Asymetric Key Cryptography & Digest

Comparison between properties of Stream ciphers vs. properties of Block ciphers:

	Stream	Block
Encryption	Individual Chars (bits)	Groups of chars (blocks)
Speed	Faster	Slower
Hardware Circuitry	Simpler	More complex
Data Buffering	Limited or none	More space, relative to block size
Error Propagation	Limited Good for noisy channel	Faster Helps assure message integrity

---

Advantages vs. Disadvantages of Stream and Block ciphers:

	Stream	Block
Advantages	• Speed of transformation • Low error propagation	• High diffusion • Immunity to insertion of symbol
Disadvantages	• Low diffusion • Susceptibility to malicious insertions and modifications	• Slowness of encryption • Padding • Error propagation

---

To encrypt a message/data using Asymmetric Key Cryptography, you will use the public key of the receiver. To decrypt this message, the receiver will use her or his private key. We ensure confidentiality because no one besides the receiver of the message can decrypt the message and learn what it means!

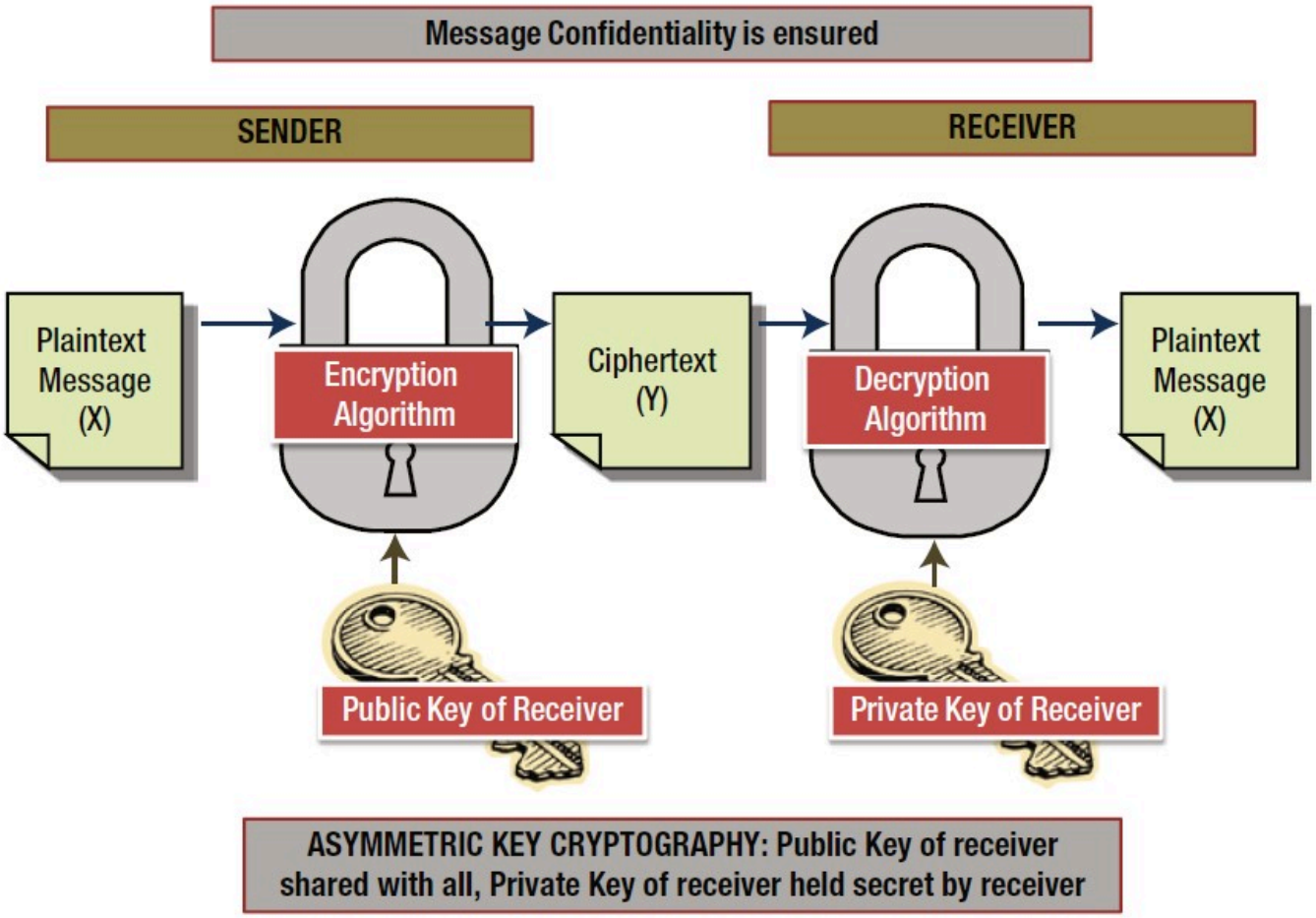
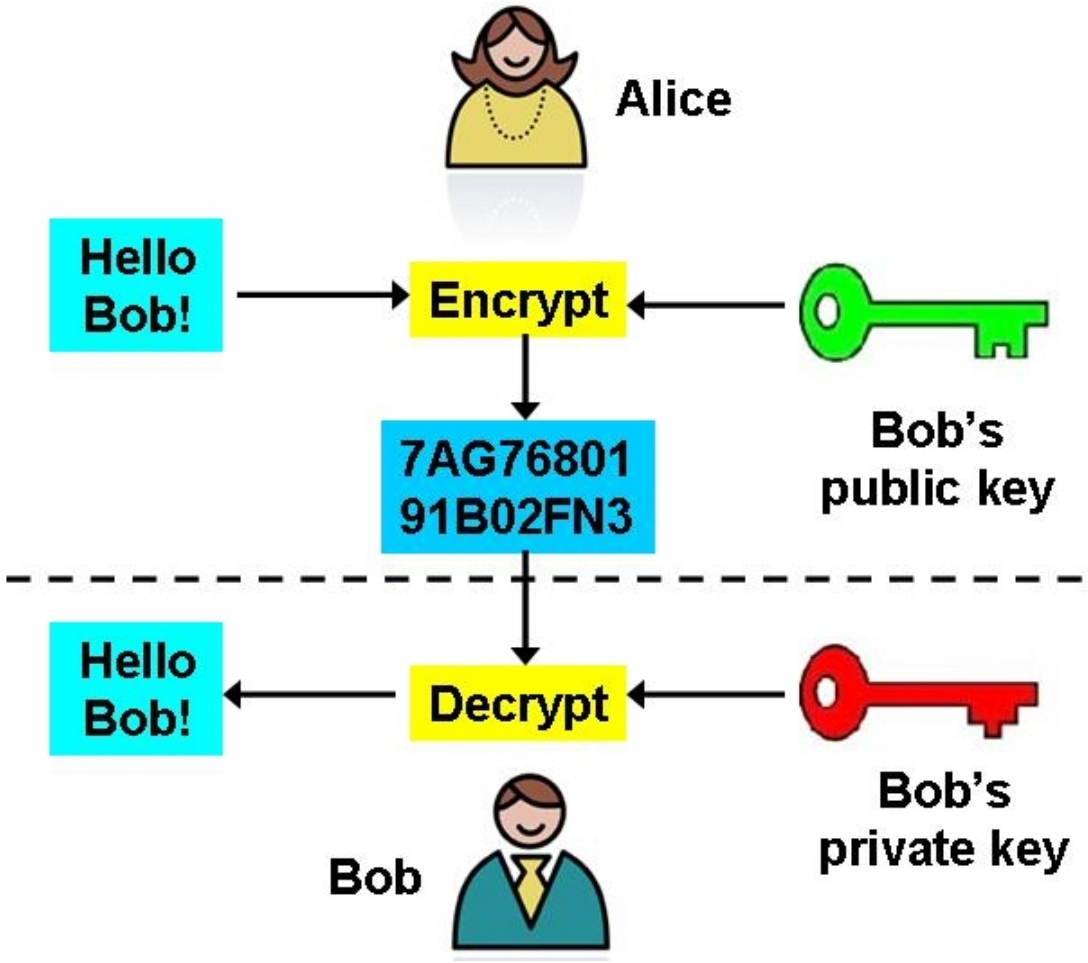
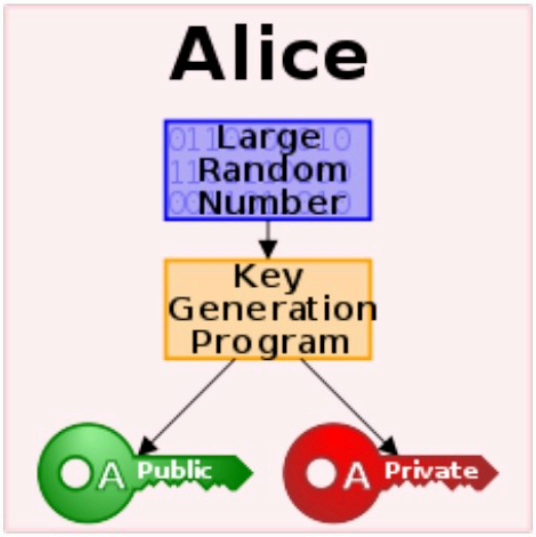


