Cookery recipe

Secret Sharing scheme

Shamir's Secret Sharing (SSS) is used to secure a secret in a distributed way, most often to secure other encryption keys. The secret is split into multiple parts, called **shares**. These shares are used to reconstruct the original secret.
To unlock the secret via Shamir's secret sharing, a minimum number of shares are needed. This is called the **threshold**, and is used to denote the minimum number of shares needed to unlock the secret. An adversary who discovers any number of shares less than the threshold will not have any additional information about the secured secret-- this is called perfect secrecy. In this sense, SSS is a generalisation of the one-time pad (which is effectively SSS with a two-share threshold and two shares in total).
Let us walk through an example:
Problem: Company XYZ needs to secure their vault's passcode. They could use something standard, such as AES, but what if the holder of the key is unavailable or dies? What if the key is compromised via a malicious hacker or the holder of the key turns rogue, and uses their power over the vault to their benefit?
This is where SSS comes in. It can be used to encrypt the vault's passcode and generate a certain number of shares, where a certain number of shares can be allocated to each executive within Company XYZ. Now, only if they pool their shares can they unlock the vault. The threshold can be appropriately set for the number of executives, so the vault is always able to be accessed by the authorized individuals. Should a share or two fall into the wrong hands, they couldn't open the passcode unless the other executives cooperated.
From <https://en.wikipedia.org/wiki/Shamir%27s_Secret_Sharing>

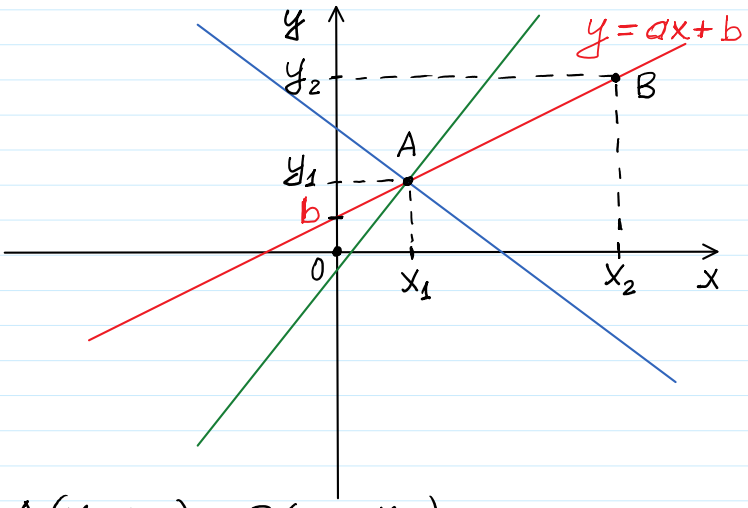
Shamir's Secret Sharing is an ideal and perfect (***t***, ***N***)-**threshold** scheme.
In such a scheme, the aim is to divide a secret ***S*** (for example, the combination to a safe) into ***N*** pieces of data $P_1, P_2, \ldots, P_N$ known as **shares** in such a way that:

1. Knowledge of any ***t*** or more P_i pieces makes ***S*** easily computable. Terefore ***t*** is is named as **threshold** . That is, the complete secret ***S*** can be reconstructed from any combination of ***t*** or more pieces of data.
2. Knowledge of any ***t***-1 or fewer P_i pieces leaves ***S*** completely undetermined, in the sense that the possible values for ***S*** seem as likely as with knowledge of **0** pieces. The secret ***S*** cannot be reconstructed with fewer than ***t*** pieces.

If ***t=N***, then every piece of the original secret is required to reconstruct the secret.

We are considering the field of real numbers $\langle R, +, -, *, : \rangle$.
Then the plain consisting of real numbers is $R^2 = \{(x, y);\ x \in R, y \in R\}$

