# Public / Private Key Cryptography

13.07.2025 12:08:52

**FAQ-Artikel-Ausdruck**

| | | | |
|---|---|---|---|
| **Kategorie:** | RRZE: Glossary | **Bewertungen:** | 0 |
| **Status:** | öffentlich (Alle) | **Ergebnis:** | 0.00 % |
| **Sprache:** | en | **Letzte Aktualisierung:** | 14:00:12 - 08.07.2010 |

***Schlüsselwörter***

Identity, Security

***Symptom (öffentlich)***

***Problem (öffentlich)***

***Lösung (öffentlich)***

A mechanism of encrypting and decrypting data using two different but related keys - hence they are termed asymmetric. While the two keys are related, one cannot be computed from the other. Something encrypted with one key can only be decrypted with the other. One key is kept secret - the private key - and the other can be given to anyone - the public key. With this system, any two people can communicate securely after exchanging their public keys. Each recipient uses the corresponding private key to decrypt the session key. In a symmetric system the keys used to encrypt and decrypt are very closely related and this \u2018relationship\u2019 is then shared between two parties, or there may only be a single key \u2013 hence they are termed symmetric. In a hybrid system, a much briefer session key may generated by one party and encrypted by each recipient's public key; each recipient uses the corresponding private key to decrypt the session key and once all parties have obtained the single shared key that was encrypted by each recipient\u2019s public key, they can use a much faster temporary symmetric algorithm to encrypt and decrypt messages. Digital signatures can be susceptible to a \u201cbirthday attack\u201d, as follows. A message m is typically signed by computing f(m), where f is a cryptographic hash function, and then using some secret key to sign it. To trick an identity Y into signing a fraudulent contract, the originator X prepares both a fair contract m and a fraudulent one m'. X then finds a number of positions where m can be changed without changing the meaning, such as inserting commas, empty lines, one versus two spaces after a sentence, replacing synonyms, etc. By combining these changes, X can create a huge number of variations on m which are all fair contracts. In a similar manner, X also creates a huge number of variations on the fraudulent contract m'. X then applies the hash function to all these variations until a version of the fair contract and a version of the fraudulent contract is found to have the same hash value, or f(m) = f(m'). X then presents the fair version to Y for signing. After Y has signed, X takes the signature and attaches it to the fraudulent contract; this signature then "proves" that Y signed the fraudulent contract. To avoid this attack, the output length of the hash function used for a signature scheme can be chosen large enough so that the birthday attack becomes computationally infeasible, i.e. about twice as large as needed to prevent an ordinary brute-force attack. It has also been recommended that Y cosmetically modify any contract presented to him before signing; however, this may not solve the problem, because now X may suspect Y of attempting to use a birthday attack.

Source: "http://identityaccessman.blogspot.com/2006/08/identity-dictionary.html"