Zero-Knowledge Proof

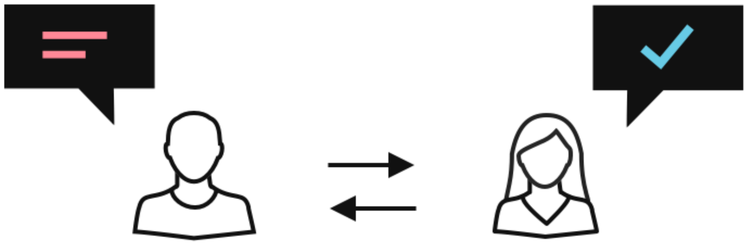
-- A Method in Blockchain

LING Zong, Ph. D.
Senior Software Engineer / Scientist
IBM Almaden Research Center
San Jose, California, U.S.A.

Lecture Outline

Definition

The Method（1/3）



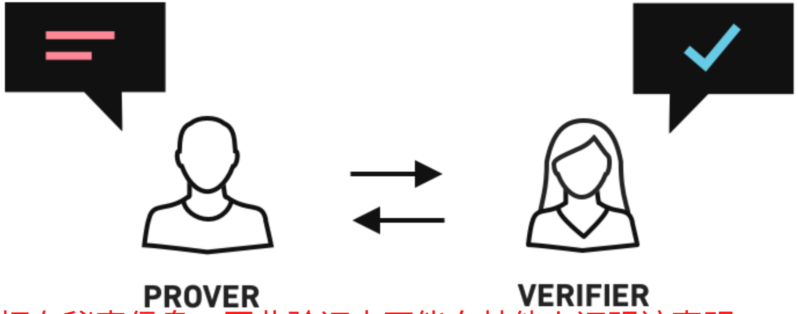
在密码学中，零知识证明或零知识协议是一种方法，通过这种方法，一方(证明者Peggy)可以向另一方(验证者Victor)证明她知道一个值x，除了她知道这个值x之外，不传递任何信息

- In cryptography, a **zero-knowledge proof** or zero-knowledge protocol is a method by which one party (the prover Peggy) can prove to another party (the verifier Victor) that she knows a value x, without conveying any information apart from the fact that she knows the value x.

另一种理解方式是:交互式零知识证明需要证明其知识的个人(或计算机系统)与验证证明的个人之间的交互

  - Another way of understanding this would be: Interactive zero-knowledge proofs require **interaction** between the individual (or computer system) proving their knowledge and the individual validating the proof.

The Method（2/3）



如果证明声明需要证明方知道一些秘密信息，则该定义意味着，由于验证方不拥有秘密信息，因此验证方不能向其他人证明该声明

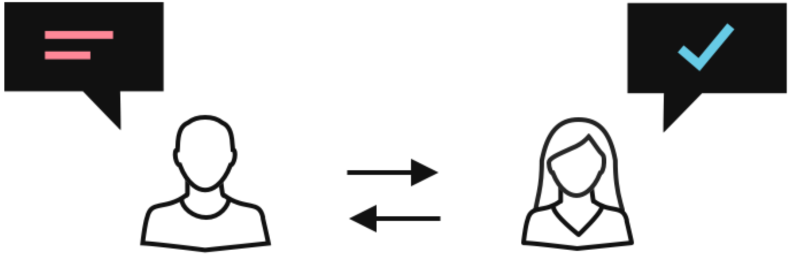
- If proving the statement requires knowledge of some **secret information** on the part of the prover, the definition implies that the verifier will not be able to prove the statement in turn to anyone else, since the verifier does not possess the secret information.

  注意,声明证明必须包括断言证明方有这样的知识(否则,声明不会在零知识中证明,因为在协议的最后，验证器将获得的额外信息,即证明方具有所需的秘密信息)

  - Notice that the statement being proved must include the assertion that **the prover has such knowledge** (otherwise, the statement would not be proved in zero-knowledge, since at the end of the protocol the verifier would gain the additional information that the prover has knowledge of the required secret information).
  - If the statement consists only of the fact that prover possesses the secret information, it is a special case known as zero-knowledge proof of knowledge, and it nicely illustrates the essence of the notion of zero-knowledge proofs: proving that one has knowledge of certain information is trivial if one is allowed to simply reveal that information; **the challenge is proving that one has such knowledge without revealing the secret information or anything else**.

  如果声明中只包含证明方拥有秘密信息的事实,它是一个特例称为零知识的证明,它很好地说明了零知识证明的概念的本质:证明某人有某些信息的知识是微不足道的，如果他被允许简单地显示信息;挑战是证明一个人有这样的知识，而不用透露秘密信息或其他任何东西

The Method（3/3）

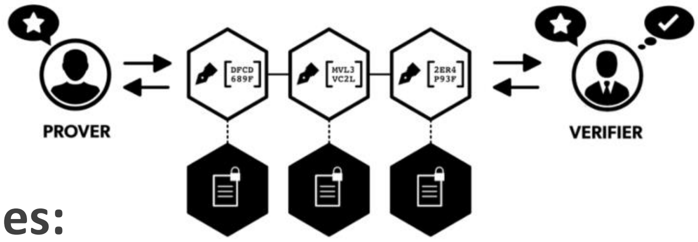


对于零知识证明,协议必须要求验证方的交互输入,通常以挑战的形式,这样当且仅当该声明为真时(也就是说,如果证明方确实有宣称的知识)，证明方的反应将使验证者信服

- For zero-knowledge proofs of knowledge, the protocol must necessarily **require interactive input** from the verifier, usually in the form of a challenge or challenges such that the responses from the prover will convince the verifier if and only if the statement is true (i.e., if the prover does have the claimed knowledge).
  - This is clearly the case, since otherwise the verifier could record the execution of the protocol and replay it to someone else: if this were accepted by the new party as proof that the replaying party knows the secret information, then the new party's acceptance is either justified—the replayer does know the secret information—which means that the protocol leaks knowledge and is not zero-knowledge, or it is spurious—i.e. leads to a party accepting someone's proof of knowledge who does not actually possess it.