$y = ax^2 + bx + c$

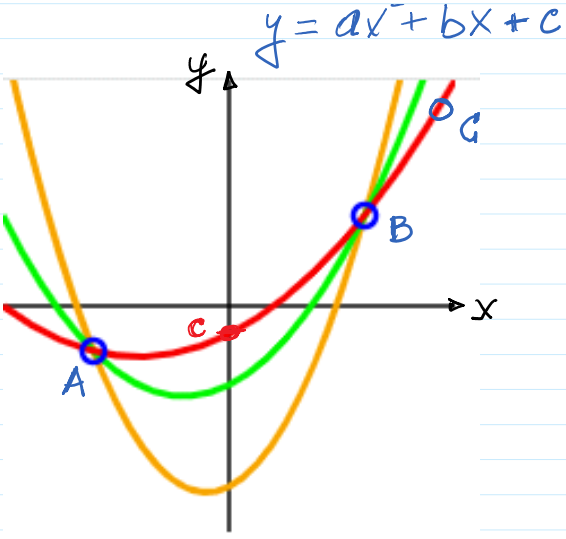


$A(x_1, y_1)$; $B(x_2, y_2)$.

$$\begin{cases} ax_1 + b = y_1 \\ ax_2 + b = y_2 \end{cases}$$

$$a(x_1 - x_2) = y_1 - y_2 \Rightarrow a = \frac{y_1 - y_2}{x_1 - x_2}$$

$$b = y(x = 0) = (ax + b)\Big|_{x=0}$$



One can draw an infinite number of polynomials of degree 2 through 2 points. 3 points are required to define a unique polynomial of degree 2. This image is for illustration purposes only — Shamir's scheme uses polynomials over a finite field, not representable on a 2-dimensional plane.

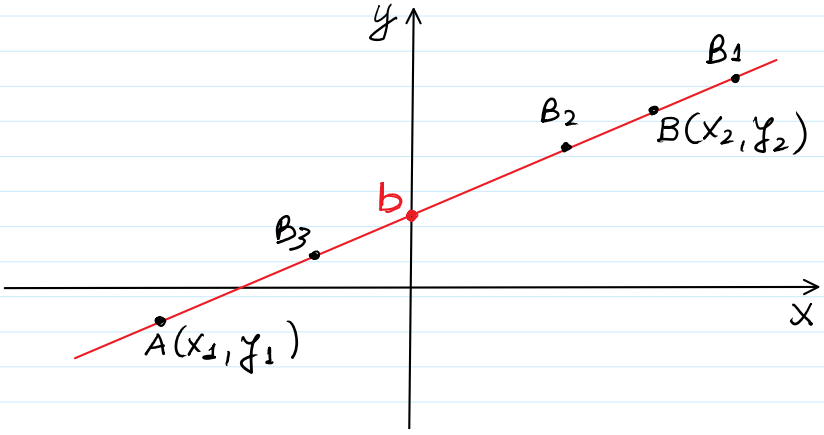
From <https://en.wikipedia.org/wiki/Shamir%27s_Secret_Sharing>

$$y = ax^2 + bx + c$$

$A(x_1, y_1)$; $B(x_2, y_2)$; $C(x_3, y_3)$

By solving this linear system of equation, parabola coefficients a, b, c can be obtained. $\Leftarrow$

$$\begin{cases} ax_1^2 + bx_1 + c = y_1 \\ ax_2^2 + bx_2 + c = y_2 \\ ax_3^2 + bx_3 + c = y_3 \end{cases}$$



