Why do we need authentication in the digital world?

In everyday life, we *authenticate* objects or people sometimes even without even realizing it. For example when we pay with a debit or credit card we give evidence that we are the genuine holder of a bank account that holds enough money to achieve a purchase.

In general, in order to establish trust we must be able to recognize that a person is the one he or she pretends to be before engaging in a transaction. The same may happen with objects: we want to verify that a bank note is genuine before recognizing that it is valid for payment or before buying a luxury good we want to be certain it is not counterfeit.

The principle of authentication applies to the digital world: we want to verify that an entity is genuine before we engage into a transaction, for example verifying that the website of my bank is not a mock-up before entering credentials such as account number, password or PIN code.

Principles

Password based authentication

Let's assume Alice wants to verify Bob's identity. The most straightforward mean that comes to mind is that Alice and Bob agree on a password and each time Bob wants to prove his identity he sends the password to Alice.

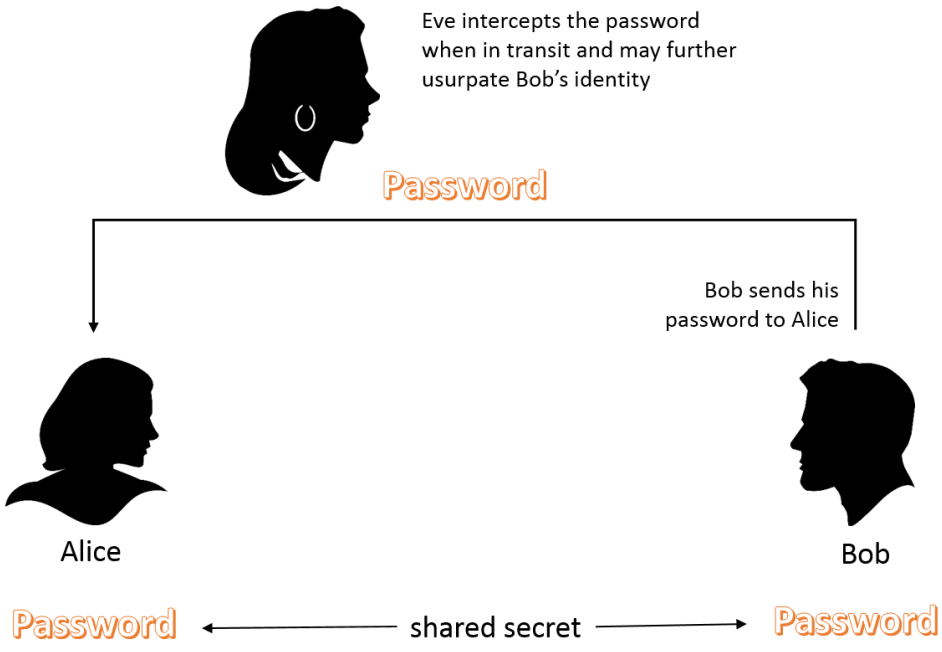

Figure 1. Password based authentication

Copyright © 2020 Thales DIS Design Services

As shown in the illustration, unfortunately this method allows Eve to get the password through eavesdropping, hence it is considered as weak authentication. Fortunately there are better solutions.

Crypto strong authentication methods

Rather than using weak password authentication we can rely on cryptographic methods.

There are two families of cryptographic authentication methods:

- MAC (Message Authentication Codes) based on shared secrets
- Digital signatures based on private and public key pairs or asymmetric cryptography

Authentication based on MAC / shared secrets

Cryptographers have establish trustable methods, so that sending party can append an authentication code and receiving party can verify it. Such authentication codes are named Message Authentication Codes (MACs). Let's say that Alice sends a message to Bob and Bob wants to make sure that Alice is the author of the message. In this scheme Alice and Bob have previously shared a secret. This secret is used as a key for computing MACs.