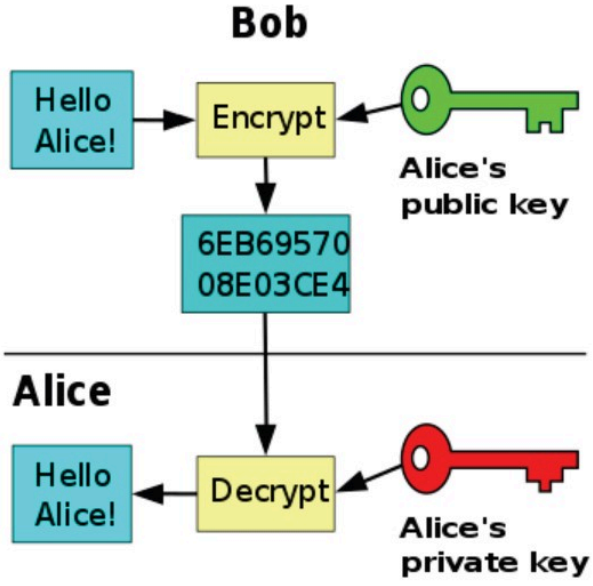
Figure 8-5. *Public Key Cryptography – How Confidentiality is ensured*

An example of using Asymmetric Key Cryptography: Alice sends an encrypted message to Bob.



A couple more diagrams:





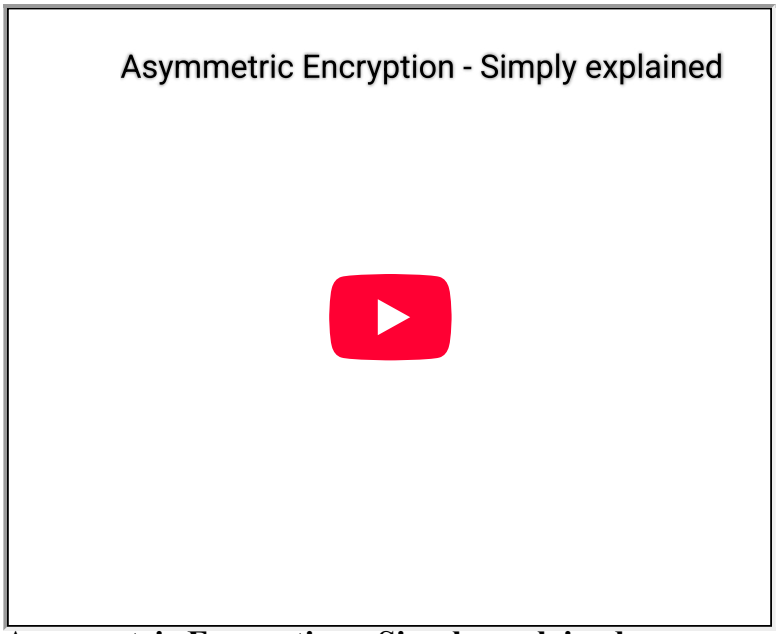
Comparison between the properties of Symmetric and Asymmetric key cryptographies:

	Secret Key (Symmetric)	Public Key (Asymmetric)
Number of keys	1	2
Key size (bits)	56–112 (DES), 128–256 (AES)	Depending on cryptosystem, from 256 to 10000 and more
Protection of key	Must be kept secret	One key must be kept secret; the other can be freely exposed
Best uses	Cryptographic workhorse. Secrecy and integrity of data, from single characters to blocks of data, messages and files	Key exchange, authentication, signing
Key distribution	Must be out-of-band	Public key can be used to distribute other keys
Speed	Fast	Slow, typically by a factor of up to 10,000 times slower than symmetric algorithms

---

Example of the RSA encryption algorithm, which uses a public and a private key and is based on prime numbers: https://en.wikipedia.org/wiki/RSA_(cryptosystem)#Example.

