Zero Knowledge Proofs

Alan R. Hahn

Clemson University

April 2021

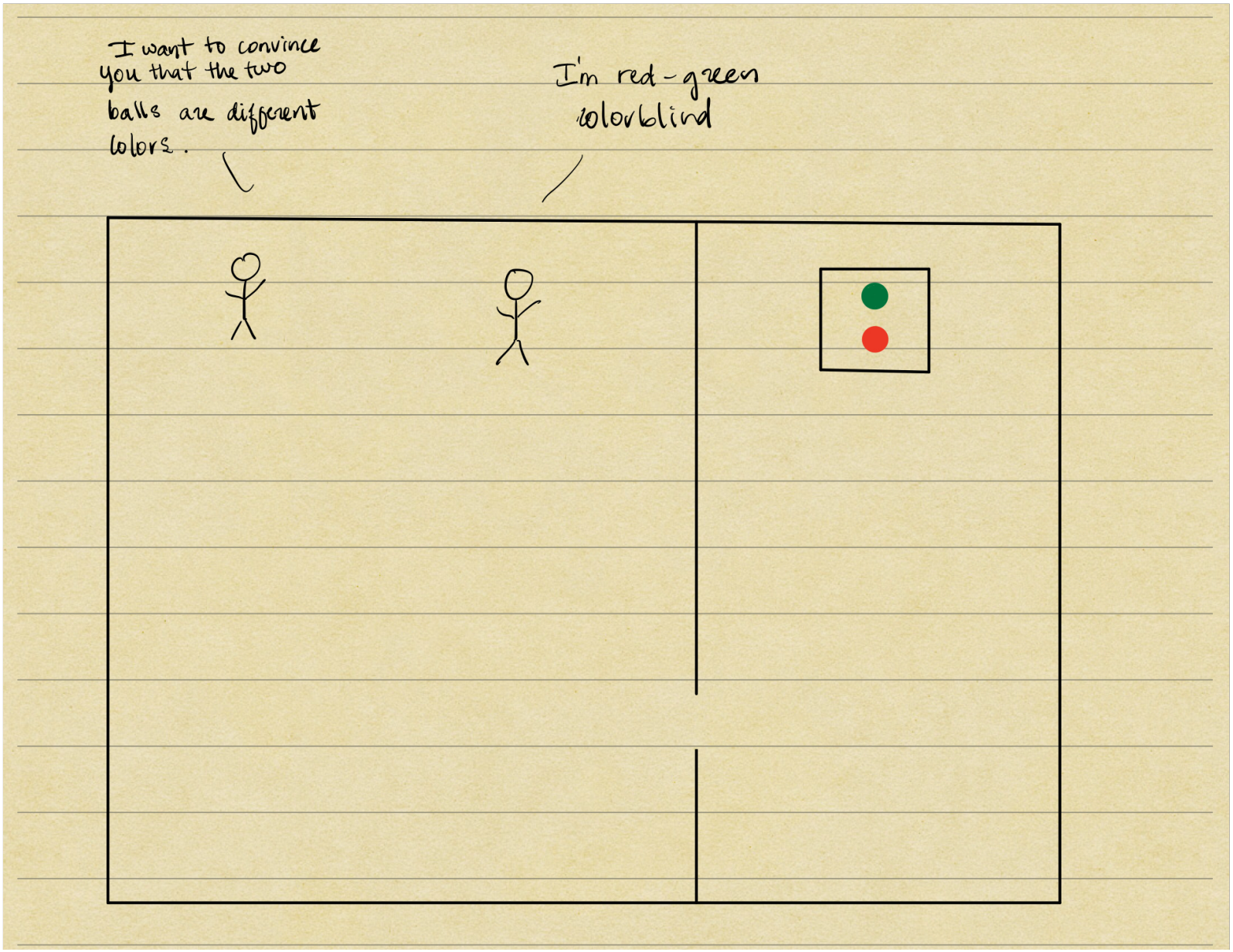
Idea (ZKP)

Want to convince to someone that you know something without giving away information about what you know

Idea (ZKP)

Want to convince to someone that you know something without giving away information about what you know

- Identification Schemes and Entity Authentication


I want to convince you that the two balls are different colors.
I'm red-green colorblind

Definitions

- **Prover**: *One proving that they have knowledge of something*
- **Verifier**: *One verifying that prover really does have knowledge*
- **Completeness**: *If statement is true, honest verifier will in fact be convinced it is true by honest prover*
- **Soundness**: *If statement is false, no cheating prover can convince honest verifier it is true, except with small probability (soundness error)*
- **Zero Knowledge**: *If statement is true, no verifier learns anything other than the fact that the statement is true*

Definitions

- **Prover**: *One proving that they have knowledge of something*
- **Verifier**: *One verifying that prover really does have knowledge*
- **Completeness**: *If statement is true, honest verifier will in fact be convinced it is true by honest prover*
- **Soundness**: *If statement is false, no cheating prover can convince honest verifier it is true, except with small probability (soundness error)*
- **Zero Knowledge**: *If statement is true, no verifier learns anything other than the fact that the statement is true*

