It's not safe for Bob. Alice could have him sign anything.

Cut and choose solves this — or makes the probability of Bob being duped as small as he wishes.

1. Alice produces ten documents and blinds each one.
2. Bob nominates nine of them to be unblinded. He checks he's happy with the content, then signs the tenth.
3. Alice unblinds that document; now she has a document signed by Bob.

— in principle, vulnerable to something like the Birthday Book attack.

Applications of blind signatures

- secure e-cash (see later)
- electronic voting
- more esoteric versions: oblivious signature protocols exist (do they have applications?)
  - Alice has *n* messages; Bob can choose one of these for Alice to sign, and she has no way of knowing which one she signed
  - (*) Alice has one message and *n* keys. Bob can select a key for Alice to use, and she has no way of knowing which one she used.

{ FIRING SQUAD: Convict takes one bullet from n Squaddies. Officer randomly ~~selects~~ selects one ~~squad~~ gun to hold bullet. Convict (or anyone else) has no way of knowing whose gun fired the ~~killing~~ ~~shot~~ bullet. } (Analogous to (*) above)

Contract Signing

how to get Alice and Bob to commit to a contract simultaneously from a distance?

Goal: both end up with a copy of the contract signed by both parties; neither can end up being bound without the other also being bound

algorithmic certainty cannot generally be achieved; these protocols work on probabilities

- use a trusted third party (e.g. Verisign?)
- use a bitwize (characterwise) synchronization

?