(1) The database of random numbers is bigger (how much may depend on whether you have expiry dates, but maybe 4Gb vs 1Gb) and there is an overhead involved in number issue as well as ~~redep~~ redemption - you have to check the number's not in use. OTOH a database is harder to steal than a crypto key

(2) A block cipher on n decimal digits (for $n = 9$, 12, 16 for example, for different parts of this question) can be constructed using the Feistel model

(3) 1bn possible codes → 200 per customer, and 4 cards in issue per customer, so a guess has a 1 in 50 chance of hitting an active number. Thus there are about 100,000 guesses/month and maybe 50,000 'chancers'.
Remedies
1. Switch to longer codes - 12 digits or even 16 anticipating part 4 of the question
2. Intrusion detection system to spot the habitual guessers and impose some penalty mechanism
3. (I'm really interested in good new ideas!)
Pros and cons - longer codes are a systematic fix but have a larger capital cost. Maybe, if you're using a database system, you need a big upgrade - or a switch to an encrypted-counter system

(4) With 100 million customers the database gets much bigger, so encrypted-counter is the way forward. 12 digits might not be enough; you might go to 16 to solve the problem definitively. But whether 12 or 16 you may have to pay attention to security-usability issues, e.g. group the digits as 3 or 4 groups of 4, to prevent honest errors in code entry