Embedded Video Player: Asymmetric Encryption - Simply explained



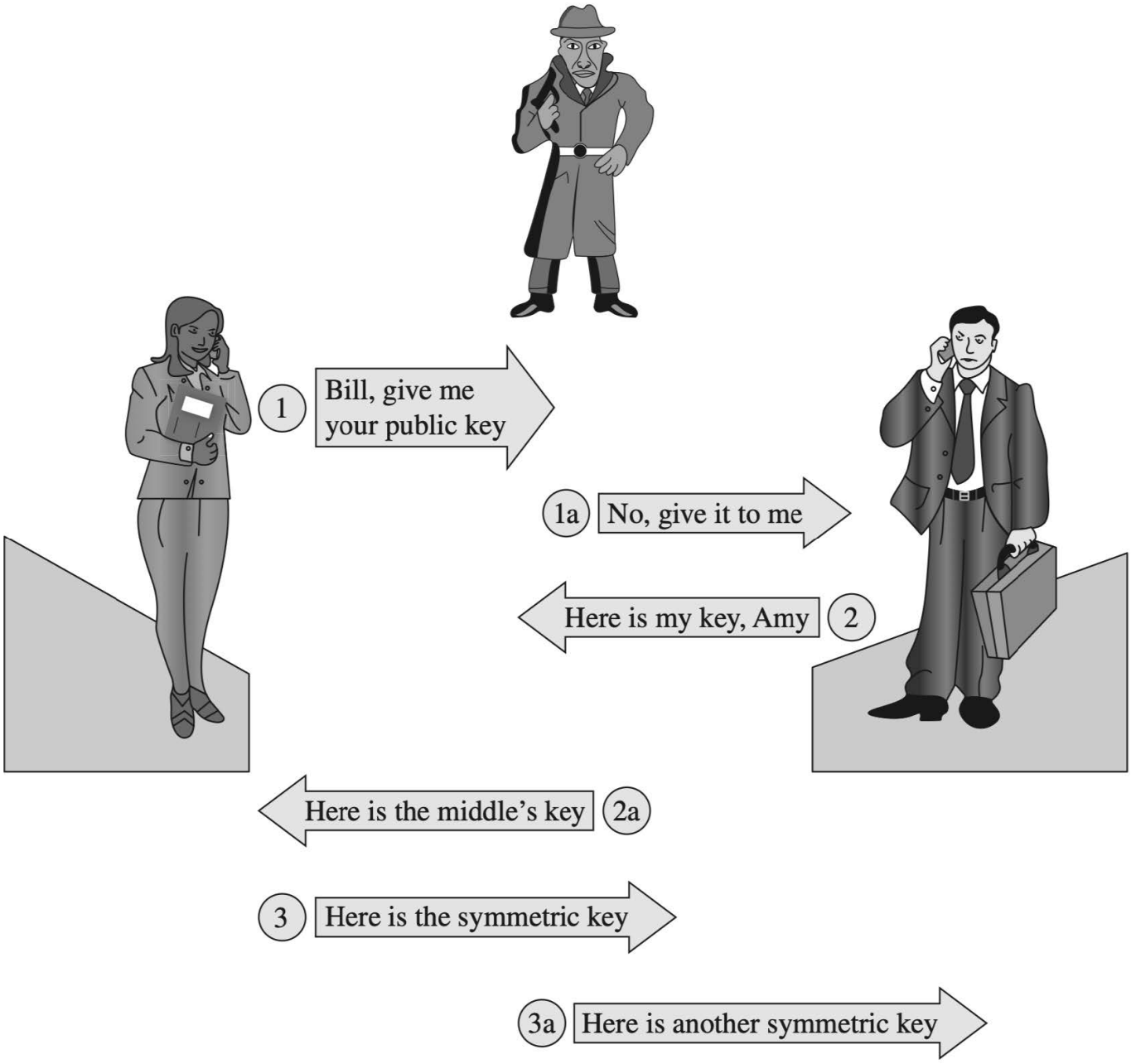
Asymmetric Encryption - Simply explained