这显然是事实，否则验证方可以记录协议的执行并回放给其他人:如果这作为重播方知道秘密信息的证据而被新的一方接受，那么新的一方的接受要么是正当的--重播者确实知道秘密信息--这意味着协议泄露了知识，而不是零知识，要么它是虚假的——也就是说，导致一方接受了某人的知识证明，但他实际上并不拥有它

Definition (1/2)



PROOFS AND SECRET DATA

零知识证明必须满足三个性质

A zero-knowledge proof must satisfy three properties:

· 完整性:如果该声明为真，诚实的验证者(即正确遵循协议的验证者)将被诚实的证明者的事实说服
· 可靠性:如果陈述是假的，除了有很小的可能性外，任何有欺骗行为的证明者都无法说服诚实的验证者该陈述是真的。
· 零知识:如果该命题为真，除了该命题为真这一事实外，验证者不会学到任何东西。换句话说，仅仅知道陈述(而不是秘密)就足以想象这样一个场景:证明者知道秘密。

- **Completeness**: if the statement is true, the honest verifier (that is, one following the protocol properly) will be convinced of this fact by an honest prover.
- **Soundness**: if the statement is false, no cheating prover can convince the honest verifier that it is true, except with some small probability.
- **Zero-knowledge**: if the statement is true, no verifier learns anything other than the fact that the statement is true. In other words, just knowing the statement (not the secret) is sufficient to imagine a scenario showing that the prover knows the secret.

  这是通过显示每个验证者都有一些模拟器来形式化的，这些模拟器只给出要被证明的语句(不访问验证者)，可以产生一个“看起来像”诚实的证明者和有问题的验证者之间的交互记录

  - This is formalized by showing that every verifier has some *simulator* that, given only the statement to be proved (and no access to the prover), can produce a transcript that "looks like" an interaction between the honest prover and the verifier in question.

➢前两个是更一般的交互式证明系统的性质。第三，是什么使证明零知识。
➢零知识证明在数学意义上不是证明因为有小概率，稳健性误差，一个作弊的证明者将能够使验证者相信一个虚假的陈述。
➢换句话说，零知识证明是概率“证明”，而不是确定的证明。但是，有一些技术可以将稳健性误差减小到可以忽略的小值。

- The first two of these are properties of more general interactive proof systems. The third is what makes the proof zero-knowledge.
- Zero-knowledge proofs are not proofs in the mathematical sense of the term because there is some small probability, the *soundness error*, that a cheating prover will be able to convince the verifier of a false statement.
- In other words, zero-knowledge proofs are **probabilistic "proofs"** rather than deterministic proofs. However, there are techniques to decrease the soundness error to negligibly small values.

Definition (2/2)

A **formal definition** of zero-knowledge has to use some computational model, the most common one being that of a Turing machine. Let ***P*，*V*，** and ***S*** be Turing machines. An interactive proof system with （***P*，*V***）for a language ***L*** is zero-knowledge if for any probabilistic polynomial time (PPT) verifier ***V*** there exists a PPT simulator ***S*** such that

$$\forall x \in L, z \in \{0,1\}^*, \mathrm{View}_{\hat{V}}\left[P(x) \leftrightarrow \hat{V}(x,z)\right] = S(x,z)$$

Where $\mathrm{View}_{\hat{V}}\left[P(x) \leftrightarrow \hat{V}(x,z)\right]$ is a record of the interactions between $P(x)$ and $\hat{V}(x,z)$. The prover ***P*** is modeled as having unlimited computation power (in practice, ***P*** usually is a probabilistic Turing machine). Intuitively, the definition states that an interactive proof system （***P*，*V***） is zero-knowledge if for any verifier $\hat{V}$ there exists an efficient simulator ***S*** (depending on $\hat{V}$) that can reproduce the conversation between ***P*** and $\hat{V}$ on any given input. The auxiliary string z in the definition plays the role of "prior knowledge" (including the random coins of $\hat{V}$). The definition implies that $\hat{V}$ cannot use any prior knowledge string z to mine information out of its conversation with ***P***, because if ***S*** is also given this prior knowledge then it can reproduce the conversation between $\hat{V}$ and ***P*** just as before.

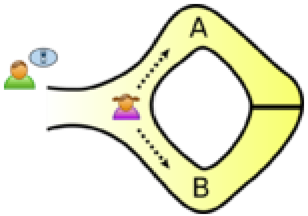
The definition given is that of perfect zero-knowledge. Computational zero-knowledge is obtained by requiring that the views of the verifier $\hat{V}$ and the simulator are only **computationally indistinguishable**, given the auxiliary string.

Abstract examples

The Ali Baba cave

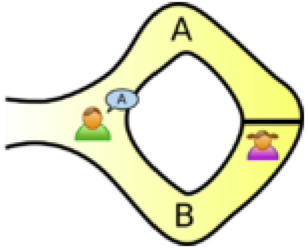
There is a well-known story presenting the fundamental ideas of zero-knowledge proofs, first published by Jean-Jacques Quisquater and others in their paper "How to Explain Zero-Knowledge Protocols to Your Children". It is common practice to label the two parties in a zero-knowledge proof as Peggy (the **prover** of the statement) and Victor (the **verifier** of the statement).