Rather than sending the key explicitly Bob sends a random number also called a *challenge* to Alice, Alice then computes a MAC - the response - which is a function of the challenge and the secret key. Bob runs the same computation using the same key and then compares his result with Alice's. If they match, the verification is successful and Alice has proven she holds the key and thus has demonstrated she is actually Alice. This process is called **challenge-response authentication**. Because challenge response authentication is not subject to eavesdropping it is much safer than password.

Copyright © 2020 Thales DIS Design Services SAS

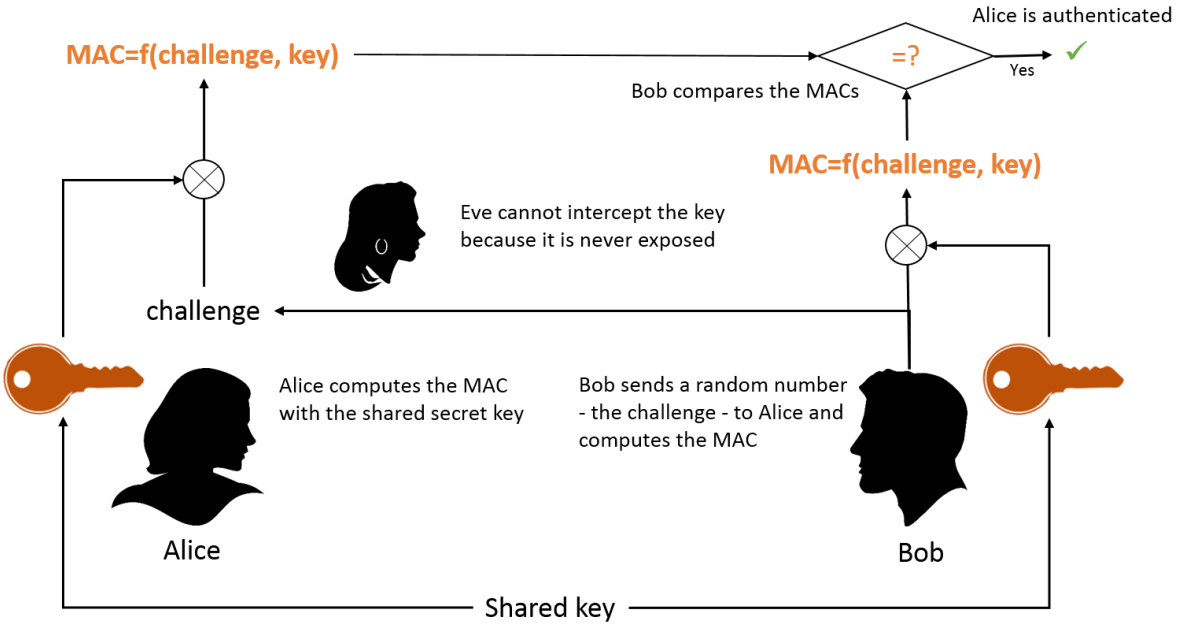

Figure 2. Shared secret based authentication

To make the challenge-response authentication strong, the function *f* used to compute the response *f(challenge, key)* must be such that it is mathematically infeasible to retrieve the key. HMAC meets this requirement. It is an authentication scheme standardized by the NIST [2] and based on the Secure Hash Algorithm (SHA) specified in [1].

Authentication based on private public key pair

The challenge-response authentication method using shared secrets as described above is mathematically strong. Also computation of SHA on short messages is pretty fast and this algorithm can be efficiently implemented in hardware or software at a reasonable cost in terms of kbytes or kgates. This method has a drawback though: it fundamentally relies on the capability to share secrets and keep them protected. While we can assume it would be reasonably easy for Alice and Bob to share a secret, it is much more challenging if Alice needs to be authenticated by a large number of users: in this case Alice would need to share her secret key through a secure channel with many users which is even more difficult if said users are geographically spread across the globe.

As an alternative to a shared secret based system, a challenge response scheme based on a private public key pair also known as asymmetric cryptography can be used. In this configuration, Alice generates a private key that she keeps secret and a public key that she can diffuse widely.

