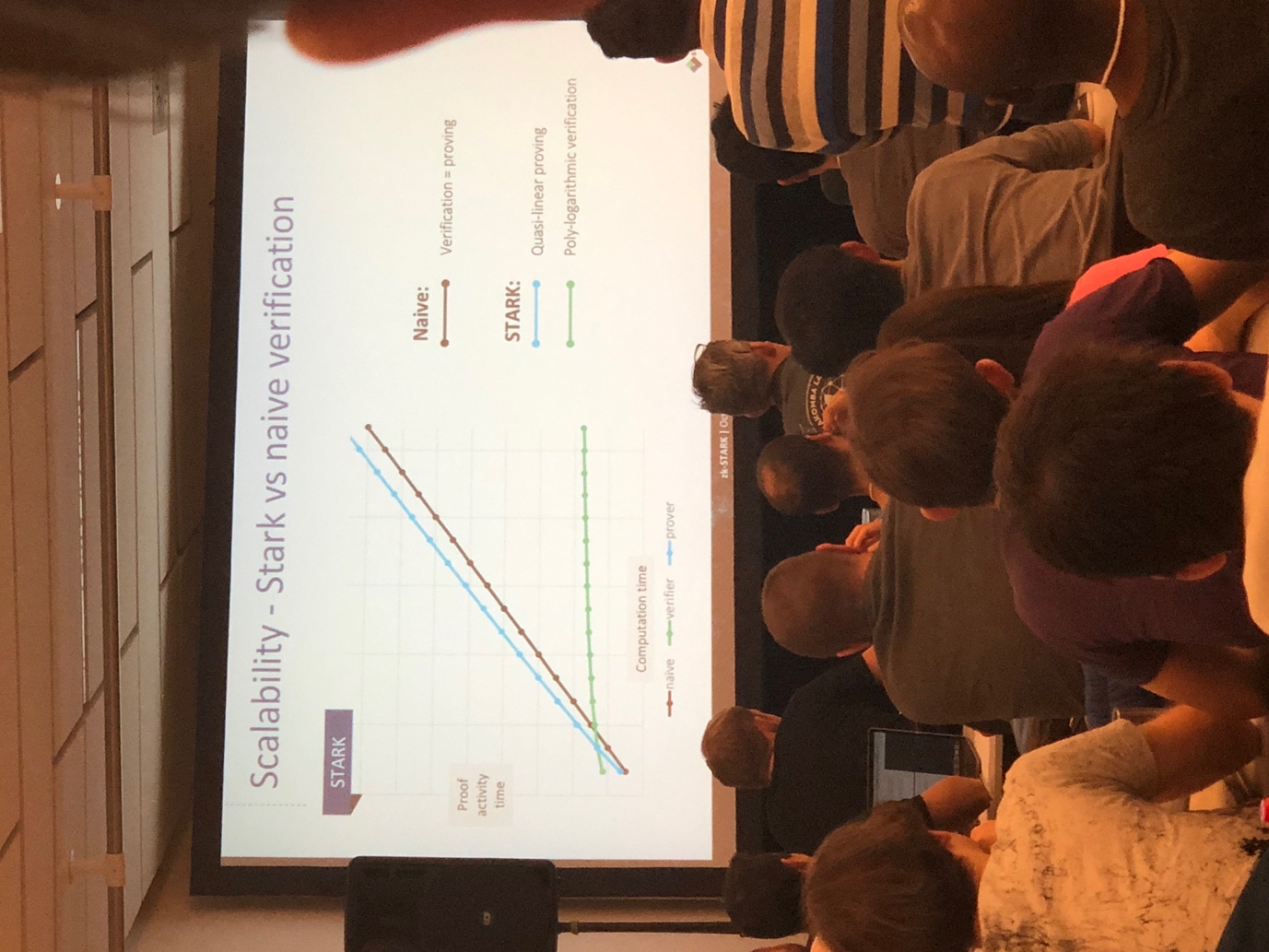
[Starkware Devcon4 Part 1/3]