Jean-Jacques Quisquater等人在他们的论文《如何向孩子解释零知识协议》中首次发表了一个讲述零知识证明基本思想的著名故事。通常的做法是将零知识证明的双方分别称为Peggy(声明的证明者)和Victor(声明的验证者)。



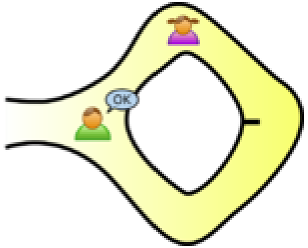
- In this story, Peggy has uncovered the secret word used to open a magic door in a cave.
  - The cave is shaped like a ring, with the entrance on one side and the magic door blocking the opposite side.
  - Victor wants to know whether Peggy knows the secret word;
  - but Peggy, being a very private person, does not want to reveal her knowledge (the secret word) to Victor or to reveal the fact of her knowledge to the world in general.

在这个故事中，佩吉发现了一个山洞中用来打开魔法门的秘密字。
· 洞穴形状像个圆环，一边是入口，另一边是魔法门。
· 维克多想知道佩吉是否知道这个秘密字;
· 但是佩吉是个非常隐私的人，她不想把知识(秘密词)告诉维克多，也不想把她知道的事实告诉全世界。



- They label the left and right paths from the entrance A and B. First, Victor waits outside the cave as Peggy goes in. Peggy takes either path A or B; Victor is not allowed to see which path she takes.
  - Then, Victor enters the cave and shouts the name of the path he wants her to use to return, either A or B, chosen at random.
  - Providing she really does know the magic word, this is easy: she opens the door, if necessary, and returns along the desired path.

他们从入口A和b开始，给左右的路径贴上标签。首先，维克多在洞口等着，佩吉进去了。佩吉选择A或B;维克托不允许看到她走哪条路。
〇然后，维克多进入洞穴，大声呼喊着他想让她返回的路径的名字，可以是A，也可以是B，随机选择
〇如果她真的知道这个神奇的词，这很容易:如果有必要，她会打开这扇门，然后沿着想要的路径返回



- However, suppose she did not know the word. Then, she would only be able to return by the named path if Victor were to give the name of the same path by which she had entered.

但是，假设她不知道这个词。然后，只有在Victor给出她进入的同一路径的名称时，她才能通过已命名的路径返回

  - Since Victor would choose A or B at random, she would have a 50% chance of guessing correctly.

〇既然Victor会随机选择A或B，她有50%的机会猜对。

  - If they were to repeat this trick many times, say 20 times in a row, her chance of successfully anticipating all of Victor's requests would become vanishingly small (about one in a million).

〇如果他们重复这个把戏很多次，比如说连续20次，她成功预测到Victor所有要求的机会将会变得微乎其微(大约是百万分之一)

- Thus, if Peggy repeatedly appears at the exit Victor names, he can conclude that it is very probable—astronomically probable—that Peggy does in fact know the secret word.

因此，如果佩吉反复出现在维克托命名的出口中，他就可以断定，佩吉确实知道这个秘密单词，这是非常有可能的——非常有可能。

Two balls and the color-blind friend

这个例子需要两个具有不同颜色的相同对象，比如两个彩色的球，它被认为是交互式零知识证明如何工作的最简单的解释之一。在2017年9月的区块链相关会议上，软件工程师Konstantinos Chalkias和Mike Hearn首次现场演示了这种卡片，其灵感来自于Oded Goldreich教授的工作，他使用了两种不同颜色的卡片

- This example requires two identical objects with different colors, such as two colored balls, and it is considered one of the easiest explanations of how interactive zero-knowledge proofs work. It was first demonstrated live by software engineers Konstantinos Chalkias and Mike Hearn at a blockchain related conference in September 2017 and is inspired by the work of Prof. Oded Goldreich, who used two differently coloured cards.

想象一下，你的朋友是色盲，你有两个球：一个红的，一个绿的，但其他方面都一样。对你的朋友来说，它们看起来完全一样，他怀疑它们是否真的可以区分。你想向他证明它们实际上是不同的颜色，但没有别的，因此你不透露哪一个是红色的，哪一个是绿色的

  - Imagine your friend is color-blind and you have two balls: one red and one green, but otherwise identical. To your friend they seem completely identical and he is skeptical that they are actually distinguishable. You want to *prove to him they are in fact differently-colored*, but nothing else, thus you do not reveal which one is the red and which is the green.

这就是证明系统。你把两个球给你的朋友，他把它们放在他背后。接下来，他拿起其中一个球，把它从背后拿出来展示。然后这个球再次被放在他背后，然后他选择只露出两个球中的一个，然切换到另一个球的概率是50%。他会问你，“我换球了吗？”然后，这整个过程按需要重复进行

- **Here is the proof system. You give the two balls to your friend and he puts them behind his back. Next, he takes one of the balls and brings it out from behind his back and displays it. This ball is then placed behind his back again and then he chooses to reveal just one of the two balls, switching to the *other* ball with probability 50%. He will ask you, "Did I switch the ball?" This whole procedure is then repeated as often as necessary.**

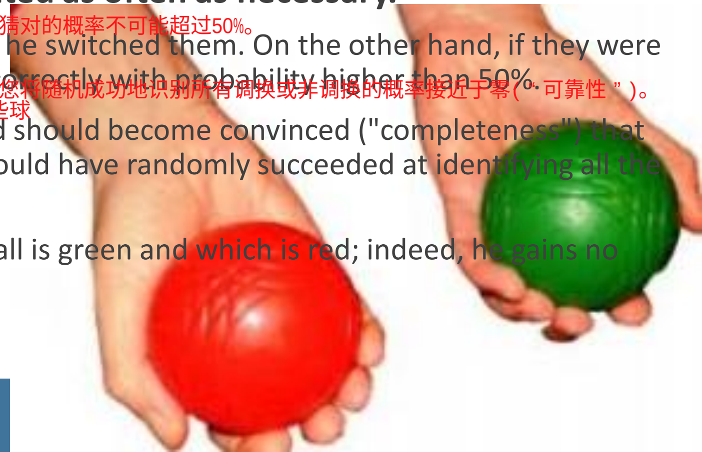
○通过观察它们的颜色，你当然可以肯定地说他是否调换了它们。另一方面，如果它们是同样的颜色，因此无法区分，你猜对的概率不可能超过50%。

  - By looking at their colors, you can of course say with certainty whether or not he switched them. On the other hand, if they were the same color and hence indistinguishable, there is no way you could guess correctly with probability higher than 50%.

