possibility to bloat the blockchain and produce an additional load on the nodes. To discourage malevolent participants from creating large blocks we introduce a penalty function:

$$NewReward = BaseReward \cdot \left(\frac{BlkSize}{M_N} - 1 \right)^2$$

This rule is applied only when *BlkSize* is greater than minimal free block size which should be close to $\texttt{max}(10\text{kb}, M_N \cdot 110\%)$. Miners are permitted to create blocks of "usual size" and even exceed it with profit when the overall fees surpass the penalty. But fees are unlikely to grow quadratically unlike the penalty value so there will be an equilibrium.

6.3 Transaction scripts

CryptoNote has a very minimalistic scripting subsystem. A sender specifies an expression $\Phi = f(x_1, x_2, \ldots, x_n)$, where n is the number of destination public keys $\{P_i\}_{i=1}^n$. Only five binary operators are supported: `min`, `max`, `sum`, `mul` and `cmp`. When the receiver spends this payment, he produces $0 \leq k \leq n$ signatures and passes them to transaction input. The verification process simply evaluates Φ with $x_i = 1$ to check for a valid signature for the public key P_i, and $x_i = 0$. A verifier accepts the proof iff $\Phi > 0$.

Despite its simplicity this approach covers every possible case:

- **Multi-/Threshold signature.** For the Bitcoin-style "M-out-of-N" multi-signature (i.e. the receiver should provide at least $0 \leq M \leq N$ valid signatures) $\Phi = x_1 + x_2 + \ldots + x_N \geq M$ (for clarity we are using common algebraic notation). The weighted threshold signature (some keys can be more important than other) could be expressed as $\Phi = w_1 \cdot x_1 + w_2 \cdot x_2 + \ldots + w_N \cdot x_N \geq w_M$. And scenario where the master-key corresponds to $\Phi = \texttt{max}(M \cdot x, x_1 + x_2 + \ldots + x_N) \geq M$. It is easy to show that any sophisticated case can be expressed with these operators, i.e. they form basis.

- **Password protection.** Possession of a secret password s is equivalent to the knowledge of a private key, deterministically derived from the password: $k = \texttt{KDF}(s)$. Hence, a receiver can prove that he knows the password by providing another signature under the key k. The sender simply adds the corresponding public key to his own output. Note that this method is much more secure than the "transaction puzzle" used in Bitcoin [13], where the password is explicitly passed in the inputs.

- **Degenerate cases.** $\Phi = 1$ means that anybody can spend the money; $\Phi = 0$ marks the output as not spendable forever.

In the case when the output script combined with public keys is too large for a sender, he can use special output type, which indicates that the recipient will put this data in his input while the sender provides only a hash of it. This approach is similar to Bitcoin's "pay-to-hash" feature, but instead of adding new script commands we handle this case at the data structure level.

7 Conclusion

We have investigated the major flaws in Bitcoin and proposed some possible solutions. These advantageous features and our ongoing development make new electronic cash system CryptoNote a serious rival to Bitcoin, outclassing all its forks.