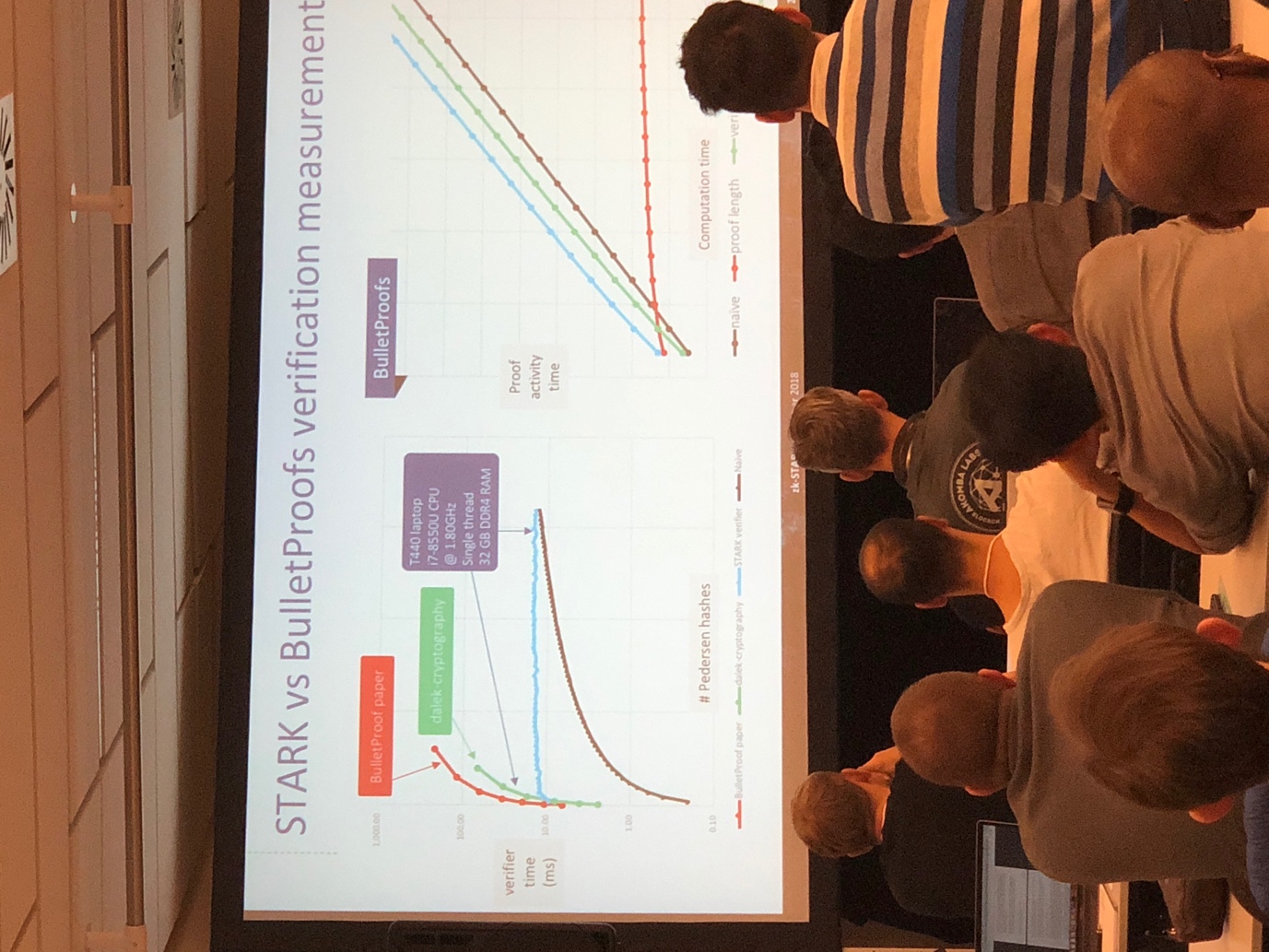
Stark should look like



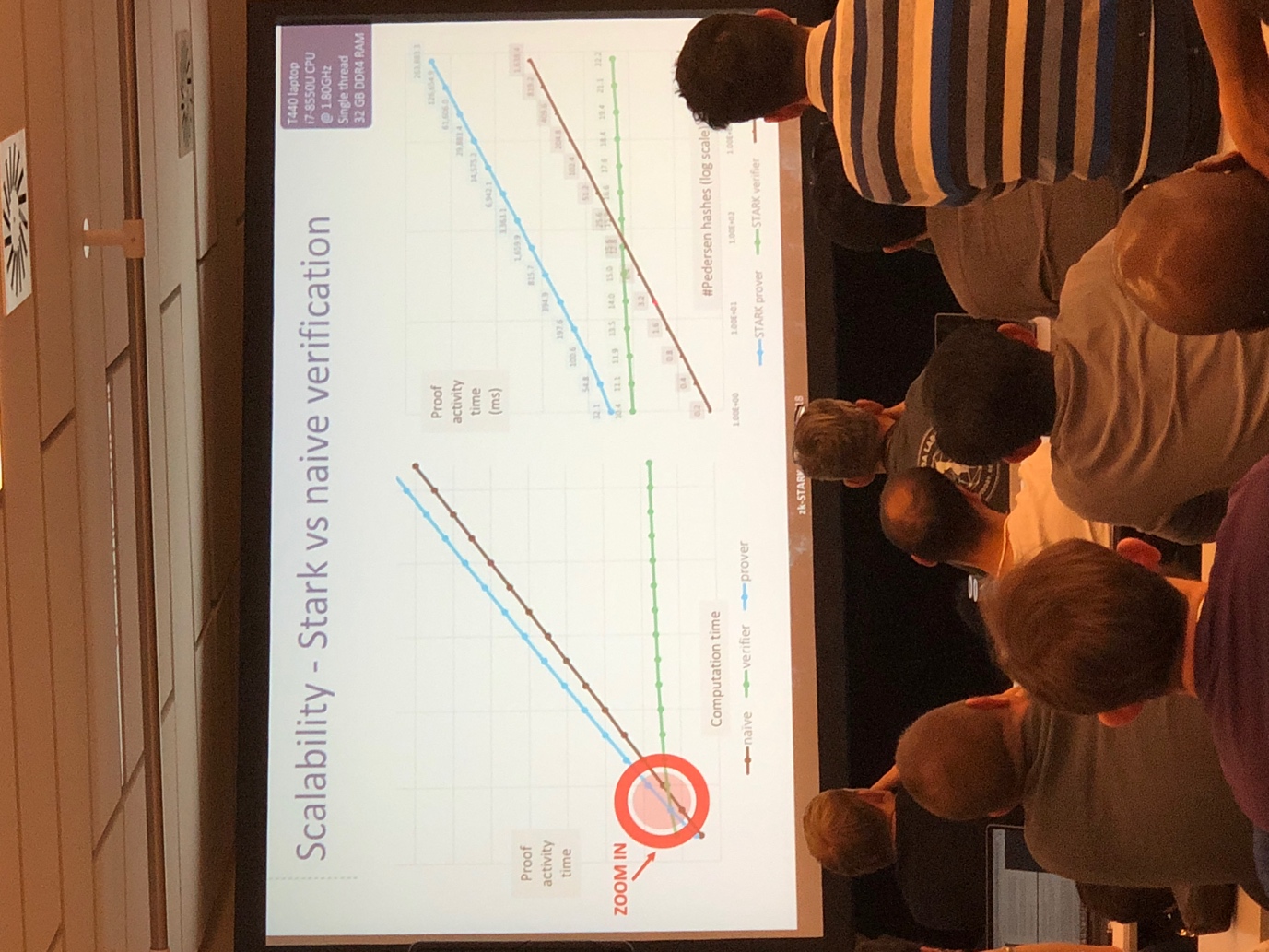
Cant be less than black line, the proving time. Be on it or above.

Coda is a recursive snark with a much shorter trusted setup. Allow large block times is necc. If u use recursive snark.

