**Week 2 Seminar Solutions**

1. How to use a cryptographic hash function to explain the two security properties, hiding and binding.

If we substitute the instantiation of *commit* and *verify* as well as H (nonce || msg) for *com*, then these properties become:

• Hiding: Given H (nonce || msg), it is infeasible to find msg.

• Binding: It is infeasible to find two pairs (msg, nonce) and (msg′, nonce′)

such that msg ≠ msg′ and H (nonce || msg) == (nonce′ || msg′).

The hiding property of commitments is exactly the hiding property that we required for our hash functions. If key was chosen as a random 256-bit value, then the hiding property says that if we hash the concatenation of key and the message, then it’s infeasible to recover the message from the hash output. And it turns out that the binding property is implied by the collision-resistant property of the underlying hash function. If the hash function is collision resistant, then it will be infeasible to find distinct values msg and msg′ such that H (nonce || msg) = H (nonce′ || msg′), since such values would indeed be a collision. (Note that the reverse implications do not hold. That is, it’s possible that you can find collisions, but none of them are of the form H (nonce ‖ msg) == H(nonce′ ‖ msg′). For example, if you can only find a collision in which two distinct nonces generate the same commitment for the same message, then the commitment scheme is still binding, but the underlying hash function is not collision resistant.)

2. Examine the two properties in the digital signature scheme.

Two properties hold:

• Valid signatures must verify: verify

.

• Signatures are existentially unforgeable.

The first property is straightforward—that valid signatures must be verifiable. If I sign a message with *sk*, my secret key, and someone later tries to validate that signature over that same message using my public key, *pk*, the signature must validate correctly. This property is a basic requirement for signatures to be useful at all.

*Unforgeability*. The second requirement is that it’s computationally infeasible to forge signatures. That is, an adversary who knows your public key and sees your signatures on some other messages can’t forge your signature on some message for which he has not seen your signature. This unforgeability property is generally formalized in terms of a game that we play with an adversary. The use of games is quite common in cryptographic security proofs.

3. Explain how to use public keys, *pk*, as identities.

If you see a message with a signature that verifies correctly under a public key, pk, then you can think of this as *pk* stating the message. You can literally think of a public key as being like an actor, or a party in a system, who can make statements by signing those statements. From this viewpoint, the public key is an identity. For someone to speak for the identity *pk*, he must know the corresponding secret key, *sk*.

A consequence of treating public keys as identities is that you can make a new identity whenever you want—you simply create a new fresh key pair, *sk* and *pk*, via the *generateKeys* operation in our digital signature scheme. This pk is the new public identity that you can use, and *sk* is the corresponding secret key that only you know and that lets you speak on behalf of the identity pk. In practice, you may use the hash of pk as your identity, since public keys are large. If you do that, then to verify that a message comes from your identity, one will have to check that (1) *pk* indeed hashes to your identity, and (2) the message verifies under public key *pk*.

Moreover, by default, your public key *pk* will basically look random, and

nobody will be able to uncover your real-world identity by examining *pk*. You can generate a fresh identity that looks random, like a face in the crowd, and is controlled only by you.

4. When was the last Bitcoin halving? How much BTC will be given to the block reward after that halving?

The most recent Bitcoin halving (3rd) occurred on 11 May 2020, when block 630,000 was mined. As a result of that halving, the block reward dropped to 6.25 BTC.

**Pre-halving period**

The first block of Bitcoin blockchain, also known as "Genesis Block" or "Block 0", was mined on 3 January 2009 by the coin's enigmatic creator, known only as Satoshi Nakamoto. The creator of Bitcoin set the initial block reward at 50 BTC. Since Bitcoin had no monetary value in those days, there was no real incentive to participate in mining, and Satoshi was almost the only miner. However, as early as 17 March 2010, BitcoinMarket.com became the first-ever Bitcoin exchange. That caused a surge of interest in the new currency, and, in spring 2011, the price of Bitcoin surpassed $1.

|  |  |
| --- | --- |
| Date | 3 January 2009 |
| Block number | 0 |
| Block reward, BTC | 50 |
| BTC created per day | 7200 |