$b = S$

Rec : secret recipe

$Enc(S, Rec) = C_{Rec}$

$Dec(S, C_{Rec}) = Rec$

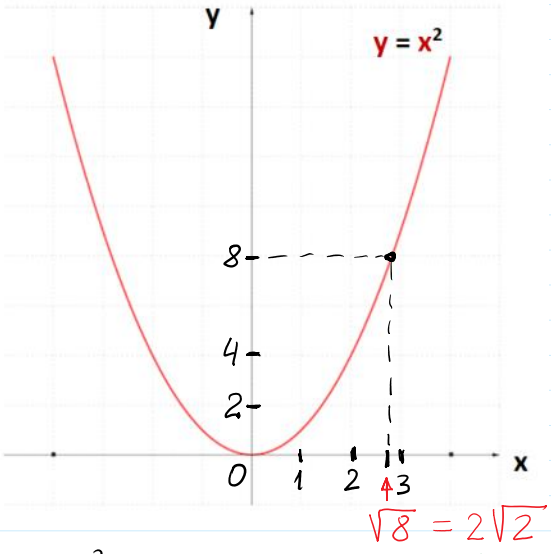
shares : $\{A, B, B_1, B_2, B_3\}$

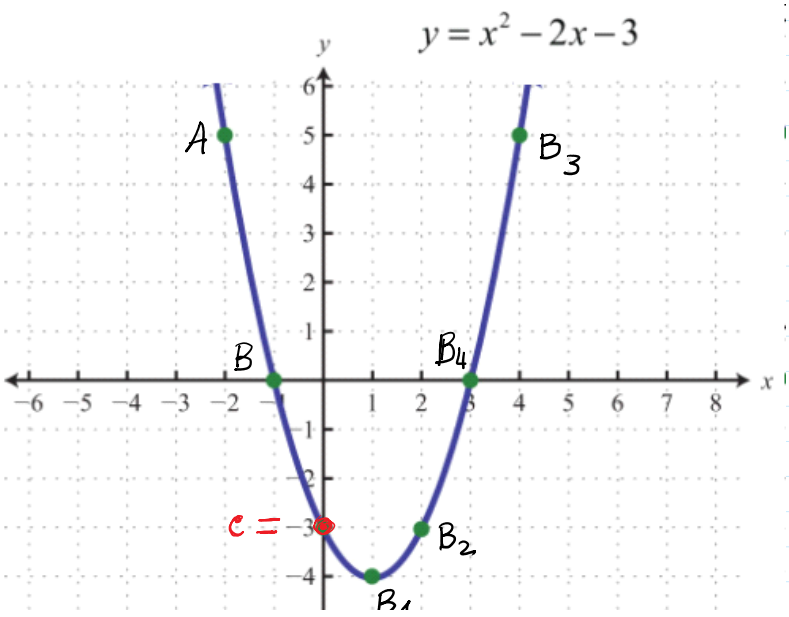
$A(x_1, y_1)$

shares : $\{A, B, B_1, B_2, B_3\}$
In the case of linear interpolation Threshold = 2.

If (A, B) can not participate in recovery secret b, then secret b can be recovered by any other pair (B_1, B_1), (B_1, B_3), (B_2, B_3)
In general, secret b can be recovered by C_5^2 poras
$(A, B), (A, B_1), (A, B_2), \ldots, (B_2, B_3)$ $\quad C_5^2 = \frac{5 \cdot 4}{2} = 10$

But 2-shares could be not enough to protect the secret b.
due to Bribering ....
Let it be 3-share created to protect the secret.
It is required to choose parabola $y = ax + bx + c$
whic can be recovered by Lagrangian interpolation using 3-points
$(A, B, C) \Rightarrow$ threshold $= 3$
A decided to share the secret to 7-parts among
$\{A, B, B_1, B_2, B_3, B_4\}$
The number of triplets to recover secret c is
$C_6^3 = \frac{6 \cdot 5 \cdot 4}{3 \cdot 2 \cdot 1} = 20.$

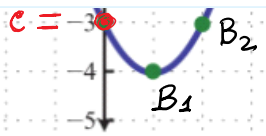




$\begin{cases} ax_1^2 + bx_1 + c = y_1 \end{cases}$

$\sqrt{8} = 2\sqrt{2}$

$$\begin{cases} a x_1^2 + b x_1 + c = y_1 \\ a x_2^2 + b x_2 + c = y_2 \\ a x_3^2 + b x_3 + c = y_3 \end{cases}$$



$t = 3 = n+1 = 2+1.$

$n = t - 1 = 3 - 1 = 2.$

$y = a x^2 + b x + c$

$S = c = -3$

For any $t = n+1$ we must choose n-th degree polynomial

$$y = P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0$$

This polynomial can be recovered by Lagrange interpolation technique having $n+1$ points - uniquely recovered!

$$\{P_1, P_2, \dots, P_{n+1}\} \longleftrightarrow \{(x_1, y_1), (x_2, y_2), \dots, (x_{n+1}, y_{n+1})\}$$

$$\begin{cases} a_n x_1^n + a_{n-1} x_1^{n-1} + \dots + a_0 = y_1 \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ a_n x_{n+1}^n + a_{n-1} x_{n+1}^{n-1} + \dots + a_0 = y_{n+1} \end{cases} \Rightarrow (a_n, a_{n-1}, \dots, a_1, a_0)$$

