# 9 zkSNARKs from Extended Algebraic Programs

In this section we provide a new type of NIZK arguments which allows to prove knowledge of a secret key, given a public key, with only one 1 gate instead of >864 as in current state of the art implementations (disregarding the constraints they require in addition to assure that the input is valid).It is a extension of Groths[20] QAP based zkSNARK construction, where we roughly speaking add the word for 'point multiplication on an elliptic curve', to the words '+' and '×'. Therefor the language of the arithmetic circuit, R1CS and QAP had to be extended consequently to what we name R1CS* and extended algebraic program (EAP) instead of the commonly used QAPs.

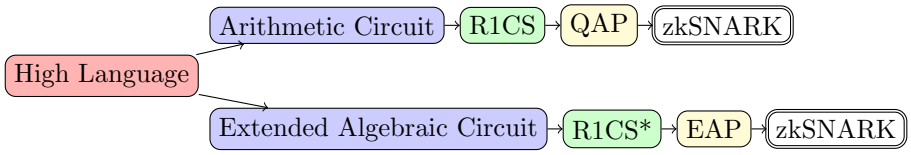



Figure 8: Shows the necessary steps to make a program written in a high level language (could be a subset of common C), provable with a zkSNARK. The upper path is the standard procedure. The path below our contribution.

The verifier work increases from 3 to 4 pairing function calls, 1 exponentiation in $\mathbb{G}_{\mathbb{T}}$, 1 in $\mathbb{G}_{\not\perp}$ and 2 in $\mathbb{G}_{\not=}$. Before this extension, proving knowledge of the discrete logarithm (e.g. I prove I know x s.t. $pk = g^x$ without revealing $x$) had to be done by expressing the point multiplication as arithmetic circuit. This required at least 864 gates for each point multiplication within a circuit, $N{\cdot}864$ for $N$ point multiplications, what increases CRS by at least $N \cdot 3 \cdot 864$ elements. Here its $N$ gates and therefore $3N$ CRS elements. The proofer time improves drastically as he does not need to compute the coefficients of the polynomials that were needed for proving knowledge of a valid arithmetic circuit assignment. Consequently the prover also does not need to perform the corresponding blind evaluations, which requires a point multiplications for each coefficient end therefor the most largest source of overhead. The polynomial division effort decrease as the degree is reduced by $N{\cdot}864$. Before this scheme, it was almost infeasible to pack multiple point multiplications into one SNARK. Performing a shielded transaction in zCash could be done in a few milliseconds instead of seconds now. To make the proof elements uniformly random and therefore zero knowledge, only one additional gate is required.

Our NIZK arguments for EAPs considers a circuit consisting of addition, multiplication and elliptic curve point multiplication gates. The computations are performed over a finite field $\mathbb{F}$. Figure 9 shows a how a EQAP enforced R1CS* constraint for a specific statement looks like.

The description of a EQAP is

$$EAP := (\mathbb{F}, e(\cdot,\cdot), M(\cdot), l, \{L_i(X), R_i(X), E_i(X), O_i(X)\}_{i=0}^{m}, D(X)),$$

where

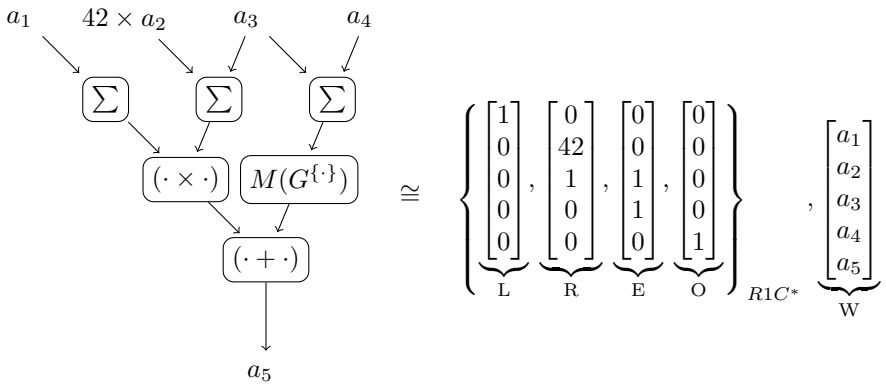



Figure 9: Shows how the example statement $a_1 \times (42 \times a_2 + a_3) + M\left(G^{a_3+a_4}\right) = a_5$ can be expressed as one R1C*, where W needs to be choosen s.t. it satisfies: $M\left(G^{\langle E,W\rangle}\right) + \langle L,W\rangle \cdot \langle R,W\rangle = \langle O,W\rangle$.