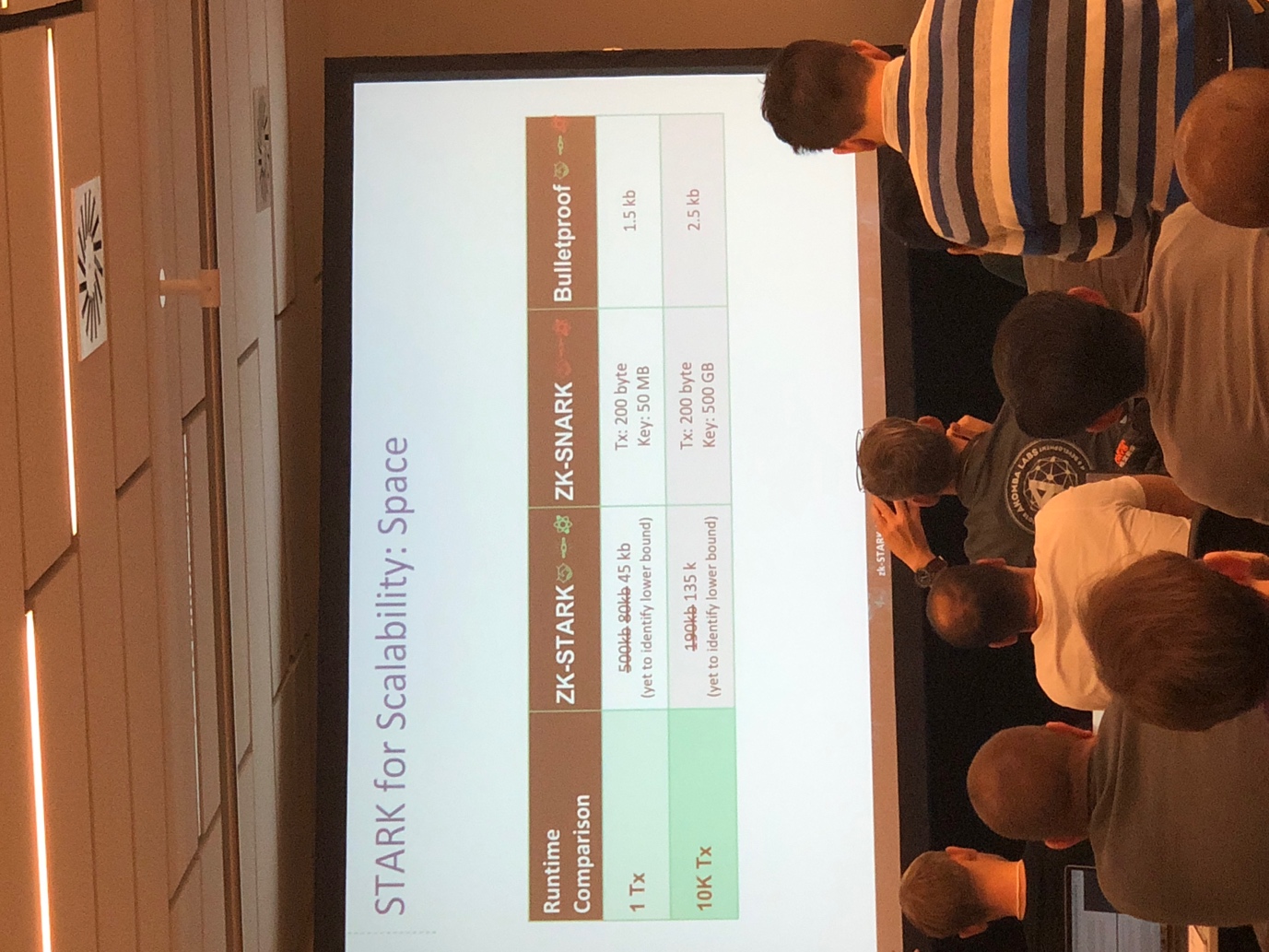




Computation that you saw a lot of hashes using pedersen hash. Red is original. Bulletproof paper. Green is more modern library. Actual measurwments from speaker. Zoom in:



Reliance on simpler cryptographic assumption. Faster verification time of starks. Several decades understanding.



Zk starks are best. Bulletproofs second best. Half a year ago, we went down on time for zk starks.