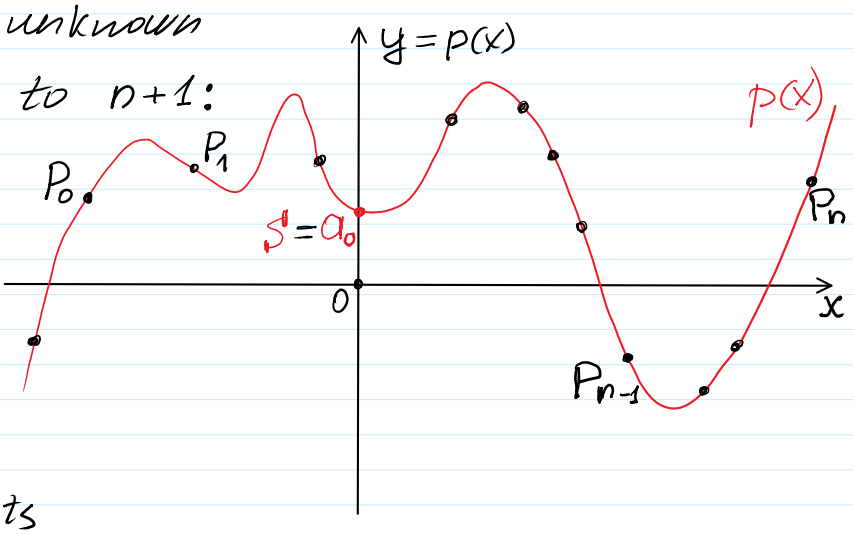
$a_0 \downarrow S$

Let $p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ be a polynomial of order n.

In $p(x)$ the number of unknown coefficients is equal to $n+1$:

to define $p(x)$ it is required to construct $n+1$ linear equations to find coefficients $\{a_0, a_1, \dots, a_{n-1}, a_n\}$.

We must have $(n+1)$ points $\{P_0, P_1, \dots, P_{n-1}, P_n\}$ where $p(x)$ is crossing these poits.



This technique is named *Lagrangian interpolation*: $t-1 = n$.

$$S = a_0 = p(x=0) = \sum_{i=0}^{t-1} y_i \prod_{\substack{j=0 \\ i \neq j}}^{t-1} \frac{x_j}{x_j - x_i}$$

Infinite field R must be replaced by finite field $F_p = \mathcal{L}_p$.

Arithmetic of Finite fields $\mathbf{Z_p} = \mathbf{F_p} = (0, 1, 2, \ldots, \mathbf{p}-1)$, when $\mathbf{p}$ is prime.

$+ \bmod \mathbf{p}$, $- \bmod \mathbf{p}$, $* \bmod \mathbf{p}$, $: \bmod \mathbf{p}$: mod p by "0": $z/0$ – is not defined.

1) $\mathcal{L}_p$ in an additive group: $< \mathcal{L}_p, + \bmod p >$
2) $\mathcal{L}_p$ has multiplicative group $\mathcal{L}_p^* = \{1, 2, 3, \ldots, p-1\}$
   $\mathcal{L}_p^* \subset \mathcal{L}_p$ $< \mathcal{L}_p^*, * \bmod p >$
3) The distributive law takes place in $\mathcal{L}_p$:
   for all $a, b, c \in \mathcal{L}_p$: $a*(b+c) = (a*b + a*c) \bmod p$

$$\prod_{i=1}^{3} a_i = a_1 \circ a_2 \circ a_3$$

$\mathcal{L}_{11} = \{0, 1, 2, \ldots, 10\}$: $+, -, \bullet, :$ mod 11

$$S = a_0 = p(x=0) = \sum_{i=0}^{t-1} y_i \prod_{\substack{j=0 \\ i \neq j}}^{t-1} \frac{x_j}{x_j - x_i} \bmod p$$

$a, b \in \mathcal{L}_p$:

// modop(,)
>> ab = mod(a*b, p)

```
>> p=11
p = 11
>> a=5
a = 5
>> b=9
b = 9
>> ab=mod(a*b,p)
ab = 1
```

$5 \cdot 9 = 45$ | 11
44 | 4
1

$5 - 9 \bmod p = -4 \bmod p =$
$= 0 - 4 \bmod p = 11 - 4 \bmod = 7$

```
>> apb=mod(a+b,p)
apb = 3
```