○如果你和你的朋友重复这个“证明”多次(例如128次)，你的朋友就会相信(“完整性”)这些球的颜色确实不同;否则，您将随机成功地识别所有调换或非调换的概率接近于零(“可靠性”)。

○上面的证明是零知识的，因为你的朋友从来不知道哪个球是绿色的，哪个是红色的;事实上，他根本不知道如何分辨这些球

  - If you and your friend repeat this "proof" multiple times (e.g. 128), your friend should become convinced ("completeness") that the balls are indeed differently colored; otherwise, the probability that you would have randomly succeeded at identifying all the switch/non-switches is close to zero ("soundness").
  - The above proof is ***zero-knowledge*** because your friend never learns which ball is green and which is red; indeed, he gains no knowledge about how to distinguish the balls.



Where's Wally?

沃利在哪儿?(或者Waldo在哪里?)是一本图画书，在这本书中，读者要挑战自己去寻找一个名叫Wally的小角色，这个小角色隐藏在一张有很多其他角色的双层展开的页面上。图片的设计使人们很难找到沃利

- *Where's Wally?* (or *Where's Waldo?*) is a picture book where the reader is challenged to find a small character called Wally hidden somewhere on a double-spread page that is filled with many other characters. The pictures are designed so that it is hard to find Wally.

  假设你是专业人士，沃利呢?解算器。一个公司来找你，问 " 沃利去哪了? " 他们需要解决的书。公司想要你证明你确实是专业人士，沃利呢?因此，请您从他们书中的一张图片中找到沃利。问题是你不想无偿为他们工作

  - Imagine that you are a professional *Where's Wally?* solver. A company comes to you with a *Where's Wally?* book that they need solved. The company wants you to prove that you are actually a professional *Where's Wally?* solver and thus asks you to find Wally in a picture from their book. The problem is that you don't want to do work for them without being paid.

  你和公司都想合作，但彼此不信任。如果不为他们做免费的工作，似乎不可能满足公司的需求,但实际上有一个零知识证明，你可以向公司证明你知道沃利在图中但没有透露他们如何发现他,或者他在哪里

  - Both you and the company **want to cooperate**, but you **don't trust each other**. It doesn't seem like it's possible to satisfy the company's demand without doing free work for them, but in fact there is a zero-knowledge proof which allows you to prove to the company that you know where Wally is in the picture without revealing to them how you found him, or where he is.

- **The proof goes as follows**:

  你让公司代表转过身来，然后你把一张很大的硬纸板放在图片上，这样硬纸板的中心就在Wally的上方

  - You ask the company representative to turn around, and then you place a very large piece of cardboard over the picture such that the center of the cardboard is positioned over Wally.

  你在纸板的中央剪出一个小窗口，这样就可以看到沃利了

  - You cut out a small window in the center of the cardboard such that Wally is visible.

  现在你可以让公司代表转过身来，看中间有个洞的大块纸板，通过这个洞可以看到Wally

  - You can now ask the company representative to turn around and view the large piece of cardboard with the hole in the middle, and observe that Wally is visible through the hole.

  纸板大到他们不能确定书在纸板下面的位置。然后你让代表转过身来，这样你就可以拿走硬纸板，把书还给他

  - The cardboard is large enough that they cannot determine the position of the book under the cardboard. You then ask the representative to turn back around so that you can remove the cardboard and give back the book.

- As described, this proof is an illustration only, and not completely rigorous. The company representative would need to be sure that you didn't smuggle a picture of Wally into the room. Something like a tamper-proof glovebox might be used in a more rigorous proof. The above proof also results in the body position of Wally being leaked to the company representative, which may help them find Wally if his body position changes in each *Where's Wally?* puzzle.

如前所述，这个证明只是一个例证，并不完全严格。公司代表需要确保你没有把沃利的照片带进房间。在更严格的证明中可能会用到防篡改手套盒之类的东西。以上证据也导致了Wally的体位泄露给了公司代表，如果Wally的体位在各个地方发生变化，可能会帮助他们找到Wally。谜题

Practical examples

Discrete log of a given value (1/2)

我们可以将这些想法应用到更现实的密码学应用中。
Peggy想要向Victor证明她知道给定组中给定值的离散对数。

We can apply these ideas to a more realistic cryptography application.
Peggy wants to prove to Victor that she knows the discrete log of a given value in a given group.

For example, given a value y, a large prime p and a generator g, she wants to prove that she knows a value x such that $g^x \mod p = y$, without revealing x. Indeed, knowledge of x could be used as a proof of identity, in that Peggy could have such knowledge because she chose a random value x that she didn't reveal to anyone, computed $y = g^x \mod p$ and distributed the value of y to all potential verifiers, such that at a later time, proving knowledge of x is equivalent to proving identity as Peggy.

The protocol proceeds as follows: in each round, Peggy generates a random number r, computes $C = g^r \mod p$ and discloses this to Victor. After receiving C, Victor randomly issues one of the following two requests: he either requests that Peggy discloses the value of r, or the value of $(x + r) \mod (p - 1)$. With either answer, Peggy is only disclosing a random value, so no information is disclosed by a correct execution of one round of the protocol.

Victor can verify either answer; if he requested r, he can then compute $g^r \mod p$ and verify that it matches C. If he requested $(x + r) \mod (p - 1)$, he can verify that C is consistent with this, by computing $g^{(x+r) \bmod (p-1)} \mod p$ and verifying that it matches $C \cdot y \mod p$. If Peggy indeed knows the value of x, she can respond to either one of Victor's possible challenges.