Diagonal is some constraint. Called AIR. Algebraic intermediate representation.

How to convert computation to AIRs. Long term is yes. There will be tools that will press a button and convert it to airs. In the mid term, we will have domain specific language. In the very short term will hand optimize errors. Very first milestones. Computation integrity statement. Everyone knows merkle root.

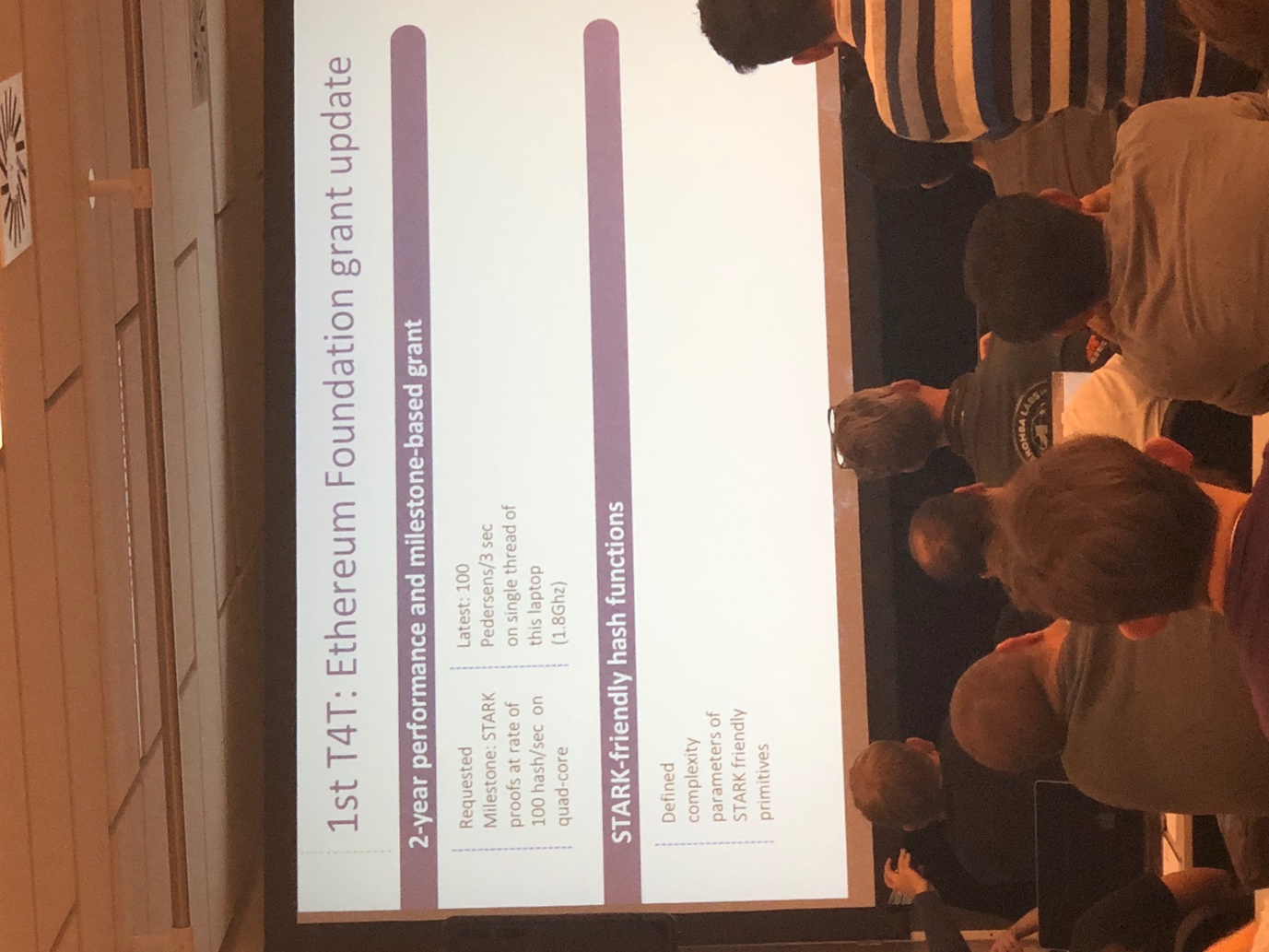
Todo: look up pedersen merkle hash. 6 months of engineering