User: n/a **- Added:** 10/30/17

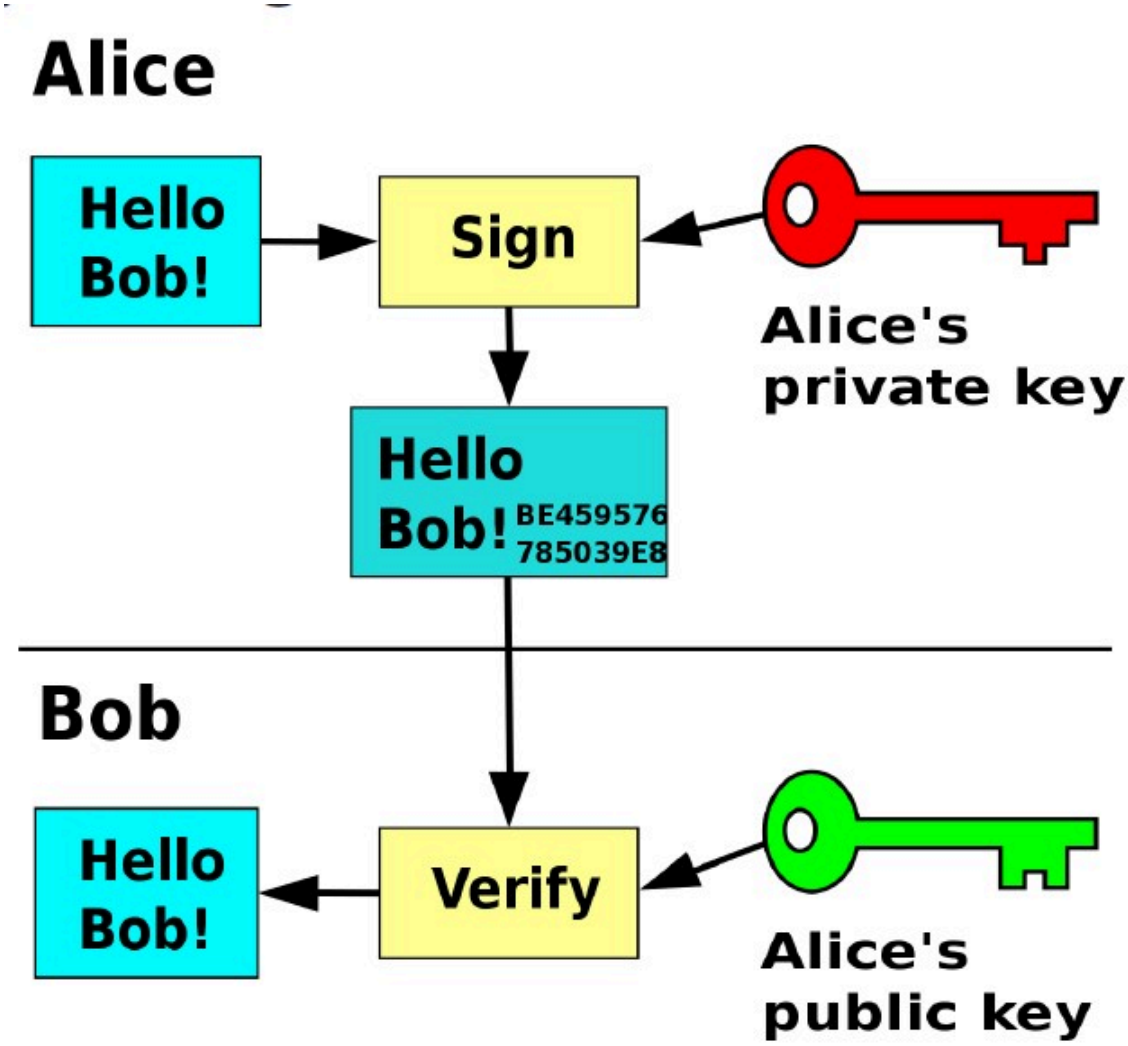
YouTube URL: http://www.youtube.com/watch?v=AQDCe585Lnc

Asymmetric key cryptography ensures the confidentiality of a message. However, it won't tell anything about the identity of the sender of the message. As such, on an insecure channel, a 3rd person ("Man in the

Middle") might send a message to Bob, claiming to be Alice. Bob can't confirm if this message was actually sent by Alice just with Asymmetric key cryptography.
That is, Alice and Bob need to use another mechanism in addition to Asymmetric key cryptography to ensure Authenticity (the ability to identify the sender of a message.) One such tool is Digital Signatures, which can be used to verify the identity of the sender (and, therefore, prevent cases of "Man in the Middle" attacks.



