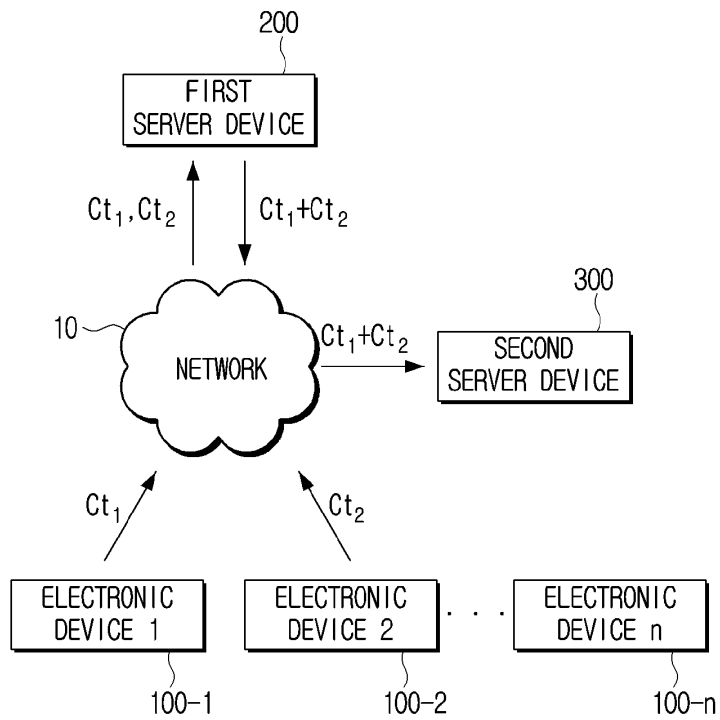
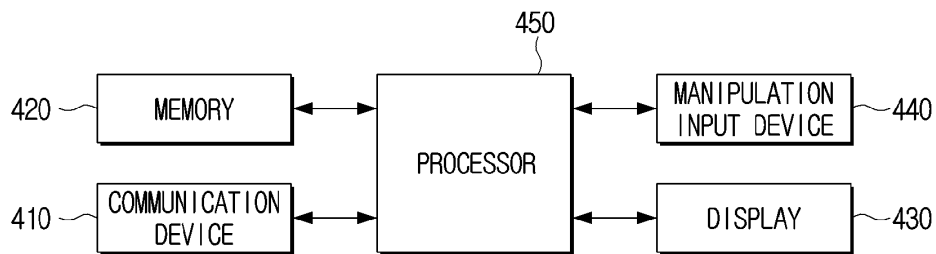


[Fig. 1]

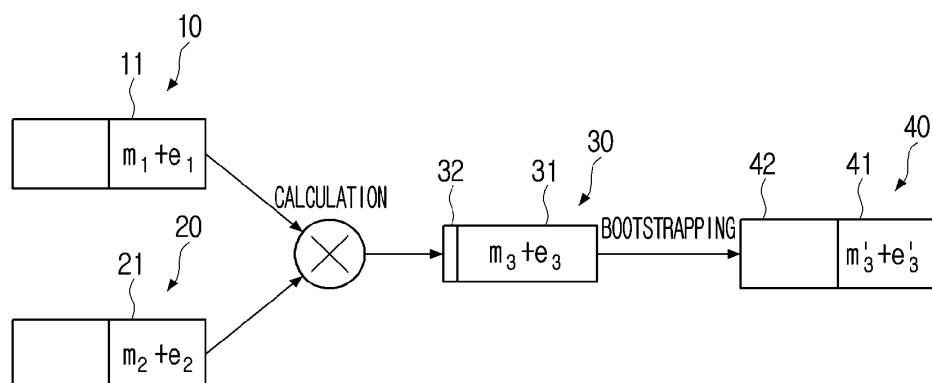


[Fig. 2]

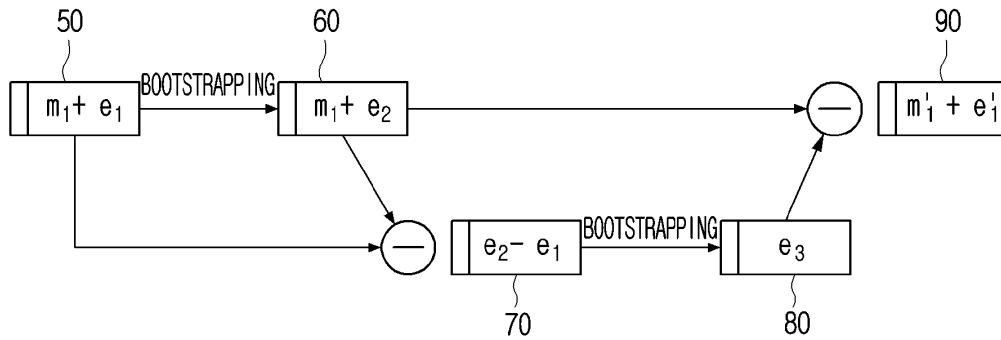
400



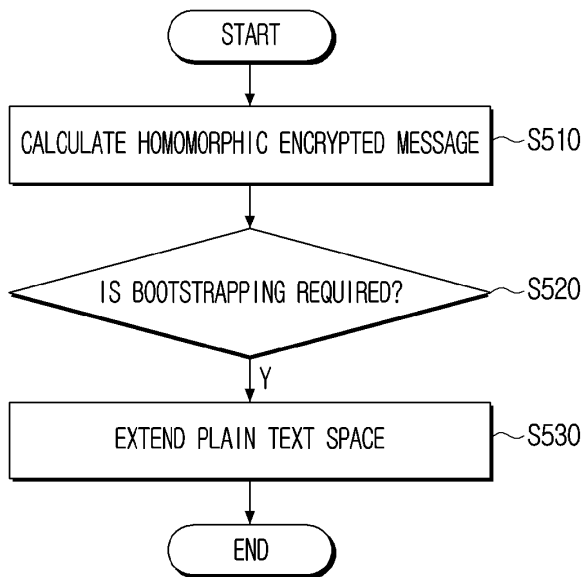
[Fig. 3]