Division mod p

>> mod(9/5,p)

```
>> apb=mod(a+b,p)
apb = 3

>> amb=mod(a-b,p)
amb = 7
```

Division mod p is not realized in mod(,) function.

```
>> mod(9/5,p)
ans = 1.8000
>> int64(mod(9/5,p))
ans = 2
>> a_m1=mulinv(a,p)
a_m1 = 9
>> mod(a*9,p)
ans = 1
>> bda=mod(b*a_m1,p)
bda = 4
```

Chips cloning technique

Integrated circuits can be compromised using Undetectable hardware Trojans

From <https://thehackernews.com/2013/09/Undetectable-hardware-Trojans.html>

M 1:120

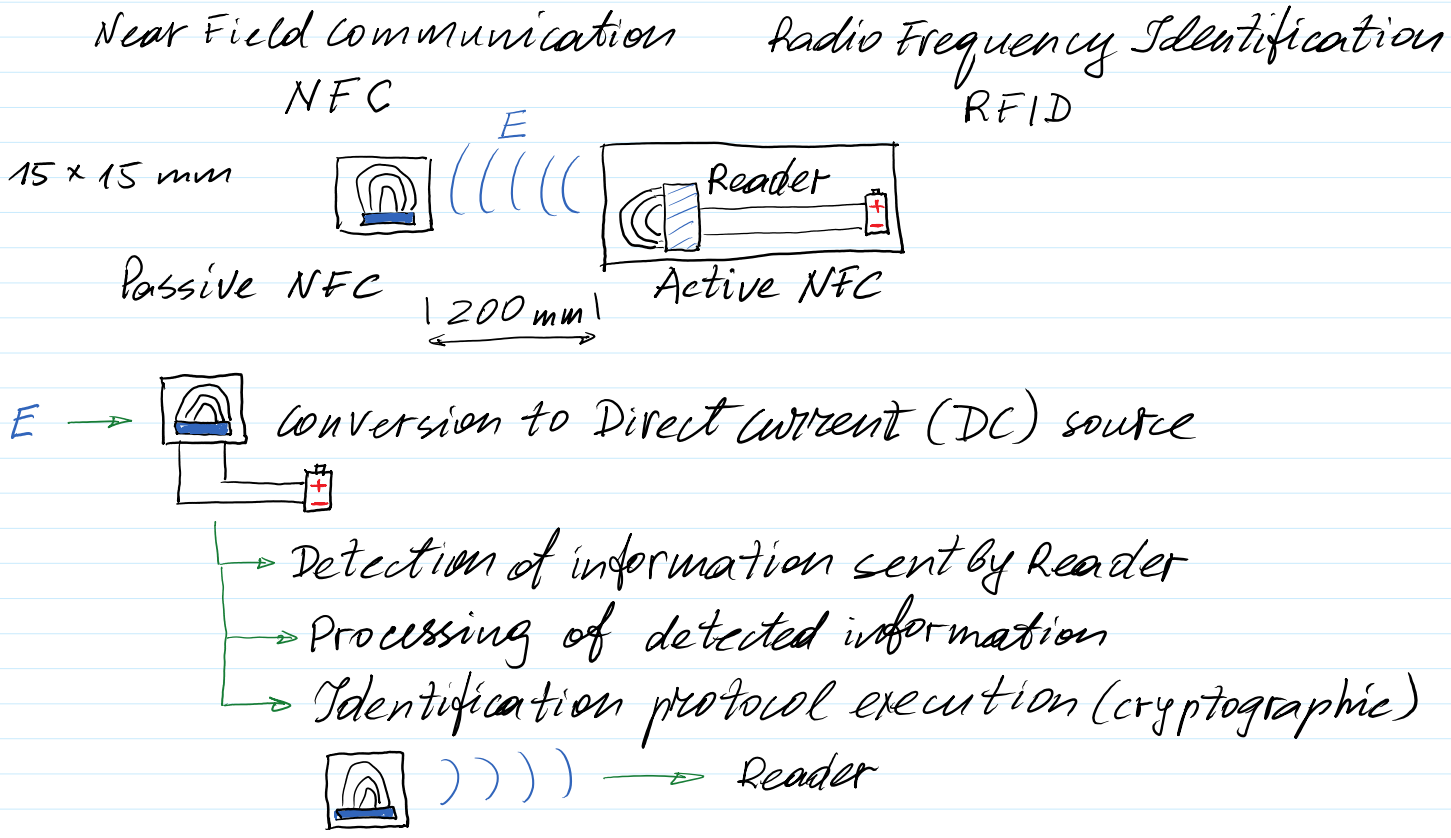
A team of researchers from the U.S. and Europe has developed a Hardware Trojan, which is an undetectable to many techniques, raising the question on need of proper hardware qualification.
They released a paper on stealthy Dopant-Level Hardware Trojans, showing how integrated circuits used in computers, military equipment and other critical systems can be maliciously compromised during the manufacturing process.
"*In this paper we propose an extremely stealthy approach for implementing hardware Trojans below the gate level, and we evaluate their impact on the security of the target device. Instead of adding additional circuitry to the target design, we insert our hardware Trojans by changing the dopant polarity of existing transistors.*" states the paper abstract.

