Pricing via Processing or Combatting Junk Mail

Presentation by John Lim

Based on


Goal

• Create a system that would deter mass spam e-mail
• Penalize ill-intentioned parties within this network while not interfering with normal usage
• Senders must pay a cost to send messages
• General idea is:
  • “If I don’t know you and you want to send me a message, then you must prove that you spent, say, ten seconds of CPU time, just for me and just for this message.”

Pricing Function - Motivation

• Authors introduce the idea of a pricing function, $f$, as a method to impose a “cost” to senders in this network
  • Computing this cost serves no purpose than as a deterrent
• Senders will have to compute this moderately expensive, tractable function as a form of verification
• Proof of effort $z = f(m, sender\_info, recep\_info, t)$
• Some may have a shortcut, $c$
• $f$ can also be a relation
Pricing Function - Properties

• Moderately easy to compute but easy to verify
• Not amenable to amortization
  • Computing $f(m, \text{sender\_info}, A, t)$ does not make computing $f(m, \text{sender\_info}, B, t)$ easier
• Parametrized by a difference parameter $k$

Pricing Functions - Notations

• Family of pricing functions $F = \{f_s \mid s \in S\}$ indexed by $s \in S \subseteq \{0,1\}^*$
• Collection of families of pricing functions $\mathcal{F} = \{F_k \mid k \geq 1\}$
• $f_s \in F_k$ is our pricing function
Pricing Function

- If verification fails or if there is a substantial difference in $t$, the message is discarded.

\[
\begin{align*}
\text{Sender} & : f_s(h((m, t, d))) = z \\
\text{Recipient} & : f_s(h((m, t, d))) = z
\end{align*}
\]

Extracting Square Roots

- This implementation is based on the difficulty of extracting square roots a modulo $p$
  - $p$ is dependent on the difference parameter. 1024 bits was a reasonable length at the time of the paper.
- Domain of $f_p$ is $\mathbb{Z}_p$
- $f_p = \sqrt{x} \ mod \ p$
- To verify, given $x$ and $y$, check $y^2 \equiv x \ mod \ p$
- No shortcut
Fiat and Shamir based scheme

• Our pricing function $f_S$ is indexed by a $(k+2)$-tuple $(N, y_1, \ldots, y_k, h)$
  • $N = pq$ where $p$ and $q$ are large primes
  • $y_1 = x_1^2, \ldots, y_k = x_k^2$ are $k$ squares modulo $N$
  • $h$ is a hash function with domain $\mathbb{Z}_N^* \times \mathbb{Z}_N^*$ and range $\{0,1\}^k$
• We want to find a $z$ and $r^2$ such that $z^2 = r^2 x^2 \prod_{i=1}^{k} y_i^{b_i} \mod N$
  • $h(x, r^2) = b_1 \ldots b_k$
  • Shortcut: The square roots $x_1 \ldots x_k$
• $f_S = (z, r^2)$

Fiat and Shamir based scheme

• Given $x, z, r^2$, we verify by computing $h(x, r^2) = b_1 \ldots b_k$ and seeing if $z^2 = r^2 x^2 \prod_{i=1}^{k} y_i^{b_i} \mod N$ holds to be true
• Evaluation with shortcut information:
  • Choose $r$ at random
  • $h(x, r^2) = b_1 \ldots b_k$, and set $z = r x \prod_{i=1}^{k} x_i^{b_i}$. $f_s(x) = (z, r^2)$
• Evaluation without shortcut information is considerably harder
Fiat and Shamir based scheme

- Evaluation without a shortcut
- Expected # of iterations is $2^k$
- Verification will take $\frac{k}{2} + 1$ multiplications

$f_s(x) = (z, r^2)$ can be computed as follows.

Guess $b_1 \ldots b_k \in \{0, 1\}^k$.
Compute $B = \prod_{i=1}^{k} y_i^{b_i} \mod N$.
Repeat:
  - Choose random $z \in \mathbb{Z}_N^*$
  - Define $r^2$ to be $r^2 = (z^2/Bx^2) \mod N$
Until $h(x, r^2) = b_1 \ldots b_k$.

Ong-Schnorr-Shamir Based Scheme

- Our pricing function $f_S$ is indexed by a 2-tuple $(N, l)$
  - $N = pq$ where $p$ and $q$ are large primes
  - $l \in \mathbb{Z}_N^*$
- The domain of $f_S$ is $\mathbb{Z}_N^*$
- We want to find $x_1$ and $x_2$ such that $x_1^2 + lx_2^2 = x \mod N$
  - Shortcut $u$ such that $u^2 = -l^{-1} \mod N$
  - $f_s = (x_1, x_2)$
- Evaluation without shortcut using Pollard’s algorithm (not discussed in paper)
Ong-Schnorr-Shamir Based Scheme

- Evaluation with shortcut $u$
- Public key random element, $l$, in $\mathbb{Z}_N^*$
- Private key $u$ such that $u^2 = -l^{-1} \mod N$
- A signature for a message $m \in \{0, \ldots, N - 1\}$ is a solution to $x_1^2 + lx_2^2 = x \mod N$
  - Efficient signing algorithm given shortcut:
    - choose random $r_1, r_2 \in \mathbb{Z}_n^*$ such that $r_1 \cdot r_2 = m \mod N$
    - set $x_1 = \frac{1}{2} \cdot (r_1 + r_2) \mod N$ and $x_2 = \frac{1}{2} \cdot u \cdot (r_1 - r_2) \mod N$.
- Evaluation without private key will take $\log N$ iterations vs. constant number of multiplications for evaluation with private key.

Hash Functions

- DES, MD4, MD5, Subset Sum, Snefru are all valid candidates for $h$
- $f_s (h (m, t, sender info, recipient info))$
- Also used for the Fiat-Shamir scheme
Summary

• Authors introduce the concept of proof of work to deal with spam messages
  • Foundation for proof of work which bitcoin + ethereum both use
• Fiat-Shamir is the preferred pricing function of the 3 introduced in the paper