Notes:

- "Proof" is not the same as in the mathematical sense

Definitions

- **Prover**: *One proving that they have knowledge of something*
- **Verifier**: *One verifying that prover really does have knowledge*
- **Completeness**: *If statement is true, honest verifier will in fact be convinced it is true by honest prover*
- **Soundness**: *If statement is false, no cheating prover can convince honest verifier it is true, except with small probability (soundness error)*
- **Zero Knowledge**: *If statement is true, no verifier learns anything other than the fact that the statement is true*

Notes:

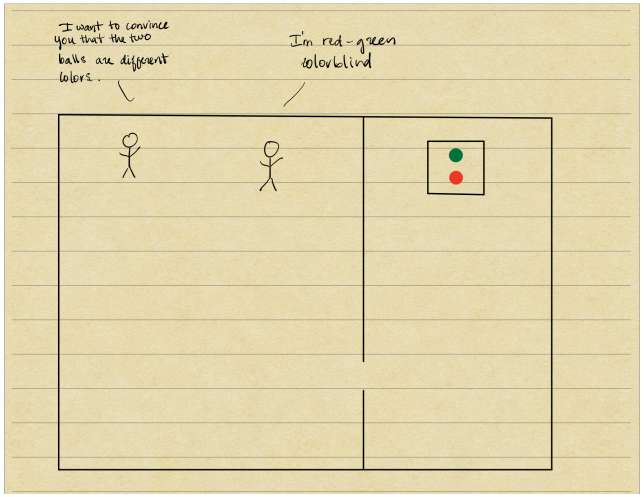
- "Proof" is not the same as in the mathematical sense
- In context of general cryptographic protocols, **Soundness** might be rephrased as *"Someone who is able to impersonate the honest prover with relatively high probability must know the secret"*

Definitions

- **Prover**: *One proving that they have knowledge of something*
- **Verifier**: *One verifying that prover really does have knowledge*
- **Completeness**: *If statement is true, honest verifier will in fact be convinced it is true by honest prover*
- **Soundness**: *If statement is false, no cheating prover can convince honest verifier it is true, except with small probability (soundness error)*
- **Zero Knowledge**: *If statement is true, no verifier learns anything other than the fact that the statement is true*

Notes:

- "Proof" is not the same as in the mathematical sense
- In context of general cryptographic protocols, **Soundness** might be rephrased as *"Someone who is able to impersonate the honest prover with relatively high probability must know the secret"*
- Can formalize above definitions/scheme through Turing machines



Example revisited:

- **Prover**: Person (you) who is trying to convince friend that the colors differ
- **Verifier**: Friend who is colorblind
- **Completeness**: If there really is a difference in the colors of the balls, then for a non-colorblind prover, answering "yes" or "no" to whether or not the ball was swapped will be trivial
- **Soundness**: If there was no difference in the color of the balls, then it would be hard to be correct every time (P(correct) $\sim \frac{1}{2^n}$ with n trials)
- **Zero Knowledge**: Answering "yes" or "no" to whether the balls were switched gives away no information to your colorblind friend about what color each ball is.

Example: Feige - Fiat - Shamir Identification Scheme Preliminaries

Problem (Computational Composite Quadratic Residues)

Instance: A positive integer n that is the product of two unknown distinct primes p and q where $p, q \equiv 3 \pmod 4$, and an integer $x \in \mathbb{Z}_n^*$ such that the Jacobi symbol $\left(\frac{x}{n}\right) = 1$.

Question: Find $y \in \mathbb{Z}_n^*$ such that $y^2 \equiv \pm x \pmod n$.

Setup:

- $n = p \cdot q$ is public while p, q are private
- Choose random $S_1, ..., S_k \in \mathbb{Z}_n$
- For $1 \leq j \leq k$, compute $I_j = \pm 1/S_j^2 \pmod n$, where the sign is chosen randomly
- $\mathbf{I} = (I_1, ..., I_k)$ is the public key and $\mathbf{S} = (S_1, ..., S_k)$ is the private key

Example: Feige - Fiat - Shamir Identification Scheme

Recall $\mathbf{I} = (I_1, ..., I_k)$ is the public key and $\mathbf{S} = (S_1, ..., S_k)$ is the private key

Protocol (Feige - Fiat - Shamir Identification Scheme)

Repeat the following steps until verifier is convinced:

com *Prover chooses random* $R \in \mathbb{Z}_n$ *and computes* $X = \pm R^2 \pmod n$ *with random sign, and sends this* X *to verifier*

chal *Verifier sends prover a random boolean vector* $\mathbf{E} = (E_1, ..., E_k) \in \{0, 1\}^k$

resp *Prover computes* $Y = R \prod_{\{j:E_j=1\}} S_j \pmod n$ *and sends this to verifier*

verif *Verifier verifies that* $X = \pm Y^2 \prod_{\{j:E_j=1\}} I_j \pmod n$. *If so, then verifier "accepts", otherwise verifier "rejects".*