Copyright © 2020 Thales DIS Design Services SAS

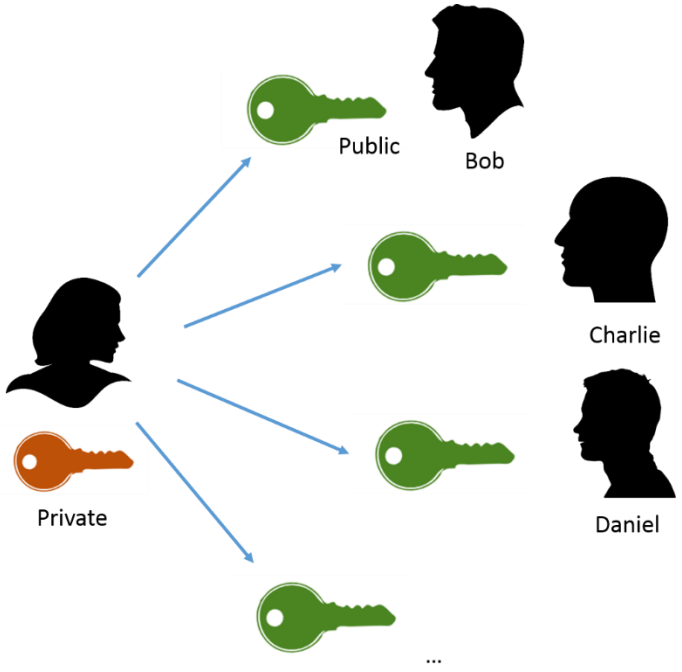

Figure 3. A public key can be easily distributed to several users

In asymmetric cryptography, the private key is used to compute signatures and the public key enables their verification. Because only Alice keeps the private key protected only she can sign (privileged operation) while anybody holding the public key can verify the signature and thus authenticate her (unprivileged operation).