- $e$ is a bilinear mapping $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ as described in section 7. $G$ as $\mathbb{G}_1$ and $H$ as $\mathbb{G}_2$ generators.
- $n$ is the index-number of EAP gates
- $m$ is the number of wires of the circuit
- $L_i(X), R_i(X), E_i(X), O_i(X)$ are polynomials of degree $n-1$ that express the scale factor the $i$-th wire at gate $X$. For example: $L_3(42) = 1$ states that the 3rd wire is the 'left' multiplication input of gate with index 42. $R_7(42) = 2$ implies, that the 7th wire is 'right' multiplication input of gate with index 42 and that this wires assigned signal will be scaled by the linear factor of 2. $E_7(42) = 1$ implies, that the 7th wire is exponentiation input of gate with index 42. We write $\mathbf{L}$ for the vector $(L_0, L_1, .., L_m)$ (same for **R,E,O**).
- Each EQAP-gate has a uniquely assigned index $r_i \in \{1, ..., n\}$, which compose the domain polynomial $D(x) = \prod_{g=1}^{n}(x - r_g)$.
- A crucial role plays the projection function $M(\cdot) : \mathbb{G}_1 \mapsto \mathbb{F}$. Its best possible design currently exceed our understanding. Its most trivial realization would be taking $x$ coordinate of the curve point. This would reduce security by factor 2 and reconstructing the point within the circuit would be troublesome. It has turned out to be very beneficial if the field order $|\mathbb{F}|$ where the arithmetic operations are performed on is equivalent to the order of the field on which the curve is defined. This brings the advantage that curve operations can be performed within the circuit using far fewer gates. In case the pairing function is based on the asymmetric Tate pairing $\mathbb{G}_1 \subset E(F_{q^k})$.

A EAP is defined by the binary relation

$$R = \left\{ (\phi\ ,\mathrm{w}) \;\middle|\; \begin{array}{l} \phi = (a_1, ..., a_l) \in \mathbb{F}^l \\ w = (a_{l+1}, ..., a_m) \in \mathbb{F}^{m-1} \\ M\left(G^{\langle \mathbf{a}, \mathbf{E}(X)\rangle_{[m]}}\right) + \langle \mathbf{a}, \mathbf{L}(X)\rangle_{[m]} \cdot \langle \mathbf{a}, \mathbf{R}(X)\rangle_{[m]} \equiv \langle \mathbf{a}, \mathbf{O}(X)\rangle_{[m]} \bmod D(X) \end{array} \right. \tag{11}$$

where $\langle \mathbf{A}, \mathbf{B}\rangle_{[N_i,N_j]} := \sum_{i=N_i}^{N_j} A_i B_i$, and $\langle \mathbf{A}, \mathbf{B}\rangle_{[N_j]} := \langle \mathbf{A}, \mathbf{B}\rangle_{[0,N_j]}$

This gives us the NIZK argument:

$(\sigma, \tau) \leftarrow$Setup$(R, 1^\lambda)$: Pick $\alpha, \beta, \gamma, \delta \xleftarrow{\$} \mathbb{Z}_p^*, x \xleftarrow{\$} \mathbb{Z}_p^* \backslash \{2, ..., n\}$. Define $\tau = (\alpha, \beta, \gamma, \delta, x)$. The common reference string is:

$$\sigma = \left( \begin{array}{c} \left\{ G^{\frac{\alpha R_i(x) + \beta(L_i(x) + E_i(x)) + O_i(x)}{\gamma}} \right\}_{i=0}^{l}, \left\{ G^{\frac{\alpha R_i(x) + \beta(L_i(x) + E_i(x)) + O_i(x)}{\delta}} \right\}_{i=l+1}^{m} \\ G^\alpha, G^\beta, G^\delta, H^\beta, H^\gamma, H^\delta, \left\{ G^{\frac{x^i D(x)}{\delta}} \right\}_{i=0}^{n-1}, \left\{ G^{x^i} \right\}_{i=0}^{n-1}, \left\{ H^{x^i} \right\}_{i=0}^{n-1} \end{array} \right)$$

$\pi = (A, B, C, D) \leftarrow$ Prove $(EAP, \sigma, \phi, w)$: Compute

$$\begin{aligned} A &= G^{\alpha + \langle \mathbf{a}, \mathbf{L}(x)\rangle_{[m]} + \langle \mathbf{a}, \mathbf{E}(x)\rangle_{[m]}} \\ B &= H^{\langle \mathbf{a}, \mathbf{R}(x)\rangle_{[m]}} \\ C &= G^{\frac{\langle \mathbf{a}, \beta\mathbf{L}(x) + \alpha\mathbf{R}(x) + \beta\mathbf{E}(x) + \mathbf{O}(x)\rangle_{[l+1,m]} + Q(x)D(x)}{\delta}} \\ F &= G^{\langle \mathbf{a}, \mathbf{E}(x)\rangle_{[m]}} \end{aligned}$$

$0/1 \leftarrow$Vfy$(EAP, \sigma, \phi, \pi)$ accept iff:

$$e\left(A, B \cdot H^\beta\right) \cdot e\left(G, H\right)^{M(F)} = \left( e\left(G^\alpha, H^\beta\right) \cdot e\left( G^{\frac{\langle \mathbf{a}, \alpha\mathbf{R}(x) + \beta(\mathbf{L}(x) + \mathbf{E}(x)) + \mathbf{O}(x)\rangle_{[l]}}{\gamma}}, H^\gamma \right) \cdot e\left(C, H^\delta\right) \cdot e\left(F, B\right) \right)$$

Regarding efficiency we observe that the one-time setup $\sigma$ runs in time linear to the circuits size $\mathcal{O}(|C|)$. The prover must perform $\mathcal{O}(|C|)$ cryptographic work and $\mathcal{O}(|C| log^2(|C|))$ to compute $Q(x)$. The polynomial representation $L_i(X), R_i(X)$ etc. is not needed for proving and the prover rather works with the vectors $l_i = (L_i(1), L_i(2), .., L_i(n)), r_i = (R_i(1), R_i(2), .., R_i(n))$ where most elements are 0. This sparsity of the evaluation vectors is then exploited. The $\mathcal{O}(|C| log^2(|C|))$ runtime is achieved by using FFT techniques and binary tree based polynomial interpolation algorithms as it is done in the Pinocchio-Protocol[21] instead of naive Lagrange interpolation and polynomial division which would take $\mathcal{O}(n^2)$.

Note that $e\left(G, H\right)$ and $e\left(G^\alpha, H^\beta\right)$ can be precomputed. Furthermore verification requires $l$ exponentiations e.g. proofing time is quasi linear in the statement size. Constructing the pairing-friendly elliptic curves such that $\mathbb{G}_1$ representations of group elements are smaller, $C$ is assigned to the first group for efficiency. For the same reason the verifier includes the statement $\phi$ into the verification process over the group $\mathbb{G}_1$.

We proof perfect completeness by direct verification:

$$
\begin{aligned}
e\left(A, B \cdot H^{\beta}\right) \cdot e\left(G, H\right)^{M(D)} &= \\
e\left(G, H\right)^{\left(\alpha+\langle \mathbf{a},\mathbf{L}\rangle_{[m]}+\langle \mathbf{a},\mathbf{E}\rangle_{[m]}\right)\left(\langle \mathbf{a},\mathbf{R}\rangle_{[m]}+\beta\right)+M(F)} &= \\
e\left(G, H\right)^{\alpha\langle \mathbf{a},\mathbf{R}\rangle_{[m]}+\langle \mathbf{a},\mathbf{L}\rangle_{[m]}\langle \mathbf{a},\mathbf{R}\rangle_{[m]}+M(F)+\langle \mathbf{a},\mathbf{E}\rangle_{[m]}\langle \mathbf{a},\mathbf{R}\rangle_{[m]}+\alpha\beta+\beta\langle \mathbf{a},\mathbf{E}+\mathbf{L}\rangle_{[m]}} &= \\
e\left(G, H\right)^{\alpha\langle \mathbf{a},\mathbf{R}\rangle_{[m]}+\langle \mathbf{a},\mathbf{O}\rangle_{[m]}+QD+\langle \mathbf{a},\mathbf{E}\rangle_{[m]}\langle \mathbf{a},\mathbf{R}\rangle_{[m]}+\alpha\beta+\beta\langle \mathbf{a},\mathbf{E}+\mathbf{L}\rangle_{[m]}} &= \\
e\left(G^{\alpha}, H^{\beta}\right) \cdot e\left(G, H\right)^{\langle \mathbf{a},\alpha\mathbf{R}+\beta\mathbf{E}+\beta\mathbf{L}+\mathbf{O}\rangle_{[m]}+QD+\langle \mathbf{a},\mathbf{E}\rangle_{[m]}\langle \mathbf{a},\mathbf{R}\rangle_{[m]}} &= \\
e\left(G^{\alpha}, H^{\beta}\right) \cdot e\left(G, H\right)^{\langle \mathbf{a},\alpha\mathbf{R}+\beta\mathbf{E}+\beta\mathbf{L}+\mathbf{O}\rangle_{[m]}+QD} \cdot e\left(F, B\right) &= \\
e\left(G^{\alpha}, H^{\beta}\right) \cdot e\left(G^{\frac{\langle \mathbf{a},\alpha\mathbf{R}+\beta\mathbf{E}+\beta\mathbf{L}+\mathbf{O}\rangle_{[l]}}{\gamma}}, H^{\gamma}\right) \cdot e\left(C, H^{\delta}\right) \cdot e\left(F, B\right) & \quad \square
\end{aligned}
$$

Since we do not know how to proof soundness, we try to argue why it might be sound. Lets call the exponents of the proof elements $a, b, c, f$ where $A = G^a, B = H^b, C = G^c, F = G^f$. We also define $In = \frac{\langle \mathbf{a},\alpha\mathbf{R}(x)+\beta\mathbf{E}(x)+\beta\mathbf{L}(x)+\mathbf{O}(x)\rangle_{[l]}}{\gamma}$ as the input that will end up in the exponent a verifier will produce given the statement $\phi = (a_1, .., a_l)$. The equation for which an adversary effectively has to find a satisfying assignment is

$$a(b+\beta) + M(F) = \alpha \cdot \beta + In \cdot \gamma + c \cdot \delta + f \cdot b. \tag{12}$$

We now make the following observation:

- $a, b, c, f$ are under 100% control of the adversary. He can forge them at will.
- $In$ is depended on $\phi = (a_1, .., a_l)$ and the identity $a_0 = 1$. An PPT adversary cannot forge $In$ at will. He can however set it to 0 if $a_1 = a_2 = .. = a_l = 0 \wedge L_0 = R_0 = E_0 = O_0 = 0 \implies In = 0$ . If the identity $a_0$ is used in the circuit e.g. at least one of the wire polynomials $L_0, R_0, E_0, O_0$ is unequal to 0 then $In \neq 0$. Notice that the unlikely possibility exists, where there is $k \in \{1, .., l\}$ s.t. $L_k = L_0, R_k = R_0, E_k = E_0, O_0 = O_k$. Then setting $a_k = |\mathbb{F}|$ pushes $In = 0$ again.
- $M(F)$ is defined as $M(G^f)$. DLP on $\mathbb{G}_1$ is assumed to be hard, therfore a PPT adversary cannot forge $M(F)$ at will.
- $\alpha, \beta, \gamma, \delta$ are unknown to the attacker, however $G^{\alpha}$ etc. are known and part of the CRS. Therefor he can set $a, c, f \in \{\alpha, \beta, \delta\}$ and $b \in \{\beta, \gamma, \delta\}$.
- If there was an efficiently computable isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$,

We now try to reduce equation 12 to the seemingly easiest satisfiable form by setting $A = G^{\alpha}$,$b = c = 0$. Then it remains for an attacker to find an assignment for:

$$M(F) = In \cdot \gamma. \tag{13}$$

In case $M$ is the trivial function that takes a curve point and returns its X-coordinate and $G^0 = (0, 1)$, an adversary can create a satisfying assignment by setting $\phi = (0, .., 0)$. Consequently a verifier needs to check the proof elements or the statement, and reject if among them is one of the trivial assignments. If the trivial cases are rejected, finding a valid assignment therefore always ends up facing the DLP, which is assumed to be hard.