To sign a message/data using Asymmetric Key Cryptography, you will use the private key of the sender (a.k.a., your private key.) To verify that this message was sent by you, the receiver will use your public key. We ensure authenticity because no one besides you has your private key, so if a message was signed using your private key, everyone can verify that this message was sent by you! Verification can be done by anyone because your public key, which is used for the signature verification, is publically-available.



A similar diagram as the above about digital signatures: how messages and sent, and how they are verified. As we mentioned, digital signatures ensure authenticity, so the receiver can confirm that the message arrived from the real sender.

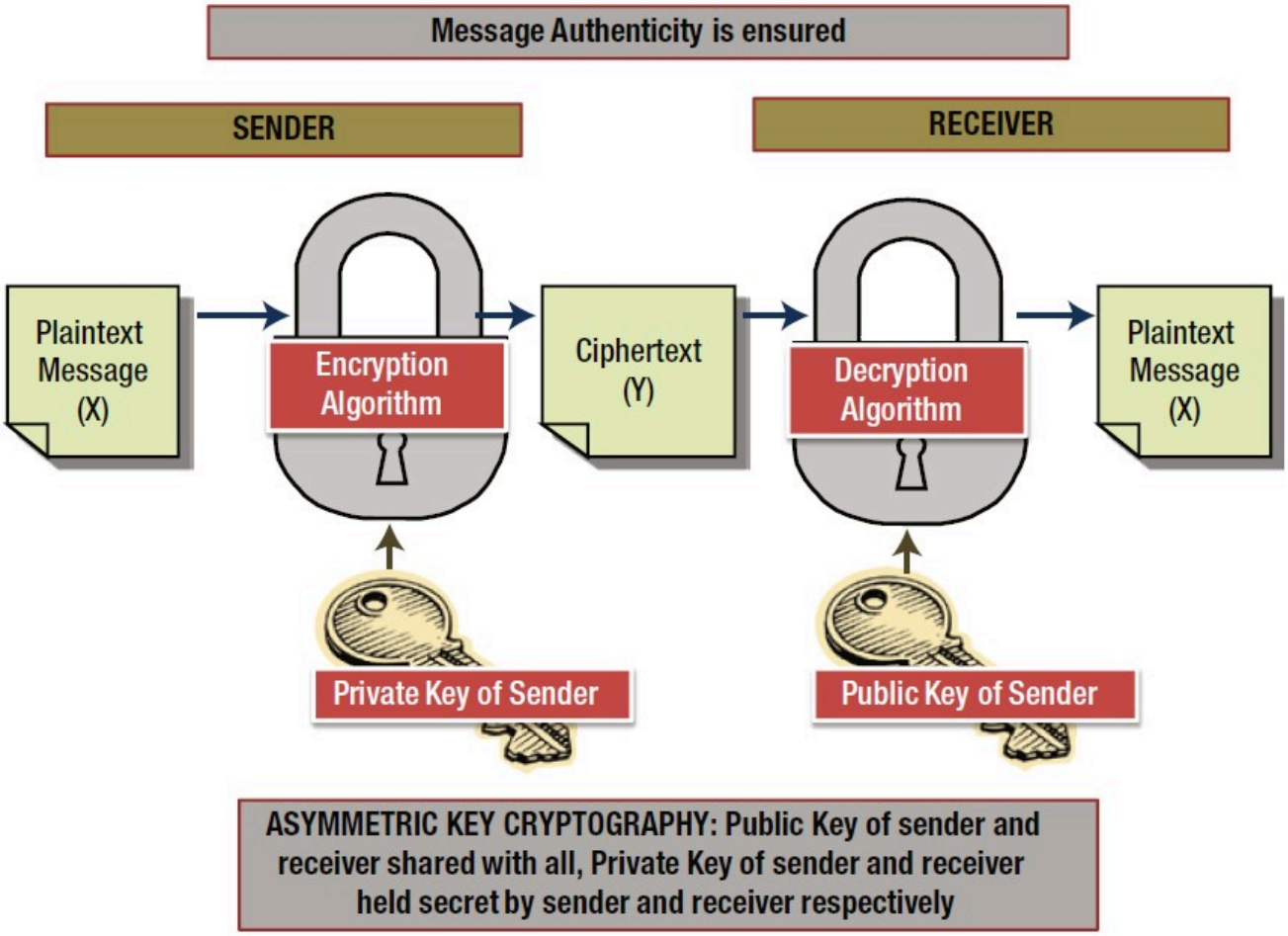
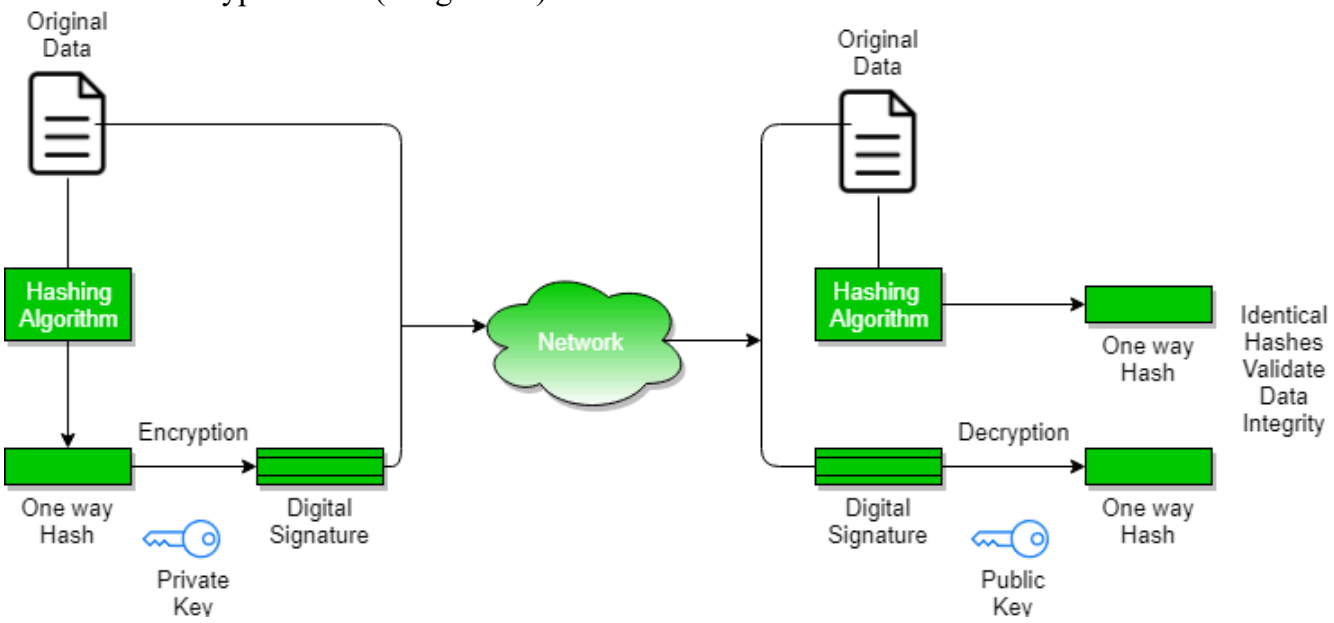


Figure 8-6. *Public Key Cryptography – How Authenticity is ensured*

A digital signature works by hashing the document, and signing the hash value. Then, both the encrypted file and the encrypted hash (= signature) will be sent out.



We can use encryption and digital signatures together: encryption will keep the confidentiality of the message, and the digital signature will be used to verify the sender's identity. One public/private key pair will be used for encryption and decryption, and another pair of keys will be used to sign and then verify the signature.