**Bitcoin halving 2012 (1st)**

The first halving took place on 28 November 2012. At first, the halving had no noticeable effect on Bitcoin's price. However, at the beginning of 2013, the coin's value began to steadily grow, and, in April, it gave way to a correction and continued again in autumn 2013, ending above $1,100. This was followed by a prolonged fall in prices, which went down to $152 on 14 January 2015. Finally, in October 2015, 9 months before the next halving, steady growth began again.

|  |  |
| --- | --- |
| Date | 28 November 2012 |
| Block number | 210,000 |
| Block reward, BTC | 25 |
| BTC created per day | 3600 |
| BTC price at the start | $12 |
| BTC price 100 days later | $42 |
| BTC price 1 year later | $964 |

**Bitcoin halving 2016 (2nd)**

The second halving took place on 9 July 2016. That date was highly anticipated by the crypto community. These expectations, coupled with a sharp rise in Bitcoin's renown and acceptance, led to a noticeable price increase that began at the end of May, a month and a half before the halving. However, a correction took place in mid-June, and, soon after the halving itself, the price fell again with its local minimum reaching May levels.

As it later turned out, that was only a short-term correction. The bullish trend soon continued and developed into exponential growth. This growth peaked on 17 December 2017, when the price reached its all-time high of $19,700. After that, a long bearish trend set in.

|  |  |
| --- | --- |
| Date | 9 July 2016 |
| Block number | 420,000 |
| Block reward, BTC | 12.5 |
| BTC created per day | 1,800 |
| BTC price at the start | $663 |
| BTC price 100 days later | $609 |
| BTC price 1 year later | $2550 |

**Bitcoin halving 2020 (3rd)**

The third halving, which took place on 11 May 2020, as well as the previous Bitcoin halving, did not cause an immediate price increase. It's true that growth began on earlier halving dates at the beginning of the year, but the coronavirus crisis that started in March caused Bitcoin's price to collapse. This makes it even more difficult to assess how much halving is already factored into the price. At the same time, it should be taken into account that the current amount of Bitcoin being mined is quite small compared to the total amount of Bitcoin traded, and it is unlikely that a relatively small drop in supply can cause a significant price increase. Thus, a possible further increase in the price of Bitcoin will be much more influenced by a growth in demand than a reduction in supply.

|  |  |
| --- | --- |
| Date | 11 May 2020 |
| Block number | 630,000 |
| Block reward, BTC | 6.25 |
| BTC created per day | 900 |
| BTC price at the start | $8740 |
| BTC price 100 days later | $11,950 |
| BTC price 1 year later | ? |

The next Bitcoin halving (4th)

The next halving is expected around 2024. It will drop the block reward to 3.125 BTC.

|  |  |
| --- | --- |
| Date | Around 2024 |
| Block number | 840,000 |
| Block reward, BTC | 3.125 |
| BTC created per day | 450 |

5. Is Bitcoin Mining Wasteful?

It’s often said Bitcoin wastes energy, because the energy expended on SHA-256 computations does not serve any other useful purpose. However, any payment system requires energy and electricity. With traditional currency, considerable energy is consumed printing currency and running ATM machines, coin-sorting machines, cash registers, and payment processing services, as well as transporting money and gold bullion in armored cars. You could equally argue that all of this energy is wasted, in that it doesn’t serve any purpose besides maintaining the currency system. So if we value Bitcoin as a useful currency system, then the energy required to support it is not really being wasted.

6. Are Mining Pools beneficial? Explain the pros and cons.

The advantages of mining pools are that they make mining much more predictable for the participants, and they make it easier for smaller miners to get involved in the game. Without mining pools, the variance would make mining infeasible for many small miners.

Another advantage of mining pools is that since there’s one central pool manager who is sitting on the network and assembling blocks, it is easier to upgrade the network. Upgrading the software that the mining pool manager is running effectively updates the software that all pool members are running.

The main disadvantage of mining pools, of course, is that they are a form of centralization. It’s an open question how much power the operators of a large mining pool actually have. In theory, miners are free to leave a pool if it is perceived as too powerful, but it’s unclear how often miners do so in practice. Another disadvantage of mining pools is that they reduce the number of participants actually running a fully validating Bitcoin node. Previously, all miners, no matter how small, had to run their own fully validating node. They all had to store the entire block chain and validate every transaction. Now, most

miners offload that task to their pool managers.