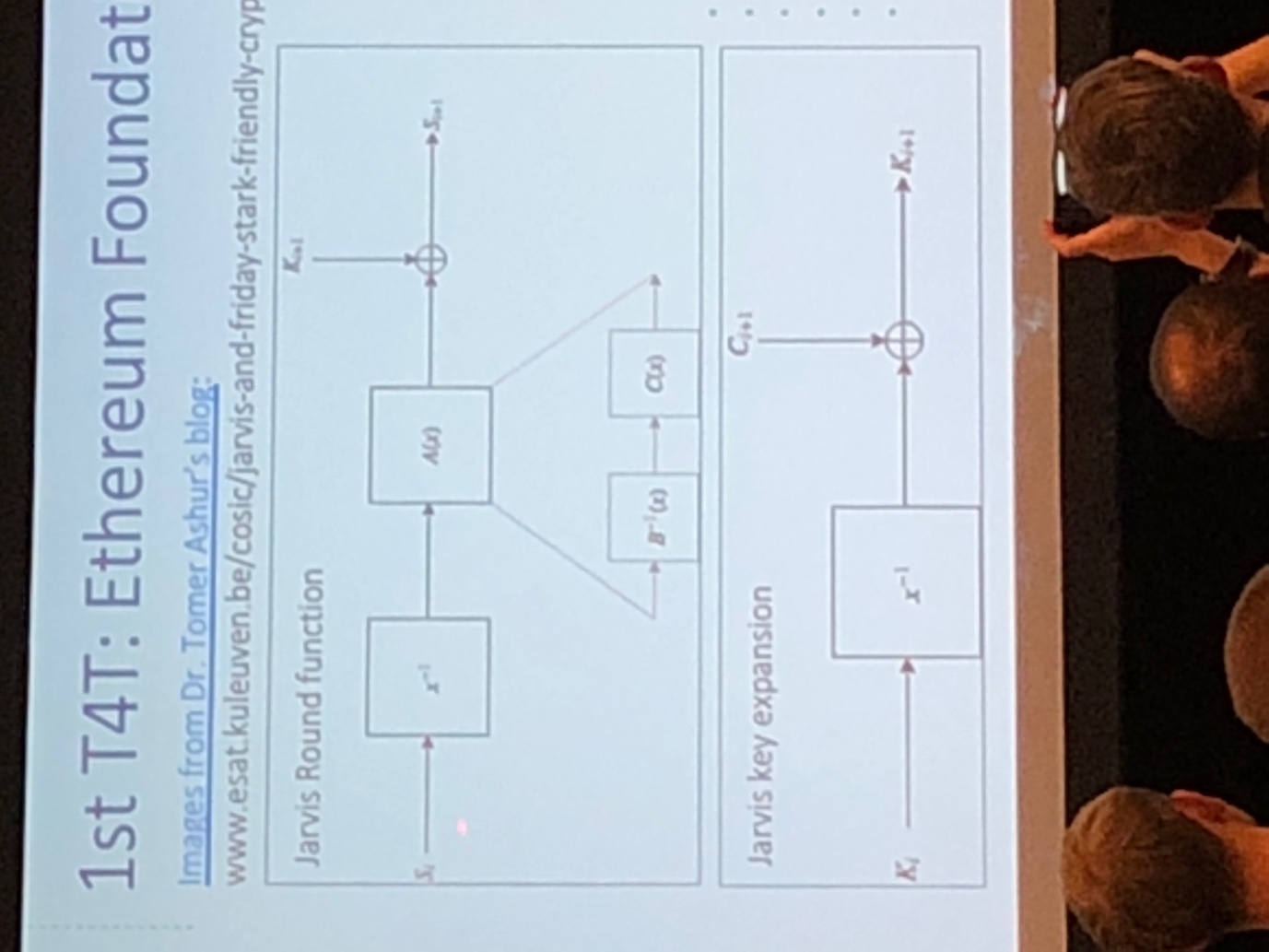
Starts at 10 never goes over 16 or 17. Top in ms bottom is in seconds. 260 seconds to generate proof. Verification factor. Proof size is 300k. Number of hashes here is 10 billion.