Personal Identification Chip that is about the size of a grain of rice and implanted under the skin.

Would YOU let your boss implant you with a microchip? Belgian firm offers to turn staff into cyborgs to replace ID cards.

Read more: http://www.dailymail.co.uk/sciencetech/article-4203148/Company-offers-RFID-microchip-implants-replace-ID-cards.html#ixzz57Z0aeYgj
Follow us: @MailOnline on Twitter | DailyMail on Facebook

From <http://www.dailymail.co.uk/sciencetech/article-4203148/Company-offers-RFID-microchip-implants-replace-ID-cards.html>



INTRINSIC ID

***Intrinsic* ID SRAM PUF Technology & Solutions: Physically Unclonable Function**

Intrinsic ID delivers strong, device-unique data security and authentication solutions for the connected world. These authentication solutions are based on Intrinsic ID's patented SRAM Physical Unclonable Function or SRAM PUF technology.

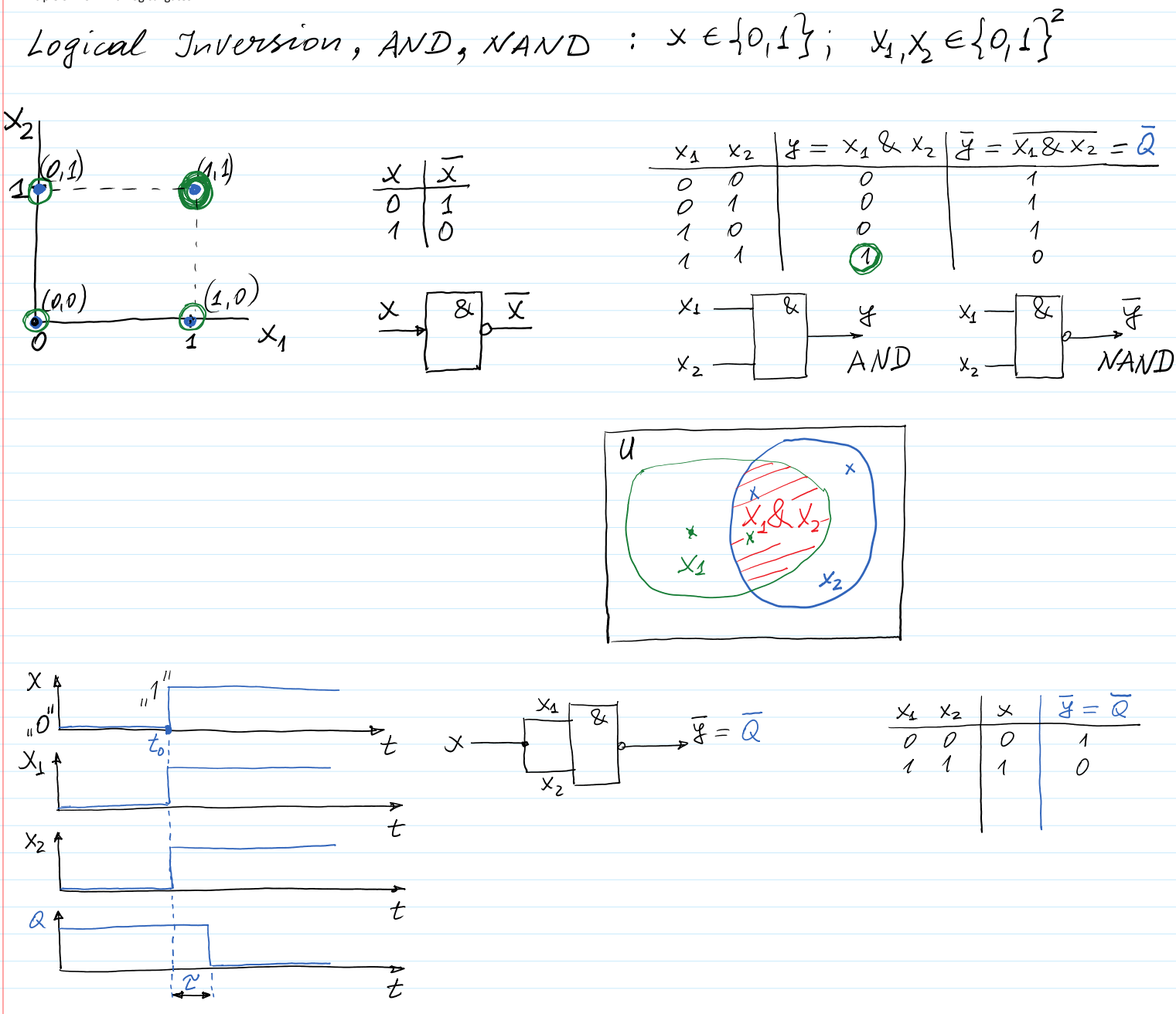
Using this technology, security keys and unique identifiers can be extracted from the innate characteristics of each semiconductor. Similar to biometrics measures, these identifiers cannot be cloned, guessed, stolen or shared. Keys are generated only when required and don't remain stored on the system, hence providing the highest level of protection.

Our SRAM PUF-based security solutions are very suitable for applications such as secure key generation and storage, device authentication, flexible key provisioning and

chip asset management. They can be used to secure payments, to protect highly sensitive data, for anti-counterfeiting and anti-cloning, to prevent identity theft, piracy of media content and software apps, software reverse engineering, and more.
Intrinsic ID's security solutions are available as hard and soft Intellectual Property (IP) and are used by companies who want a proven, easy and cost-efficient way to provide a solid trust base within their devices and applications.

From <https://www.intrinsic-id.com/sram-puf-technology-solutions/>

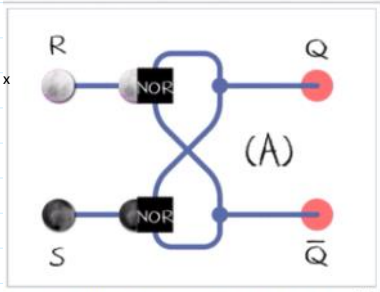
Principle of work with logical gates



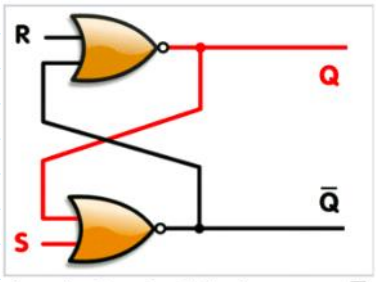
Flip-flop (electronics)

From <https://en.wikipedia.org/wiki/Flip-flop_(electronics)>

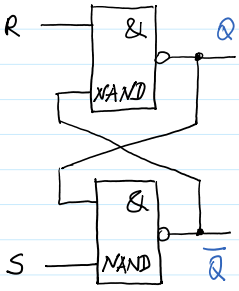
From <https://en.wikipedia.org/wiki/Flip-flop_(electronics)>



An animated SR latch. Black and white mean logical '1' and '0', respectively.
(A) S = 1, R = 0: set
(B) S = 0, R = 0: hold
(C) S = 0, R = 1: reset
(D) S = 1, R = 1: not allowed
The restricted combination (D) leads to an unstable state.