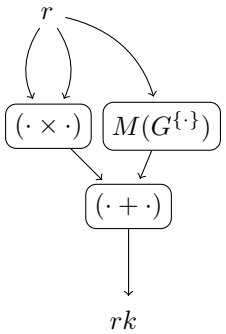
We want to point out, that Groth's protocol [20] upon we build this work, has the same trivial satisfying assignment. Using the same tricks and argumentation leads to:

$$0 = In \cdot \gamma, \tag{14}$$

which is satisfiable if $\phi = (0, .., 0), A = G^{\alpha}, B = H^{\beta}, C = G^0$. In both schemes the trivial assignment is not an immanent thread however since $a_0 = 1$ and its very unlikely, that the identity element is never used in any real world application circuit.

### 9.0.1 Perfekt zero-knowledge

Since the prover does not include any randomness at this point, the proof elements are not zero-knowledge. Proofing the same statement twice leads to equivalent proof elements and an adversary could learn more then the just the correctness of the statement. To make the proof elements uniformly random, we introduce a randomization gate. This gate has no connection to the rest of the circuit and is suggested to be the last gate regarding its index. The witness vector is then extended by two more elements $a_m$ and $a_{m-1}$ to satisfy this gate. Its structure and constraint is shown in figure 10. The prover picks $r \xleftarrow{\$} \mathbb{Z}_p^*$ and set $a_m = r * r + M(G^r)$ and $a_{m-1} = r$. The proof elements are now uniformly random.



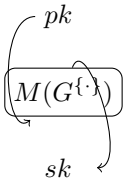

$$\rightsquigarrow \left\{ \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\text{L}}, \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\text{R}}, \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\text{E}}, \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{\text{O}} \right\}_{R1C^*}, \underbrace{\begin{bmatrix} a_0 = 1 \\ \vdots \\ a_{m-2} \\ a_{m-1} = r \\ a_m \end{bmatrix}}_{\mathbf{a}}$$

Figure 10: Shows the randomization gate and constraint. It enforces $a_m \equiv a_{m-1} \cdot a_{m-1} + M(G^{a_{m-1}}) \bmod p$. Since the $r^2$ might reveal some information (gut feeling) the randomization circuit could use three different random inputs for the price of increasing $|\mathbf{a}|$ by 4 instead of 2 elements.

## 9.1 Example

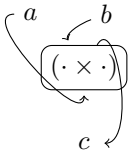
### 9.1.1 Inverse Gates

In case a public key is passed into the circuit as an argument and we want to proof knowledge of the secret key, we immediately observe that one gate to perform this inversion is sufficient since gates are only a guarantee that the input-output relation is satisfied and therefor they can be used in either direction.




$$\rightsquigarrow \quad \left\{ \underbrace{\begin{bmatrix}0\\0\end{bmatrix}}_{\mathrm{L}}, \underbrace{\begin{bmatrix}0\\0\end{bmatrix}}_{\mathrm{R}}, \underbrace{\begin{bmatrix}0\\1\end{bmatrix}}_{\mathrm{E}}, \underbrace{\begin{bmatrix}1\\0\end{bmatrix}}_{\mathrm{O}} \right\}_{R1C^*}, \underbrace{\begin{bmatrix}pk\\sk\end{bmatrix}}_{\mathbf{a}}.$$