If Peggy knew or could guess which challenge Victor is going to issue, then she could easily cheat and convince Victor that she knows x when she does not: if she knows that Victor is going to request r, then she proceeds normally: she picks r, computes $C = g^r \mod p$ and discloses C to Victor; she will be able to respond to Victor's challenge. On the other hand, if she knows that Victor will request $(x + r) \mod (p - 1)$, then she picks a random value r', computes $C' = g^{r'} \cdot (g^x)^{-1} \mod p$, and discloses C' to Victor as the value of C that he is expecting. When Victor challenges her to reveal $(x + r) \mod (p - 1)$, she reveals r', for which Victor will verify consistency, since he will in turn compute $g^{r'} \mod p$, which matches $C' \cdot y$, since Peggy multiplied by the inverse of y.

However, if in either one of the above scenarios Victor issues a challenge other than the one she was expecting and for which she manufactured the result, then she will be unable to respond to the challenge under the assumption of infeasibility of solving the discrete log for this group. If she picked r and disclosed $C = g^r \mod p$, then she will be unable to produce a valid $(x + r) \mod (p - 1)$ that would pass Victor's verification, given that she does not know x. And if she picked a value r' that poses as $(x + r) \mod (p - 1)$, then she would have to respond with the discrete log of the value that she disclosed – but Peggy does not know this discrete log, since the value C she disclosed was obtained through arithmetic with known values, and not by computing a power with a known exponent.

Thus, a cheating prover has a 0.5 probability of successfully cheating in one round. By executing a large enough number of rounds, the probability of a cheating prover succeeding can be made arbitrarily low.
因此，一个作弊证明者有0.5的概率在一轮中成功作弊。通过执行足够多的轮数，作弊证明者成功的概率可以变得任意低

Discrete log of a given value (2/2)

Short summary

Peggy proves to know the value of x (for example her password).

1. Peggy calculates first for one time the value $y = g^x \bmod p$ and transfer the value to Victor.
2. Peggy repeatedly calculates a random value r and $C = g^r \bmod p$. She transfers the value C to Victor.
3. Victor asks Peggy to calculate and transfer the value $(x + r) \bmod (p - 1)$ or simply to transfer the value r. in the first case Victor verifies $(C \cdot y) \bmod p \equiv g^{(x+r) \bmod (p-1)} \bmod p$. In the second case he verifies $C \equiv g^r \bmod p$.

The value $(x + r) \bmod (p - 1)$ can be seen as the encrypted value of $x \bmod (p - 1)$. If r is true random, equally distributed between zero and $(p - 1)$, this does not leak any information about x (see one-time pad).

Hamiltonian cycle for a large graph (1/2)

在这个场景中，Peggy知道一个大图G的哈密顿循环，Victor知道G，但不知道这个循环(例如，Peggy生成了G并告诉了他)。在给定一个大图的情况下，寻找哈密顿循环被认为在计算上是不可行的，因为它对应的决策版本已知是np完备的。佩吉会证明她知道这个周期，而不只是简单地透露它(也许维克多有兴趣购买，但想先确认，又或者佩吉是唯一知道这个信息并向维克多证明自己身份的人)

- In this scenario, Peggy knows a Hamiltonian cycle for a large graph *G*. Victor knows *G* but not the cycle (e.g., Peggy has generated *G* and revealed it to him.) Finding a Hamiltonian cycle given a large graph is believed to be computationally infeasible, since its corresponding decision version is known to be NP-complete. Peggy will prove that she knows the cycle without simply revealing it (perhaps Victor is interested in buying it but wants verification first, or maybe Peggy is the only one who knows this information and is proving her identity to Victor).

为了表明佩吉了解这个汉密尔顿循环，她和维克多玩了几轮游戏

- To show that Peggy knows this Hamiltonian cycle, she and Victor play several rounds of a game.
- At the beginning of each round, Peggy creates *H*, a graph which is isomorphic to *G* (i.e. *H* is just like *G* except that all the vertices have different names). Since it is trivial to translate a Hamiltonian cycle between isomorphic graphs with known isomorphism, if Peggy knows a Hamiltonian cycle for *G* she also must know one for *H*.

在每一轮的开始，Peggy创建H，这是一个与G同构的图(也就是说，H就像G一样，除了所有的顶点都有不同的名称)。由于在具有已知同构的同构图之间转换哈密顿循环是很容易的，如果Peggy知道G的哈密顿循环，她也一定知道H的哈密顿循环

- Peggy commits to *H*. She could do so by using a cryptographic commitment scheme. Alternatively, she could number the vertices of *H*, then for each edge of *H* write on a small piece of paper containing the two vertices of the edge and then put these pieces of paper face down on a table. The purpose of this commitment is that Peggy is not able to change *H* while at the same time Victor has no information about *H*.

佩吉向h承诺，她可以使用一种加密的承诺方案。或者，她可以给H的顶点编号，然后把H的每条边都写在一张包含这条边的两个顶点的小纸片上然后把这些纸面朝下放在桌子上。这个承诺的目的是，Peggy不能改变H，而Victor没有关于H的信息

- Victor then randomly chooses one of two questions to ask Peggy. He can either ask her to show the isomorphism between *H* and *G* (see graph isomorphism problem), or he can ask her to show a Hamiltonian cycle in *H*.

然后维克多从两个问题中随机选择一个问佩吉。他可以让她展示H和G之间的同构(参见图同构问题)，或者他可以让她展示H中的哈密顿循环

- If Peggy is asked to show that the two graphs are isomorphic, she first uncovers all of *H* (e.g. by turning over all pieces of papers that she put on the table) and then provides the vertex translations that map *G* to *H*. Victor can verify that they are indeed isomorphic.