Example: Feige - Fiat - Shamir Identification Scheme

Recall $\mathbf{I} = (I_1, ..., I_k)$ is the public key and $\mathbf{S} = (S_1, ..., S_k)$ is the private key

Protocol (Feige - Fiat - Shamir Identification Scheme)

Repeat the following steps until verifier is convinced:

com *Prover chooses random $R \in \mathbb{Z}_n$ and computes $X = \pm R^2 \pmod n$ with random sign, and sends this X to verifier*

chal *Verifier sends prover a random boolean vector $\mathbf{E} = (E_1, ..., E_k) \in \{0, 1\}^k$*

resp *Prover computes $Y = R \prod_{\{j:E_j=1\}} S_j \pmod n$ and sends this to verifier*

verif *Verifier verifies that $X = \pm Y^2 \prod_{\{j:E_j=1\}} I_j \pmod n$. If so, then verifier "accepts", otherwise verifier "rejects".*

The point is that the prover wants to convince the verifier that the prover does in fact know the inverses of the S_j^2s, meaning that the prover does know S_j for each j, and that the prover does not want to simply tell the verifier the S_js.

Example: Feige - Fiat - Shamir Identification Scheme Completeness

Recall:

- $\mathbf{I} = (I_1, ..., I_k)$ is the public key and $\mathbf{S} = (S_1, ..., S_k)$ is the private key
- **Completeness**: If statement is true, honest verifier will in fact be convinced it is true by honest prover

resp Prover computes $Y = R \prod_{\{j:E_j=1\}} S_j \pmod{n}$ and sends this to verifier

verif Verifier verifies that $X = \pm Y^2 \prod_{\{j:E_j=1\}} I_j \pmod{n}$. If so, then verifier "accepts", otherwise verifier "rejects".

Completeness

$Y^2 \prod_{\{j:E_j=1\}} I_j \equiv (R \prod_{\{j:E_j=1\}} S_j)^2 (\prod_{\{j:E_j=1\}} I_j) \pmod{n}$

$\equiv R^2 (\prod_{\{j:E_j=1\}} S_j^2 I_j)) \equiv \pm R^2 \equiv \pm X \pmod{n}$

Example: Feige - Fiat - Shamir Identification Scheme Soundness

Recall:

- **Soundness**: If statement is false, no cheating prover can convince honest verifier it is true, except with small probability (soundness error)

resp Prover computes $Y = R \prod_{\{j:E_j=1\}} S_j \pmod{n}$ and sends this to verifier

verif Verifier verifies that $X = \pm Y^2 \prod_{\{j:E_j=1\}} I_j \pmod{n}$. If so, then verifier "accepts", otherwise verifier "rejects".

A dishonest prover may try to guess the challenge $\mathbf{E} = (E_1, ..., E_k) \in \{0,1\}^k$ ahead of time. Then given a particular $\mathbf{E}$, choosing X to fool the verifier is straight forward:

Soundness

Suppose the adversary can guess ahead of time $\mathbf{E}$, in step 1. Then the adversary chooses $X \equiv Y^2(\prod_{\{j:E_j=1\}} I_j)$ for a random $Y \in \mathbb{Z}_n$, and sends X to the verifier as the adversary's commitment. The verifier sends back the correctly guessed $\mathbf{E}$ as the challenge, and the adversary sends back Y as the response. Then the verifier verifies that $Y^2(\prod_{\{j:E_j=1\}} I_j) \equiv X$, as this was how X was initially chosen by the adversary.

Example: Feige - Fiat - Shamir Identification Scheme Soundness continued

Recall:

- **Soundness**: If statement is false, no cheating prover can convince honest verifier it is true, except with small probability (soundness error)

Soundness continued

The ability of the adversary to fool the verifier depends upon correctly guessing $\mathbf{E} = (E_1, ..., E_k) \in \{0,1\}^k$ ahead of time. The probability of correctly guessing $\mathbf{E}$ is $\frac{1}{2^k}$, and so the probability of correctly guessing $\mathbf{E}$ for t rounds of the protocol is $\frac{1}{2^{tk}}$, which is exceedingly low.

Example: Feige - Fiat - Shamir Identification Scheme Soundness continued

Recall:

- **Soundness**: If statement is false, no cheating prover can convince honest verifier it is true, except with small probability (soundness error)

Soundness continued

The ability of the adversary to fool the verifier depends upon correctly guessing $\mathbf{E} = (E_1, ..., E_k) \in \{0,1\}^k$ ahead of time. The probability of correctly guessing $\mathbf{E}$ is $\frac{1}{2^k}$, and so the probability of correctly guessing $\mathbf{E}$ for t rounds of the protocol is $\frac{1}{2^{tk}}$, which is exceedingly low.

Final notes: