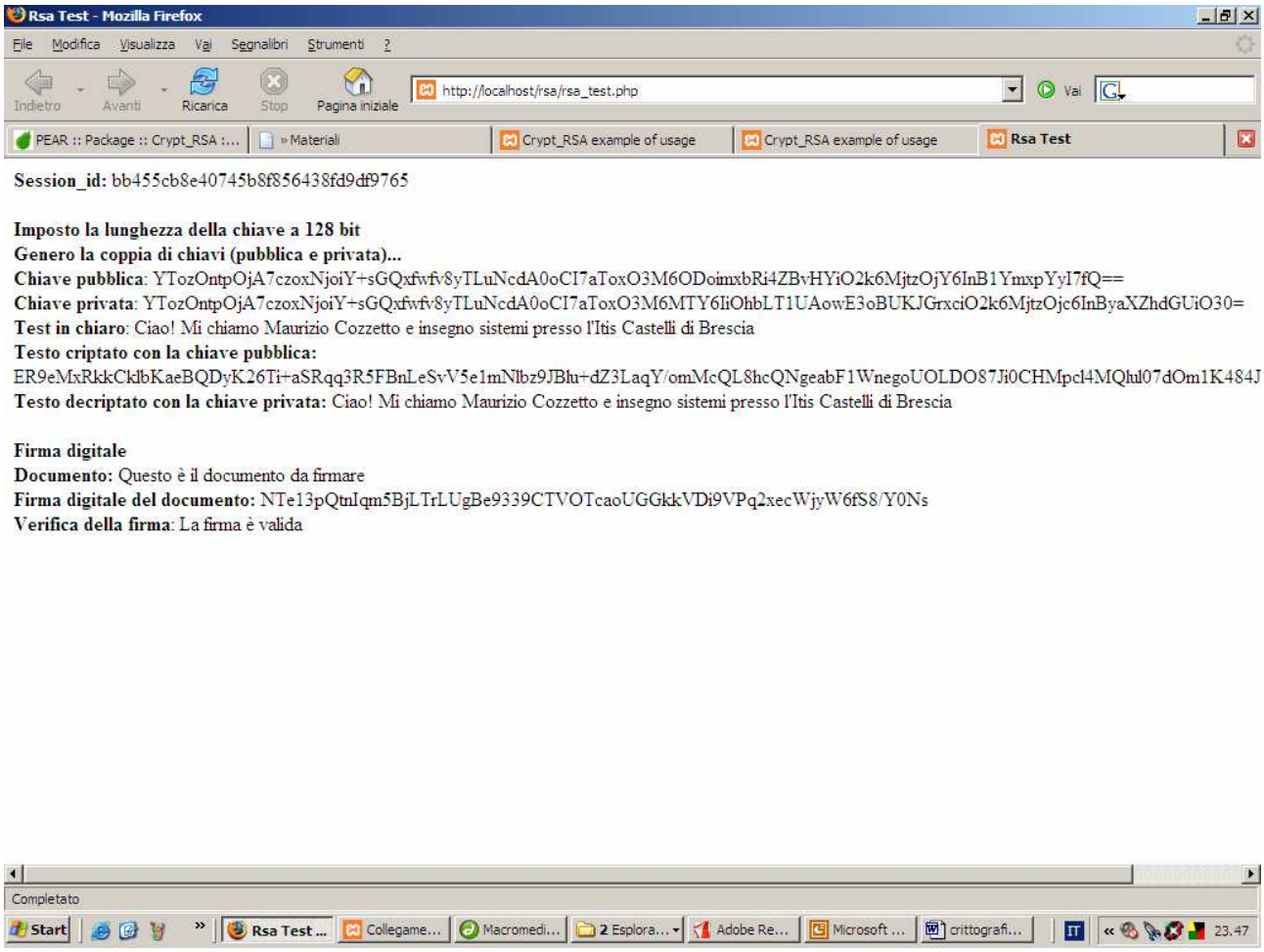


Figure 8.17 ♦ Verifing the integrity of a signed message

Scriviamo un programmino in php che, utilizzando il Package Crypt_RSA 1.0.0 (stabile) contenuto nel repository PEAR (http://pear.php.net/package/Crypt_RSA/download/1.0.0), implementa questi concetti.
Questo è il risultato.



Questo è il file rsa_test.php che sfrutta le classi messe a disposizione del programmatore dal package Crypt_RSA

```
<?php
	session_start();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Rsa Test</title>
</head>

<body>
<?php
	//uso le classi del Package Crypt_RSA
	require_once 'Crypt/RSA.php';

	//Imposto la chiave di lunghezza pari a 128 bit. Possiamo avere lunghezza pari a 32, 64,
	// 128, 256, 512, 1024  2048 bit
	$key_length=128;

	//Le chiavi calcolate hanno effetto solo per la sessione corrente
	echo "<b>Session_id:</b> ".session_id()."<br/><br/>";
	echo "<b>Imposto la lunghezza della chiave a ".$key_length." bit</b><br/>";

	echo "<b>Genero la coppia di chiavi (pubblica e privata)...</b><br/>";

	//Creo la coppia di chiavi pubblica e privata
	$key_pair = new Crypt_RSA_KeyPair($key_length);

	$public_key = $key_pair->getPublicKey();
	$private_key = $key_pair->getPrivateKey();

	echo "<b>Chiave pubblica</b>: ". $public_key->toString()."<br/>";
	echo "<b>Chiave privata</b>: ". $private_key->toString()."<br/>";

	//Questo è il testo in chiaro da crittografare
	$plain_text="Ciao! Mi chiamo Maurizio Cozzetto e insegno sistemi presso l'Itis Castelli di
	Brescia";

	echo "<b>Testo in chiaro</b>: ".$plain_text."<br/>";

	//Istanzio un oggetto di classe Crypt_RSA
	$rsa_obj = new Crypt_RSA;

	//Cifro il testo usando la chiave pubblica, fase (1)
	$enc_text=$rsa_obj->encrypt($plain_text, $public_key);

	echo "<b>Testo criptato con la chiave pubblica:</b> ".$enc_text."<br/>";

	//Per decifrarlo uso la chiave privata, fase (2)
	$plain_text=$rsa_obj->decrypt($enc_text, $private_key);
```

```
        echo "<b>Testo decriptato con la chiave privata:</b> ".$plain_text."<br/>";

        echo "<br/>";

        echo "<b>Firma digitale</b><br/>";

        //Definisco quale documento firmare
        $document="Questo è il documento da firmare";

        echo "<b>Documento:</b> ".$document."<br/>";

        //Firmo il documento con la chiave privata (firma digitale), fase (3)
        $signature=$rsa_obj->createSign($document,$private_key);

        echo "<b>Firma digitale del documento:</b> ".$signature."<br/>";

        // Per fare il controllo della firma, prendo la chiave pubblica.
        // Quindi ricalcolo le due impronte e le metto a confronto, fase (4).
        // Se le due firme coincidono, tutto ok
        if ($rsa_obj->validateSign($document, $signature, $public_key))
                $is_sign_valid='valida';
        else
                $is_sign_valid='non valida';

        echo "<b>Verifica della firma</b>: ";
        echo "La firma è " . $is_sign_valid;

?>
</body>
</html>
```