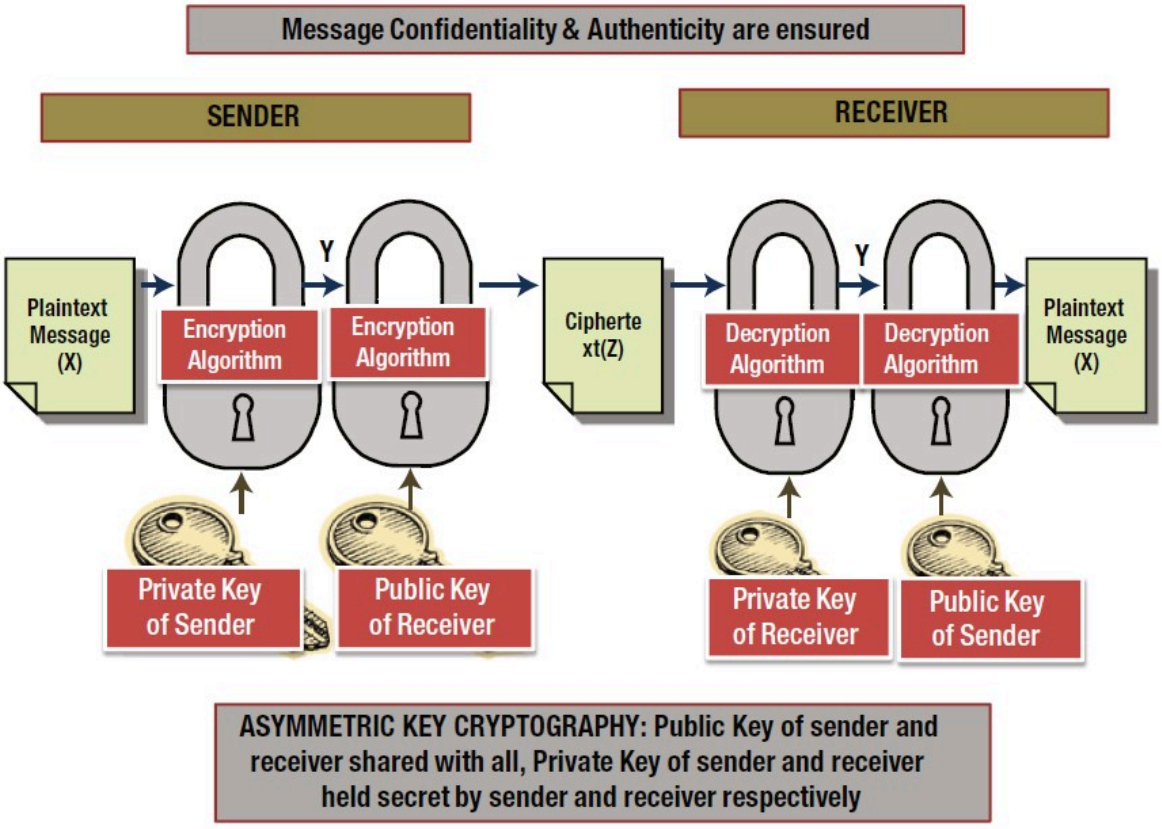


Figure 8-7. *Public Key Cryptography – How both Confidentiality and Authenticity are ensured*

An embedded lecture notes slide on Hash functions. Taken from https://www.sci.brooklyn.cuny.edu/~briskman/cisc/3320/lecture_notes/topic_02/19.html.
See the next slide, Slide 20, as well.

Data Structures

5. **Hash Map**: A data structure that relates (or 'maps') a key with a value, each of which could be a number, string, object, etc.

   A key is mapped to a value by using a **hash function**, which applies a mathematical calculation to the key and returns a numerical value. This value is then used as an index to a hash table, where the key's associated value is stored.

   Because a hash function returns 'almost' unique values, there is a slight chance that two different keys will produce the same hash function result. As such, the hash map might resort to storing the values of the two keys in a list. That is, the hash table is an array of lists.

   The advantage of hash maps is that, when we search for a value based on a given key, we can find it in constant time because the underlying data structure is an array.

On Linux, to find the hash value (= digest) of a string of data using SHA-256 (Secure Hash Algorithm), type:

```
echo -n data | sha256sum
```

where "data" above is replaced by the actual data that you want to hash, e.g.:

```
echo -n hello world! | sha256sum
```

Also, you can use online SHA-256 hash calculators, like this one: https://xorbin.com/tools/sha256-hash-calculator.