For at beskytte personlig data i databasen vil det være effektivt at benytte salt + hashing på personfølsomme data. Salt er random generet strings som du kan tilføje til f.eks. password, cpr-nummer osv. Herefter bliver det 'saltede password' hashed. Ved brug af hash undgår vi at informationer ikke 'bare' bliver gemt som rå tekst i databasen. Altså at hash funktionen tager noget input og 'mapper' det til et output, men hvis du ændre så meget som ét bogstav ændres hashen fuldstændig, f.eks hvis du benytter md5 hash:

Mathias : **534ba0736a590a84e4140f3457a7a626**
Mathia: **66d52a008a8b4a61a1e75a8506eb20df**
Mathias (hash + salt) : **790f1875f55c4da24cac582a8e16cf4b:1f5e7f2748adabf08629a6312ac3bfdd**

Kryptering er noget man benytter til at gøre data ulæseligt ved brug af en algoritme, som nødvendiggøre en nøgle for at 'låse' indholdet op og gøre det læsbart. F.eks under 2. verdens krig benyttede tyskerne sig af Enigma chiffermaskinen til at kryptere og dekryptere meddelser, som næsten var umulig at knække, men som til sidst blev.

RSA algoritmen var en af de første public-key krypteringssystemer som blev brugt meget, det bliver blandt andet brugt digital signatur, det anvender asymmetrisk krypteringssystem som egentlig betyder at der er to relaterede krypteringsnøgler hvor den ene er public og den anden er private. Hvis nøgle a har krypteret værdier, er det kun nøgle b som kan dekryptere værdierne.

SSL som er Secure Socket Layer er også asynkron krypteringsmetode som gør at din browser og den specfikke hjemmeside ikke kender hinanden. Ved brug af SSL gør det muligt at at browseren kan kryptere noget data til hjemmesiden som kun den kan dekryptere og vice verca.

Ligeledes er der symmetrisk kryptering som modsat assymetrisk kryptering, har den kun én nøgle som både kryptere og dekryptere værdierne. Et eksempel på en symmetrisk kryptering standard er AES, hvor at kender man nøglen som det er krypteret med, kan man bruge den samme nøgle til at dekryptere det med.