What does stark verifier do? 95% of the process is checking hashes. Prover commits to large data sets. Verifier checks small number of linear algebra equationz

Commitments from merkle is correct. If some one shows you a proof, saw slmething correspond to 10k pedersen hashes. 1200 to 1900 hashes that they would need to check. Gas cost, this number times gas cost.

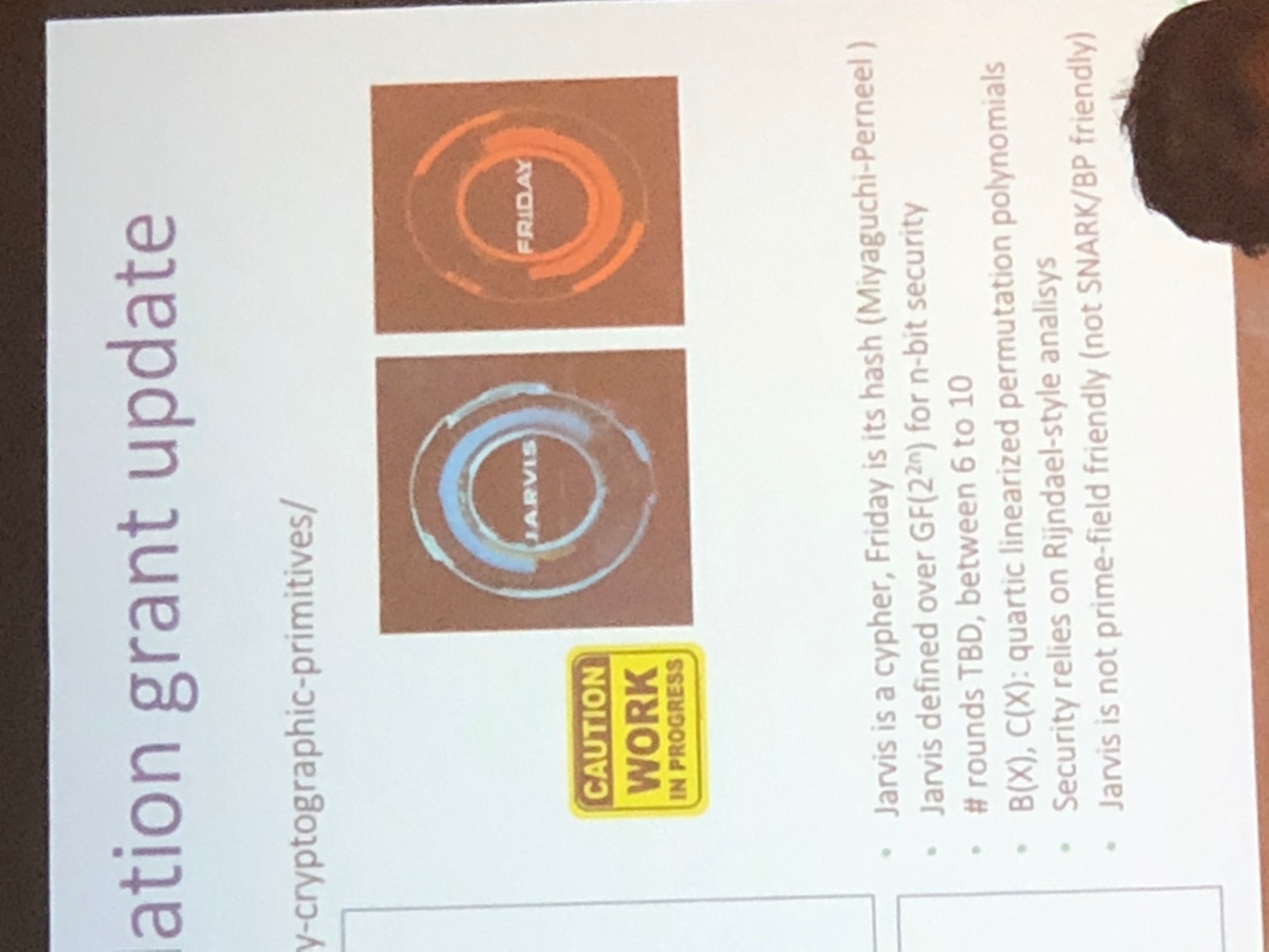


A way to gauge how to construct stark friendly primitives. Announced by dr tom oshul.­­ Vincent iman, professor rijmen co creator of aes encryption algorithm. Cea



