Cryptography and Public Keys Solutions

1. The outfile is attached in the mail.
2. Tom could easily verify that the message came from Dick because Tom had Dick’s public key at that time.

However, when Dick tried to verify that the message came from Tom, he couldn’t because he did not have Tom’s public key with him at the time of reading the message to verify authenticity of origin.

When using the RSA, to ensure that the sign from a sender is valid, to decrypt the entire message, the receiver needs the public key. That is one of the principles of the RSA algorithm.

1. Tom could verify that the message actually came from Dick because he had Dick’s public key. However, the danger lies when the adversary could have pretended to be Dick and sent the message.

The trust command can be used to add the key to a database, and make sure that it is a trusted key, hence removing the problem of authenticity from the original sender. This updates the trust-db automatically (Additional options include full trust, ultimate trust, marginal trust, etc.). We could also just get the authenticity of the key to be verified by the certificate authority, so that there is no confusion. This solves the problem.

1. The earliest public key that Joseph put up was 2005/10/06.

pub 1024D/[F7959C45](http://pgp.mit.edu:11371/pks/lookup?op=get&search=0xA5E5492FF7959C45) 2005-10-06 [Joseph Greenfield <greenfie@usc.edu>](http://pgp.mit.edu:11371/pks/lookup?op=vindex&search=0xA5E5492FF7959C45)

The earliest public key that Clifford put up was 1994/08/03

pub 1024R/[6CFB4BAD](http://pgp.mit.edu:11371/pks/lookup?op=get&search=0xFF1EFC486CFB4BAD) 1994-08-03 [B. Clifford Neuman <bcn@isi.edu>](http://pgp.mit.edu:11371/pks/lookup?op=vindex&search=0xFF1EFC486CFB4BAD)