如果佩吉被要求证明这两个图是同构的,她首先揭示H(例如,通过翻转所有她放在桌上的纸条),然后提供了图G的顶点翻译H .维克多可以确认他们确实是同构的

- If Peggy is asked to prove that she knows a Hamiltonian cycle in *H*, she translates her Hamiltonian cycle in *G* onto *H* and only uncovers the edges on the Hamiltonian cycle. This is enough for Victor to check that *H* does indeed contain a Hamiltonian cycle.

如果Peggy被要求证明她知道H中的哈密顿循环，她会把G中的哈密顿循环转化为H，并且只揭示哈密顿循环的边。这足以让维克多检验H确实包含一个哈密顿循环

Hamiltonian cycle for a large graph (2/2)

Completeness 如果佩吉知道哈密顿循环在G,她可以很容易地满足维克多对从G产生H的图同构的要求(她致力于在第一步)或哈密顿循环H(她可以通过将同构应用到G中的循环来构造)

If Peggy does know a Hamiltonian cycle in G, she can easily satisfy Victor's demand for either the graph isomorphism producing H from G (which she had committed to in the first step) or a Hamiltonian cycle in H (which she can construct by applying the isomorphism to the cycle in *G*).

佩吉的答案并没有揭示G中原来的哈密顿循环.每一轮循环,Victor将只学习H与G的同构或者H中的哈密顿循环.他需要一个H的两个答案来发现G的循环，所以只要Peggy每轮都能产生一个不同的H，这个信息就仍然是未知的.如果佩吉不知道G中的哈密顿循环，但不知怎么地提前知道了每一轮维克多会要求看什么，那么她就可以作弊。例如，如果佩吉事先知道维克多会要求看H中的哈密顿循环，那么她就可以为一个不相关的图生成一个哈密顿循环。同样地，如果佩吉事先知道维克多会要求看同构，那么她就可以简单地生成一个同构图H(她也不知道其中的哈密顿循环)。Victor可以自己模拟这个协议(没有Peggy)，因为他知道他要看什么。因此，维克多不能从每一轮的哈密顿循环中得到关于G中的哈密顿循环的信息

Zero-knowledge

Peggy's answers do not reveal the original Hamiltonian cycle in *G*. Each round, Victor will learn only *H*'s isomorphism to *G* or a Hamiltonian cycle in *H*. He would need both answers for a single *H* to discover the cycle in *G*, so the information remains unknown as long as Peggy can generate a distinct *H* every round. If Peggy does not know of a Hamiltonian Cycle in *G*, but somehow knew in advance what Victor would ask to see each round then she could cheat. For example, if Peggy knew ahead of time that Victor would ask to see the Hamiltonian Cycle in *H* then she could generate a Hamiltonian cycle for an unrelated graph. Similarly, if Peggy knew in advance that Victor would ask to see the isomorphism then she could simply generate an isomorphic graph *H* (in which she also does not know a Hamiltonian Cycle). Victor could simulate the protocol by himself (without Peggy) because he knows what he will ask to see. Therefore, Victor gains no information about the Hamiltonian cycle in *G* from the information revealed in each round.

Soundness 如果Peggy不知道这个信息，她可以猜测Victor会问什么问题，并生成一个与G同构的图，或者一个与G不相关的图的哈密顿循环，但是由于她不知道G的哈密顿循环，所以她不能同时做两个。根据这个猜测，她愚弄维克托的机会是2^-n，其中n是轮数。出于所有现实的目的，用这种方法用合理的回合数击败零知识证明是不可能的

If Peggy does not know the information, she can guess which question Victor will ask and generate either a graph isomorphic to *G* or a Hamiltonian cycle for an unrelated graph, but since she does not know a Hamiltonian cycle for *G* she cannot do both. With this guesswork, her chance of fooling Victor is 2^{-n}, where n is the number of rounds. For all realistic purposes, it is infeasibly difficult to defeat a zero knowledge proof with a reasonable number of rounds in this way.

Applications

认证系统

Authentication systems

零知识证明(ZKP)的研究是由身份验证系统推动的，其中一方希望通过一些秘密信息(比如密码)向另一方证明其身份，但不希望另一方了解有关该秘密的任何信息

Research in zero-knowledge proofs (ZKP) has been motivated by authentication systems where one party wants to prove its identity to a second party via some secret information (such as a password) but doesn't want the second party to learn anything about this secret.

这被称为 " 知识的零知识证明 " 。然而，在许多用于知识的零知识证明的方案中，密码通常太小或不够随机

This is called a "zero-knowledge proof of knowledge". However, a password is typically too small or insufficiently random to be used in many schemes for zero-knowledge proofs of knowledge.

零知识密码证明是一种特殊的零知识知识证明，它解决了密码的有限大小问题

A zero-knowledge password proof is a special kind of zero-knowledge proof of knowledge that addresses the limited size of passwords.

道德行为

Ethical behavior

●零知识证明在加密协议中的一个用途是在保持隐私的同时强制诚实的行为。
●大体上，这个想法是强迫用户使用零知识证明，证明其行为根据协议是正确的。
●由于可靠性，我们知道用户必须真正诚实地行动，才能提供有效的证明。
●由于零知识，我们知道用户在提供证据的过程中不会泄露其秘密的隐私

- One of the uses of zero-knowledge proofs within cryptographic protocols is to enforce honest behavior while maintaining privacy.
- Roughly, the idea is to force a user to prove, using a zero-knowledge proof, that its behavior is correct according to the protocol.
- Because of soundness, we know that the user must really act honestly in order to be able to provide a valid proof.
- Because of zero knowledge, we know that the user does not compromise the privacy of its secrets in the process of providing the proof.