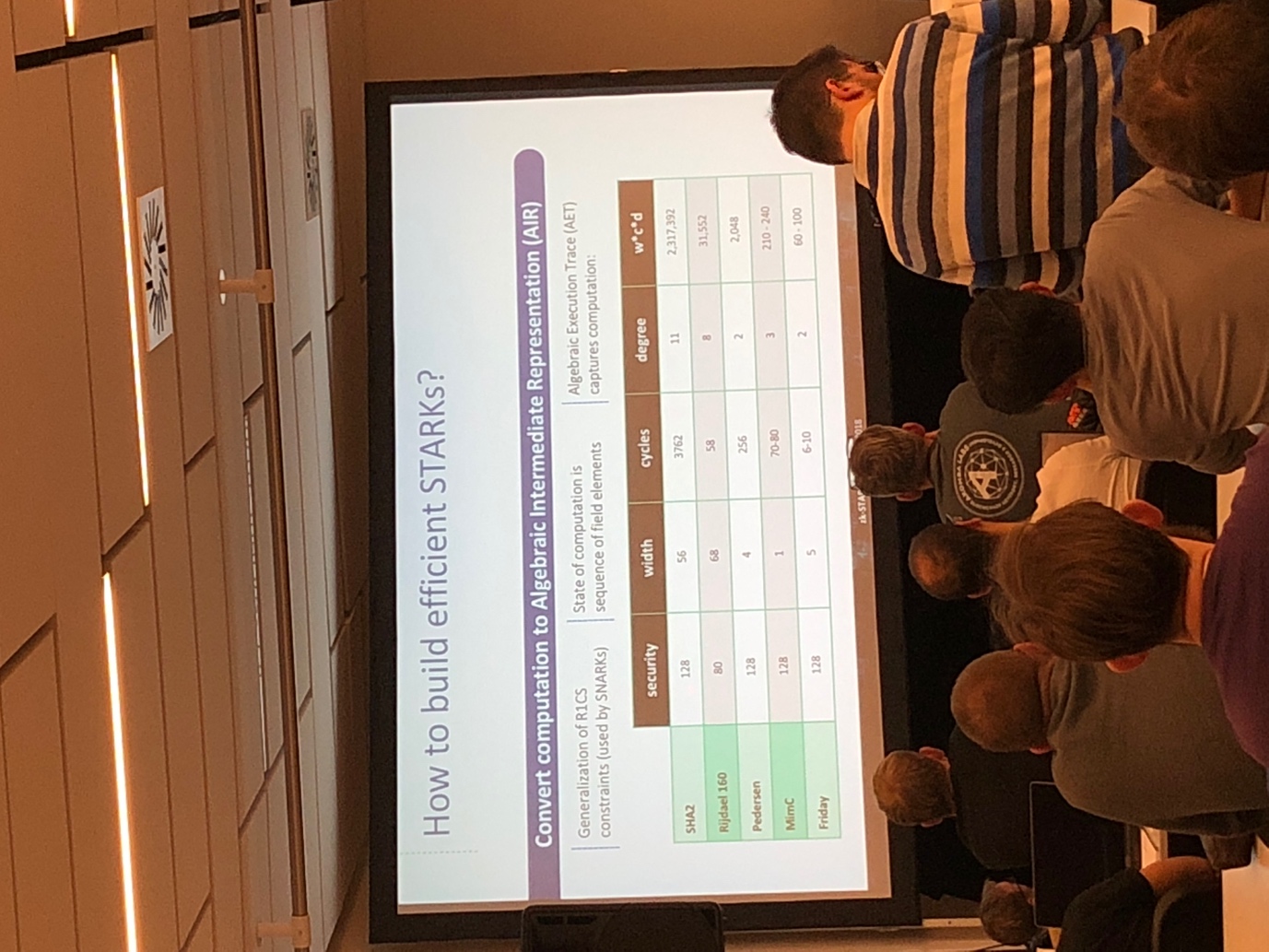
The stake in a stark sustem. Stake will be single field element one with 256 bits. First appply inversion 1 over current state. Then plug it into transformation. Plug to a 2 affine polynomial low degree.

­­­



Main trick. Friday is the standard hash. This how u go from jarvis to friday.