Division can be understood likewise. Consider the example where we need to compute $c = a/b$ given $a, b$. The corresponding circuit and R1C* is




$$\rightsquigarrow \quad \left\{ \underbrace{\begin{bmatrix}0\\1\\0\end{bmatrix}}_{\mathrm{L}}, \underbrace{\begin{bmatrix}0\\0\\1\end{bmatrix}}_{\mathrm{R}}, \underbrace{\begin{bmatrix}0\\0\\0\end{bmatrix}}_{\mathrm{E}}, \underbrace{\begin{bmatrix}1\\0\\0\end{bmatrix}}_{\mathrm{O}} \right\}_{R1C^*}, \underbrace{\begin{bmatrix}a\\b\\c\end{bmatrix}}_{\mathbf{a}}.$$

### 9.1.2 Combined Gate

Consider a example where we want to apply multiplication and curve group multiplication at once in one gate. It can be done but its is **very important** to notice possible side effects! Lets say we have $a, b, c$ and want to compute $d = a * b$ and $f = M(G^c)$:

$$\left\{ \underbrace{\begin{bmatrix}1\\0\\0\\0\\0\end{bmatrix}}_{\mathrm{L}}, \underbrace{\begin{bmatrix}0\\1\\0\\0\\0\end{bmatrix}}_{\mathrm{R}}, \underbrace{\begin{bmatrix}0\\0\\1\\0\\0\end{bmatrix}}_{\mathrm{E}}, \underbrace{\begin{bmatrix}0\\0\\0\\1\\1\end{bmatrix}}_{\mathrm{O}} \right\}_{R1C^*}, \underbrace{\begin{bmatrix}a\\b\\c\\d\\f\end{bmatrix}}_{\mathbf{a}},$$

would be **insufficient** since all we could derive from this assignment is that $d + f = a * b + M(G^c)$. If we continue to use $d$ and $f$ we loose their determinism, however they remain entangled. If another constraint assigns a value to one of them, the other one will be fixed too. This phenomena we call 'un-spooky action in a circuit'.

### 9.1.3 Single point multiplication circuit

Lets consider the case, where only one point multiplication defines the circuit e.g. the prover wants to convince the verifier, that he knows $sk$ s.t $pk = M(G^{sk})$ without revealing $sk$. $M$ maps a curve point onto its $x$ coordinate for example. We therefor know that there must exist a second distinct $sk$ that map to $pk$, however this is still sufficiently secure for this example. The circuit, its corresponding constraint and the assignment vector become

$$\begin{array}{c} sk \\ \downarrow \\ \boxed{M(G^{\{\cdot\}})} \\ \downarrow \\ pk \end{array} \quad \rightsquigarrow \quad \left\{ \underbrace{\begin{bmatrix}0\\0\end{bmatrix}}_{\text{L}}, \underbrace{\begin{bmatrix}0\\0\end{bmatrix}}_{\text{R}}, \underbrace{\begin{bmatrix}1\\0\end{bmatrix}}_{\text{E}}, \underbrace{\begin{bmatrix}0\\1\end{bmatrix}}_{\text{O}} \right\}_{R1C^*}, \underbrace{\begin{bmatrix}sk\\pk\end{bmatrix}}_{\mathbf{a}},$$

therefor the polynomials $L_1(X) = L_2(X) = R_1(X) = R_2(X) = E_1(X) = O_2(X) = 0$, $E_2(X) = O_1(X) = 1$ are simple constant lines. The domain polynomial times the extracted polynomial therefor also $Q(X)D(X)$=0. The statement $\phi = (pk)$, the witness $w = (sk)$.

The proof elements become:

$$\begin{aligned} A &= G^{\alpha+sk} \\ B &= H^0 \\ C &= G^{\frac{\beta sk}{\delta}} \\ F &= G^{sk} \end{aligned}$$

The verifier checks if

$$\begin{aligned} e\left(A, B \cdot H^{\beta}\right) \cdot e\left(G, H\right)^{M(F)} = \\ e\left(G^{\alpha}, H^{\beta}\right) \cdot e\left(G^{\frac{pk}{\gamma}}, H^{\gamma}\right) \cdot e\left(C, H^{\delta}\right) \cdot e\left(F, B\right) \end{aligned}$$

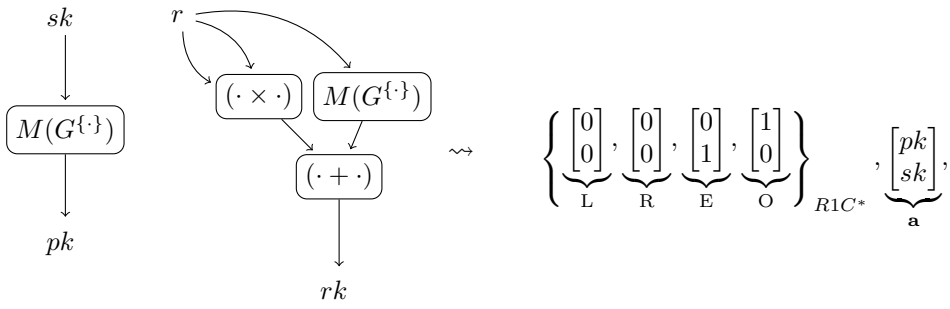
what is indeed true since

$$\begin{aligned} e\left(A, B \cdot H^{\beta}\right) \cdot e\left(G, H\right)^{M(F)} = \\ e\left(G, H\right)^{(\alpha+sk)\beta} \cdot e\left(G, H\right)^{pk} = \\ e\left(G, H\right)^{\alpha\beta} \cdot e\left(G, H\right)^{\frac{pk}{\gamma}\cdot\gamma} \cdot e\left(G, H\right)^{\frac{\beta sk}{\delta}\cdot\delta} \cdot e\left(G, H\right)^{sk\cdot 0} = \\ e\left(G^{\alpha}, H^{\beta}\right) \cdot e\left(G^{\frac{pk}{\gamma}}, H^{\gamma}\right) \cdot e\left(C, H^{\delta}\right) \cdot e\left(F, B\right) \quad \square \end{aligned}$$

From this we derive a simple cryptographic protocol: Proving knowledge of $sk$, s.t. $pk = G^{sk}$: Verifier: pick random $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$ send it to the prover. Prover: set the proof $\pi = (A) = (G^{sk+\alpha})$ Verifier take $\pi = (A), \alpha, pk$ and accept iff:

$$e(pk, H) \cdot e(G, H)^{\alpha} = e(A, H). \tag{15}$$

### 9.1.4 Point multiplication and randomization circuit




## 9.2 Trapdoors are Toxic Waste in PDLs

Trapdoors for pairing based NIZK protocols in general contain the point $x$ on which a proofer should blindly evaluate his witnessing QAP polynomial. If $x$ is known however, an adversary can forge a proof for arbitrary statements without knowledge of the witness $w$. For the protocol in section 9 with trapdoor $\tau = (\alpha, \beta, \gamma, \delta, x)$ we observe that proofs for arbitrary statements can easily be forge since:

$\pi \leftarrow \text{Sim}(R, \tau, \phi)$ : Choose $r, s \xleftarrow{\$} \mathbb{Z}_p$. Compute $\pi = (A, B, C)$ as

$$\begin{aligned} A &= g^r \\ B &= h^s \\ C &= g^{\frac{rs - \alpha\beta - \sum_{i=0}^{l} a_i(\beta L_i(x) + \alpha R_i(x) + O_i(x))}{\delta}}. \end{aligned}$$

Plugged into equation **??**, one can quickly verify its validity. This means that in case $\tau$ is not unrecoverable destroyed and forgotten, proofs without a witness $w$ at all can be created very efficiently. Due to the zero-knowledge property one cannot distinguish a fake proof from a true one. In a PDL environment used for coin transfer, using zkSNARKs to provide user privacy, coins could be minted out of nothing and silent inflation could take place.

# 10 Critics, Considerations and Speculations

In this final chapter I frequently state personal opinions which are no necessarily shared by my professors or institute. I will try to give arguments for my thoughts whenever necessary, however knowing, that my constraint knowledge and perception will led to wrong conclusions at some places. I hopefully find my arguments falsified, corrected or extended as the years go by.

## 10.1 Smart Contracts to Undermine Justice

Smart Contract (**SC**) are neither smart, nor contracts. Vitalik Buterin co founder of Ethereum once tried to point out the misconceptions resulting from