Nuclear disarmament

2016年，普林斯顿等离子物理实验室和普林斯顿大学展示了一种可能适用于未来核裁军谈判的新技术。
它将允许核查人员确认一个物体是否确实是核武器，而无需记录、分享或透露可能是秘密的内部工作方式

In 2016, the Princeton Plasma Physics Laboratory and Princeton University demonstrated a novel technique that may have applicability to future nuclear disarmament talks.

It would allow inspectors to confirm whether or not an object is indeed a nuclear weapon without recording, sharing or revealing the internal workings which might be secret.

Blockchains

ZKPs可用于保证事务是有效的，尽管有关发送方、接收方和其他事务细节的信息仍然隐藏

ZKPs can be used to guarantee that transactions are valid despite the fact that information about the sender, the recipient and other transaction details remain hidden.

零知识协议允许在完全隐私的分布式对等区块链网络上转移资产。在常规区块链事务中，当资产从一方发送到另一方时，该事务的详细信息对网络中的每一方都是可见的。相比之下，在零知识交易中，其他交易方只知道发生了有效的交易，而不知道发送方、接收方、资产类别和数量。消费者的身份和消费金额可以隐藏起来，这样就可以避免抢钱的问题

Zero-knowledge protocols enable the transfer of assets across a distributed, peer-to-peer blockchain network with complete privacy. In regular blockchain transactions, when an asset is sent from one party to another, the details of that transaction are visible to every other party in the network. By contrast, in a zero knowledge transaction, the others only know that a valid transaction has taken place, but nothing about the sender, recipient, asset class and quantity. The identity and amount being spent can remain hidden, and problems such as "front-running" can be avoided.

最突出的使用零知识证明的基于区块链的系统是ZCash，它也是第一个实现zk- snark的加密货币。此后，其他基于区块链的系统也将零知识证明纳入其解决方案，以便在保护用户/交易隐私的同时对交易进行验证。也许其中最著名的是以太坊，它实现了zk- snark作为拜占庭升级的一部分

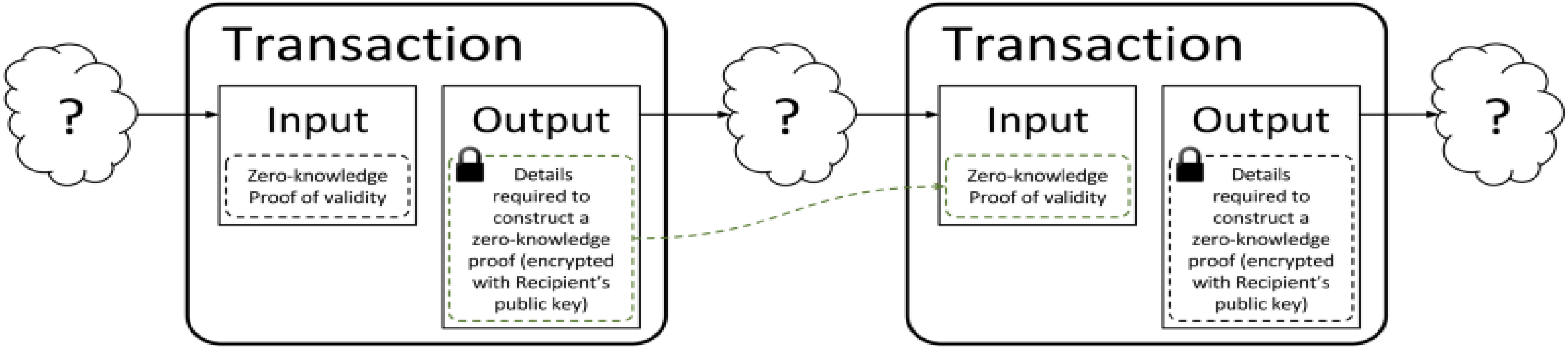
The most prominent blockchain-based system using zero-knowledge proofs is ZCash, which was also the first cryptocurrency to implement zk-SNARKs. Other blockchain-based systems have since also incorporate zero-knowledge proofs into their solutions to allow for transactions to be verified while protecting user/transaction privacy. Probably the best known of which is Ethereum, which implemented zk-SNARKS as part of the Byzantium upgrade.

What are zk-Snarks?

您可能已经偶然发现了zk- snark这个术语。这个词是由Nir Bitansky, Ran Canetti, Alessandro Chiesa & Eran Tromer于2012年提出的,它描述了零知识技术的一种特殊变化。
zk- snark引入了许多创新，使它们可以在区块链中使用。最重要的是，zk- snark减少了证明的大小和验证它们所需的计算量

You might already have stumbled upon the term ‘zk-Snarks’. The term was introduced in 2012 by Nir Bitansky, Ran Canetti, Alessandro Chiesa & Eran Tromer and describes a special variation of the zero-knowledge technique.

zk-SNARKs introduce a number of innovations that render them usable in blockchains. Most importantly, zk-SNARKs reduce the size of the proofs and the computational effort required to verify them.



https://z.cash/zh/technology/zksnarks/

Variants of zero-knowledge

零知识的不同变体可以通过以下方式形式化模拟器输出 " 看起来像 " 实际证明协议执行的直观概念来定义

- Different variants of zero-knowledge can be defined by formalizing the intuitive concept of what is meant by the output of the simulator "looking like" the execution of the real proof protocol in the following ways:

  如果模拟器和证明协议产生的分布是完全相同的，我们就说完全零知识。这就是上面第一个例子中的例子

  - We speak of *perfect zero-knowledge* if the distributions produced by the simulator and the proof protocol are distributed exactly the same. This is for instance the case in the first example above.

  统计零知识意味着分布不一定完全相同，但它们在统计学上是接近的，这意味着它们的统计差异是一个可以忽略的函数

  - *Statistical zero-knowledge* means that the distributions are not necessarily exactly the same, but they are statistically close, meaning that their statistical difference is a negligible function.