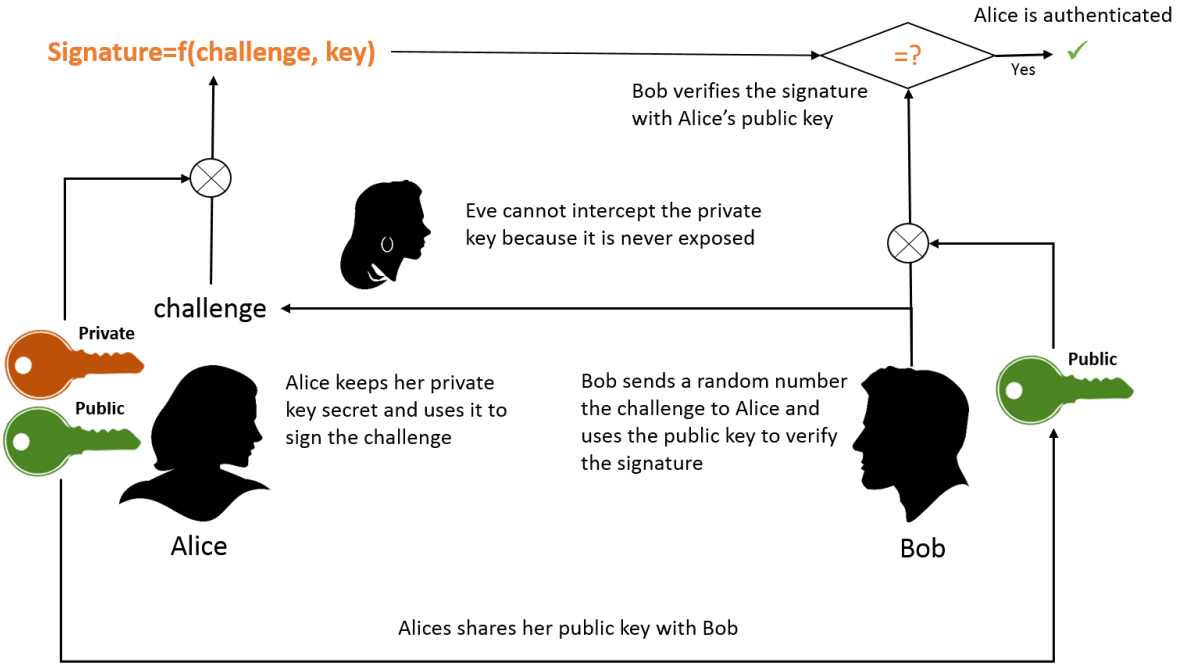

Figure 4. Challenge-response authentication based on digital signature

In the digital world, cryptographers have established algorithms allowing this asymmetric transaction such as DSA and ECDSA. [3] These algorithms are such that:

Copyright © 2020 Thales DIS Design Services SAS

- It is possible to verify a signature with a public key if and only if it was computed by the matching private key
- It is impossible to forge a signature without knowing the private key
- Knowing the public key it is practically infeasible to retrieve or compute the matching private key

Certification authorities and Public Key Infrastructures (PKI)

As explained above, easy key distribution is the main benefit of asymmetric cryptography.

However, while a public key as its name indicates can be freely distributed, there is still a potential weakness as a public key could be replaced. If no precautions are taken, Eve could maliciously substitute Alice's public key previously given to Bob with her own public key and further sign challenges with her private key. Bob would successfully verify the signature using Eve's public key and thus believe the challenge was signed by Eve because he does not know he is using the wrong public key. It is interesting to notice that while a public key can be disclosed it still needs to be trustable and thus require some protection.

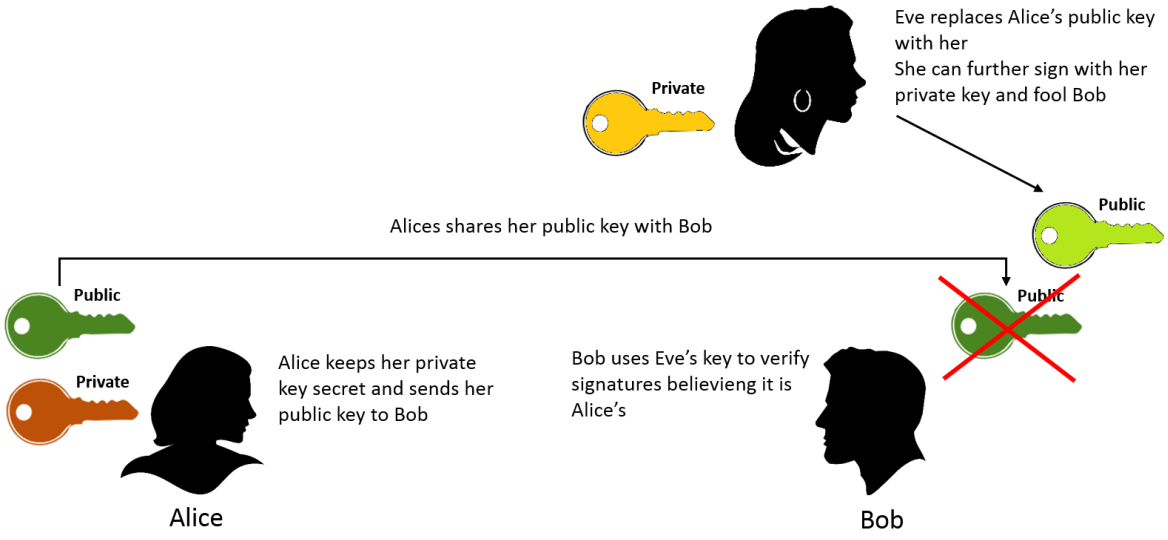

Figure 5. Public key substitution

To avoid Eve misleading Bob, we need to find a solution to attest the origin of a public key. This can sometimes be done manually through an alternate verification media such as phone or e-mail. However this process is manual and cannot be applied in all situations, the preferred way is using a Public Key Infrastructure (PKI).