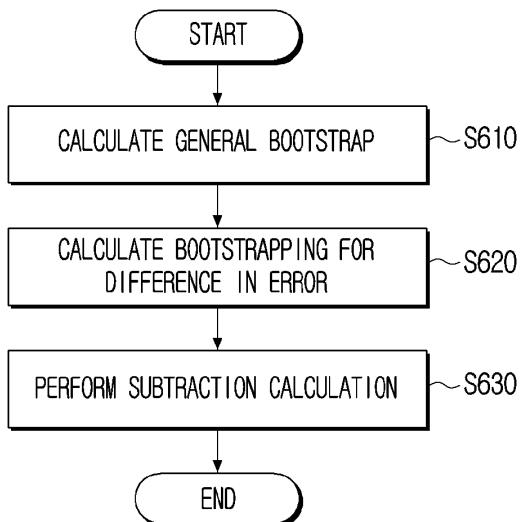
[Fig. 4]



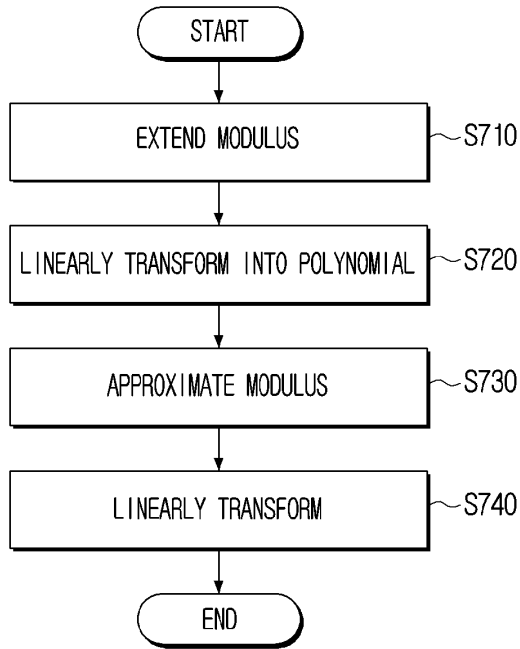
[Fig. 5]



[Fig. 6]



[Fig. 7]



[Fig. 8]

Algorithm 1 Bootstrap Extension

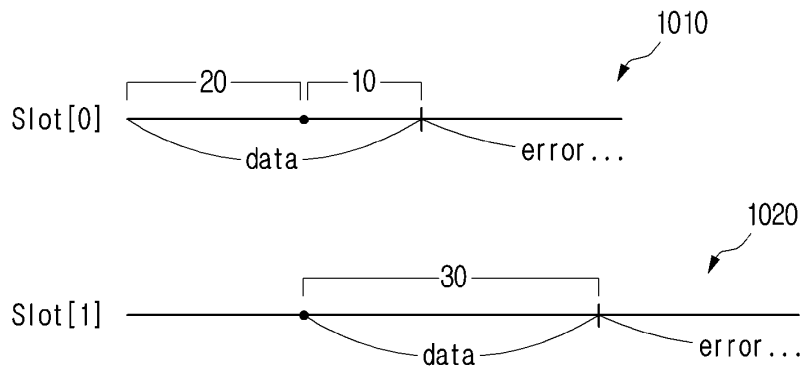
$\begin{aligned} & op \leftarrow x \\ & tmp \leftarrow 2^{-n} * op \\ & tmp \leftarrow \text{BTS}(tmp) \\ & tmp \leftarrow 2^n * tmp \\ & tmp2 \leftarrow tmp - op \\ & tmp2 \leftarrow \text{BTS}(tmp2) \\ & res \leftarrow tmp - tmp2 \\ & \text{return } res \end{aligned}$	$\begin{aligned} & \triangleright op = [x]_p \\ & \triangleright tmp = [2^{-n}x]_p \\ & \triangleright tmp = [2^{-n}x + e_1]_q \\ & \triangleright tmp = [x + 2^n e_1]_q \\ & \triangleright tmp2 = [2^n e_1]_p \\ & \triangleright tmp2 = [2^n e_1 + e_2]_q \\ & \triangleright res = [x - e_2]_q \end{aligned}$
--	---

[Fig. 9]

Algorithm 2 $\text{BTS}^{(k+1)}$ using $\text{BTS}^{(k)}$

$\begin{aligned} & op \leftarrow x \\ & tmp \leftarrow 2^{-n} * op \\ & tmp \leftarrow \text{BTS}^{(k)}(tmp) \\ & tmp \leftarrow 2^n * tmp \\ & tmp2 \leftarrow tmp - op \\ & tmp2 \leftarrow \text{BTS}(tmp2) \\ & res \leftarrow tmp - tmp2 \\ & \text{return } res \end{aligned}$	$\begin{aligned} & \triangleright op = [x]_{p_{k+1}} \\ & \triangleright tmp = [2^{-n}x]_{p_k} \\ & \triangleright tmp = [2^{-n}x + e_1]_q \\ & \triangleright tmp = [x + 2^n e_1]_q \\ & \triangleright tmp2 = [2^n e_1]_{p_k} \\ & \triangleright tmp2 = [2^n e_1 + e_2]_q \\ & \triangleright res = [x - e_2]_q \end{aligned}$
--	---

[Fig. 10]



[Fig. 11]