  如果没有有效的算法能够区分这两种分布，我们称之为计算零知识

  - We speak of *computational zero-knowledge* if no efficient algorithm can distinguish the two distributions.

Zero knowledge types

知识隐藏在指数中，如上例所示

- Proof of knowledge: the knowledge is hidden in the exponent like in the example shown above.
- Pairing based cryptography: given f(x) and f(y), without knowing x and y, it is possible to compute f($x \times y$).
- Witness indistinguishable proof: verifiers cannot know which witness is used for producing the proof.

  验证者不能知道哪个证人是用来产生证明的
- Multi-party computation: while each party can keep their respective secret, they together produce a result.

  虽然每一方都能保守各自的秘密，但他们一起产生一个结果
- Ring signature: outsiders have no idea which key is used for signing.

  外部人士不知道哪把钥匙是用来签名的

History

History (1/2)

零知识证明最早是由Shafi Goldwasser，Silvio Micali和Charles Rackoff在1985年的论文《交互证明系统的知识复杂性》中提出的

Zero-knowledge proofs were first conceived in **1985** by Shafi Goldwasser, Silvio Micali, and Charles Rackoff in their paper "**The Knowledge Complexity of Interactive Proof-Systems**".

本文引入了交互式证明系统的IP层次结构(见交互式证明系统)，并提出了知识复杂度的概念，知识复杂度是对证明者向验证者传递的证明知识量的度量

This paper introduced the **IP** hierarchy of interactive proof systems (*see* *interactive proof system*) and conceived the concept of *knowledge complexity*, a measurement of the amount of knowledge about the proof transferred from the prover to the verifier.

They also gave the first zero-knowledge proof for a concrete problem, that of deciding quadratic nonresidues mod *m* (this more or less means that there isn't any number *x* where x^2 is "equivalent" to some given number). Together with a paper by László Babai and Shlomo Moran, this landmark paper invented interactive proof systems, for which all five authors won the first Gödel Prize in 1993.

二次非残差问题同时具有NP和co-NP两种算法，属于NP和co-NP的交点。这也适用于随后发现的其他几个零知识证明问题，例如Oded Goldreich证明二素数模数不是Blum整数的一个未发表的证明系统

The quadratic nonresidue problem has both an **NP** and a **co-NP** algorithm, and so lies in the intersection of **NP** and **co-NP**. This was also true of several other problems for which zero-knowledge proofs were subsequently discovered, such as an unpublished proof system by Oded Goldreich verifying that a two-prime modulus is not a Blum integer.

History (2/2)

Oded Goldreich, Silvio Micali和Avi Wigderson进一步证明了这一点，他们表明，假设存在不可破解的加密，我们可以为NP完全图三色着色问题创建一个零知识证明系统。既然NP中的每个问题都可以有效地简化为这个问题，这就意味着在这个假设下，NP中的所有问题都具有零知识证明。这种假设的原因是，如上面的示例所示，它们的协议需要加密。不可破解加密存在的一个常被引用的充分条件是存在单向函数，但可以想象，某些物理方法也可能实现这一点

Oded Goldreich, Silvio Micali, and Avi Wigderson took this one step further, showing that, assuming the existence of unbreakable encryption, one can create a zero-knowledge proof system for the NP-complete graph coloring problem with three colors. Since every problem in **NP** can be efficiently reduced to this problem, this means that, under this assumption, **all problems in NP have zero-knowledge proofs**. The reason for the assumption is that, as in the above example, their protocols require encryption. A commonly cited sufficient condition for the existence of unbreakable encryption is the existence of one-way functions, but it is conceivable that some physical means might also achieve it.

在此之上，他们还证明了图的非同构问题，即图同构问题的补集，具有零知识证明。这个问题存在于共同NP中，但目前还没有出现在NP或任何实用课程中。更普遍的是,Russel Impagliazzo 和Moti以及Ben-Or等人将继续表明,还假设单向函数或牢不可破的加密,还有零知识证明在IP = PSPACE所有问题,换句话说,任何能够证明一个交互式系统可以用零知识证明的证据

On top of this, they also showed that the **graph nonisomorphism problem**, the complement of the graph isomorphism problem, has a zero-knowledge proof. This problem is in **co-NP**, but is not currently known to be in either **NP** or any practical class. More generally, Russell Impagliazzo and Moti Yung as well as Ben-Or et al. would go on to show that, also assuming one-way functions or unbreakable encryption, that there are zero-knowledge proofs for *all* problems in **IP** = **PSPACE**, or in other words, anything that can be proved by an interactive proof system can be proved with zero knowledge.

In September 2017, the first ZKP was conducted on the Byzantium fork of Ethereum.

References

External links

"What is a zero-knowledge proof and why is it useful?". 16 November 2017.

"Ethereum Upgrade Byzantium Is Live, Verifies First ZK-Snark Proof". Cointelegraph. Retrieved 2017-12-18.

A tutorial by Oded Goldreich on zero knowledge proofs

Demonstrate how Zero-Knowledge Proofs work without using maths

The Bitcoin's Zero knowledge proof to binding

https://en.wikipedia.org/wiki/Zero-knowledge_proof

Lecture Outline

- ✓ Definition
- ✓ Abstract examples
- ✓ Practical examples
- ✓ Applications
- ✓ History
- ✓ References

完

धन्यवाद

Hindi Hindi

Спасибо

Russian

شكرا

Arabic

Grazie

Italian

நன்றி

Tamil

多謝

Traditional Chinese

Thank You

English

多谢

Simplified Chinese

ありがとうございました

Japanese

ขอบคุณ

Thai

Gracias

Spanish

Obrigado

Brazilian Portuguese

Danke

German

Merci

French

감사합니다

Korean