The goal of a PKI is to bind public keys with their respective owner's identity. To do so, it establishes *certificates.* A certificate contains the public key and a set of information such as owner's name, validity dates and relevant crypto algorithm. The certificate is signed by a so called Certification Authority (CA), thus the signature can be further verified by the CA's public key.

Copyright © 2020 Thales DIS Design Services SAS

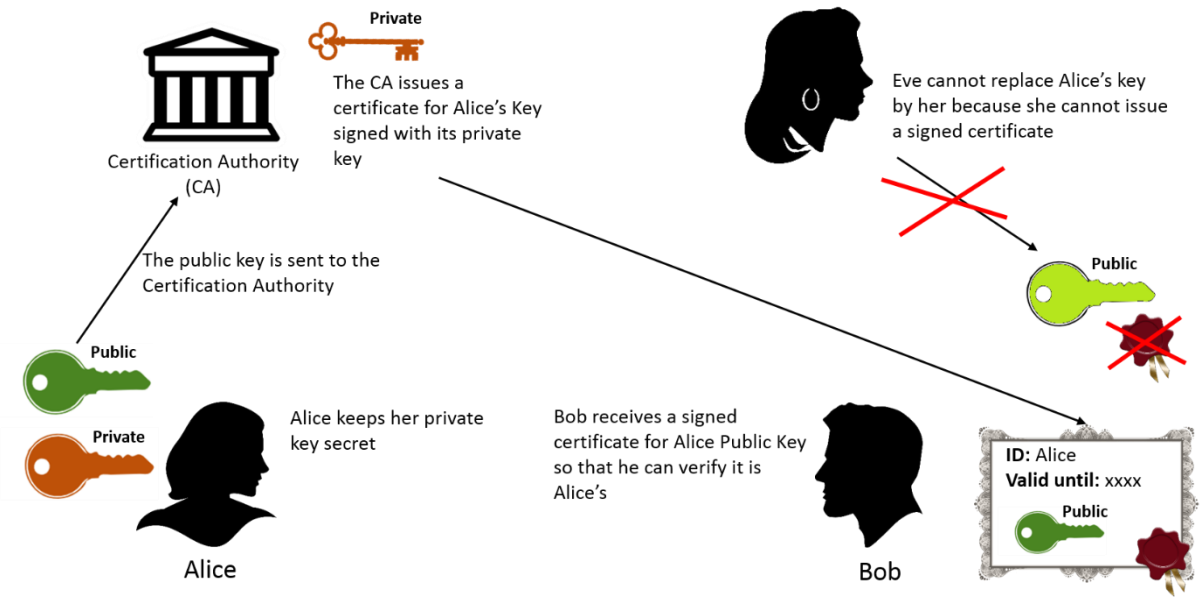

Figure 6. Simplified diagram of a Public Key Infrastructure

Conclusion

While security is often associated with confidentiality, authentication is essential as the prelude to any trusted digital transaction.

Crypto strong authentication methods are well proven and can be implemented based on shared secret or asymmetric algorithms.

Invia offers

- Software libraries
  - HMAC-SHA
  - RSA
  - RSA key generator
  - ECC / ECDSA
  - DRBG
- Synthetizable hardware silicon IPs
  - SHA-256, 384 and 512 coprocessor
  - Public Key Cryptographic Coprocessor supporting
    - RSA
    - ECDSA
- Analog IPs
  - AIS31 Certified and NIST SP800-90 True Random Number generator
  - Digital post processing IP available

Copyright © 2020 Thales DIS Design Services SAS

Contact us at sales@invia.fr

References

[1] https://web.archive.org/web/20150905053839/http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf, "FIPS PUB 180-4 Secure Hash Standard," August 2015. [Online]. Available: https://web.archive.org/web/20150905053839/http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf.

[2] National Institute of Standards and Technology, "FIPS PUB 198-1," July 2018. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.198-1.pdf.

[3] National Institute of Standards and Technology, "FIPS PUB 186-4 Digital Signature Standard (DSS)," July 2013. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf.

Copyright © 2020 Thales DIS Design Services SAS