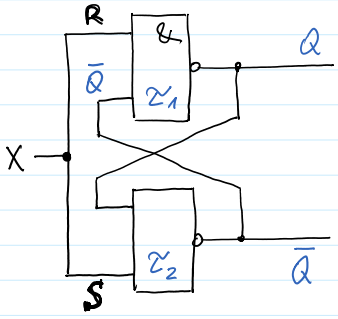


An animation of a SR latch, constructed from a pair of cross-coupled NOR gates. Red and black mean logical '1' and '0', respectively.

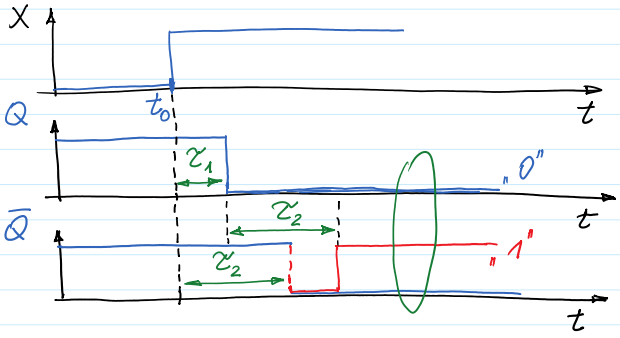


R	S	Q	$\bar{Q}$
1	0	0	1
1	1	0	1
0	1	1	0
1	1	1	0
0	0	1	1
1	1	?	?

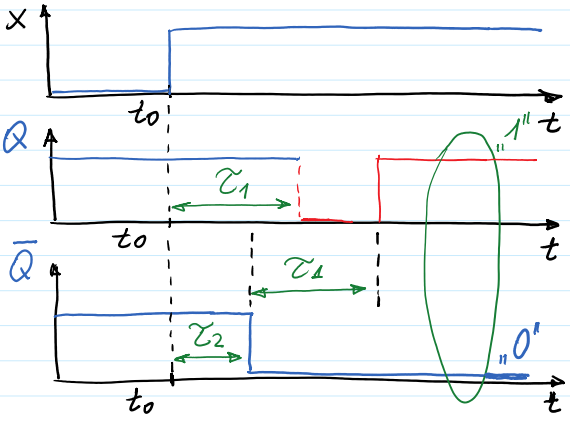
RS-Flip-flop

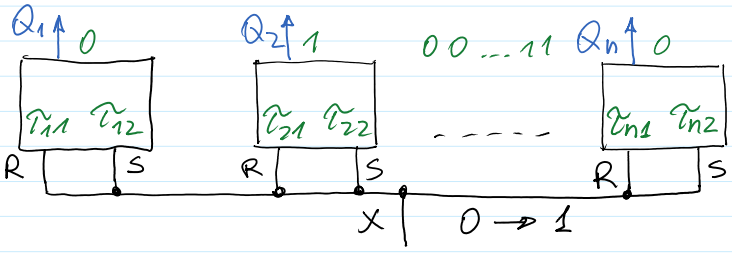


If $\tau_1 < \tau_2$



If $\tau_1 > \tau_2$



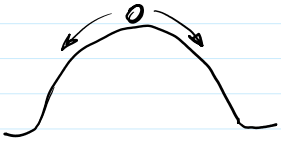


(0100....110)

1 Eingang an n RS-Trig.

$\tau_{21} \approx \tau_{22}$ $\tau_{i1} \approx \tau_{i2}$ $\tau_{j1} \approx \tau_{j2}$

$\tau_{21} < \tau_{22} \rightarrow \tau_{21} > \tau_{22}$ $\quad$ $\tau_{i1} > \tau_{i2} \rightarrow \tau_{i1} < \tau_{i2}$



About 15% of RS flip-flops are unstable due to $\tau_{i1} \approx \tau_{i2}$ and are sensible to random changing of environment conditions.
Every transition of X from 0 → 1 will have 15% errors
Solution is to apply Error Correcting Codes (ECC):
to correct ~ 25% errors.
It is recomended to use 320 – 400 bits PUF.
Using 400 bits PUF we have 25% of correct bits values,
i.e. 300 bits.
How many PUFs can be produced and distributed?
different
$N_{PUFs} = 2^{300} \approx 10^{100}$; The number of Planet population will be soon 8 Mlrd $= 8 \cdot 10^{9}$.