Write an EIP for binary field operations. Like add mult submit EIP requests, Ling to do this asap. Excited about jarvis and Friday.



This pic is next level. Take the product of these 3 things as a complexity measure.



Prove that I know the merkle path from a leaf to a root. 2 leaves in a specific merkle roof. Eqv to prove shielded transaction.

3 args required:

Leaf hash

Tree depth

Security hash

The next step is

Merkle root

Leaf hash

Tree depth

Proof: (tx hash or signature)

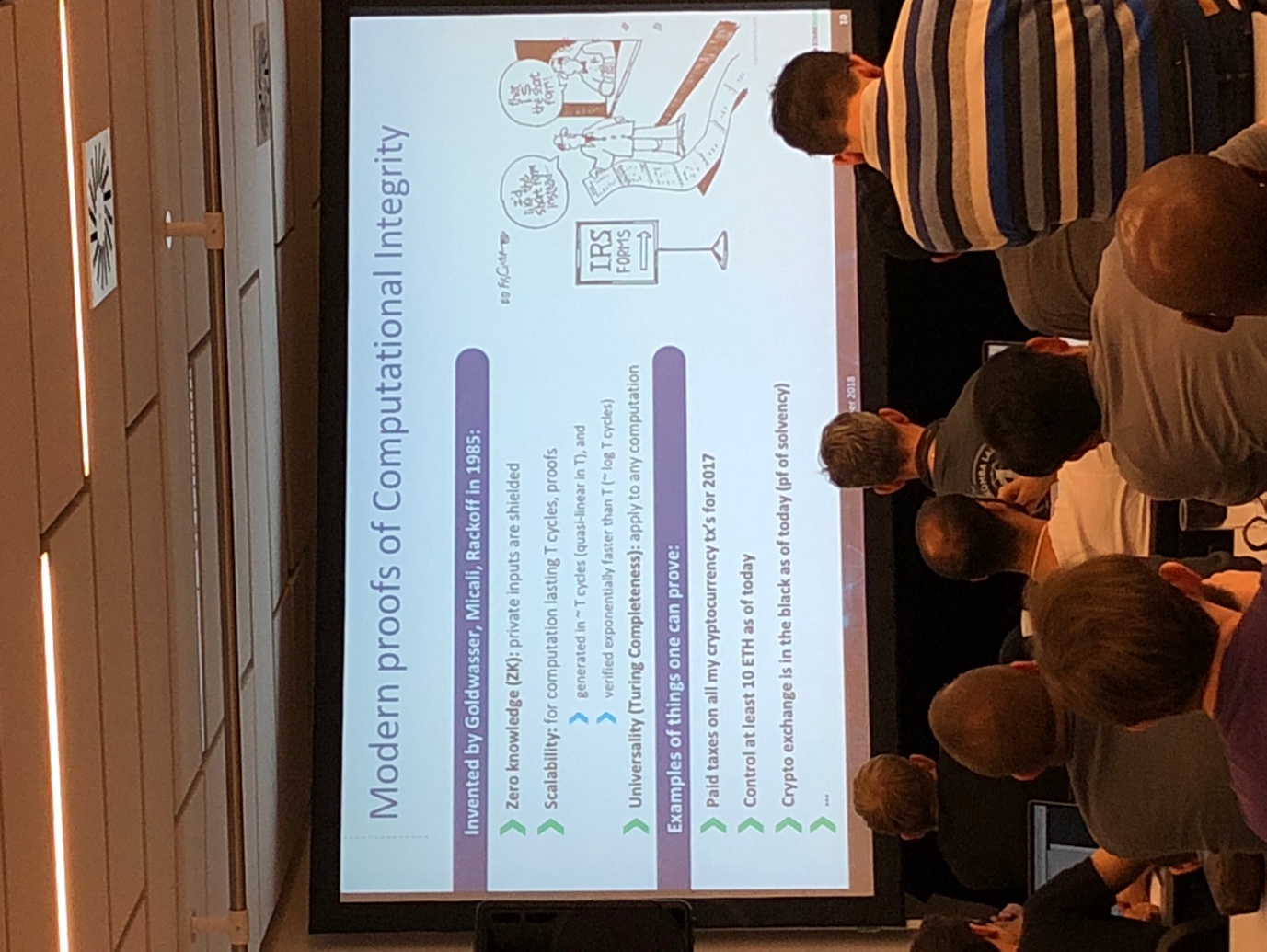
Security bits can be 80 bits or 120 bits. Less than 15 seconds. Our code compiled to web assembly. Thats the secret sauce for mobile optimization.

Merkle verification path is even more complex. As computation w

[Starkware Devcon4 Part 2/3]



Putting in things that are new. What is integrity. Founder of algorand. 62 million raised.



Randomless. Small probability of error. Talk specifically about starks. Many constructions.

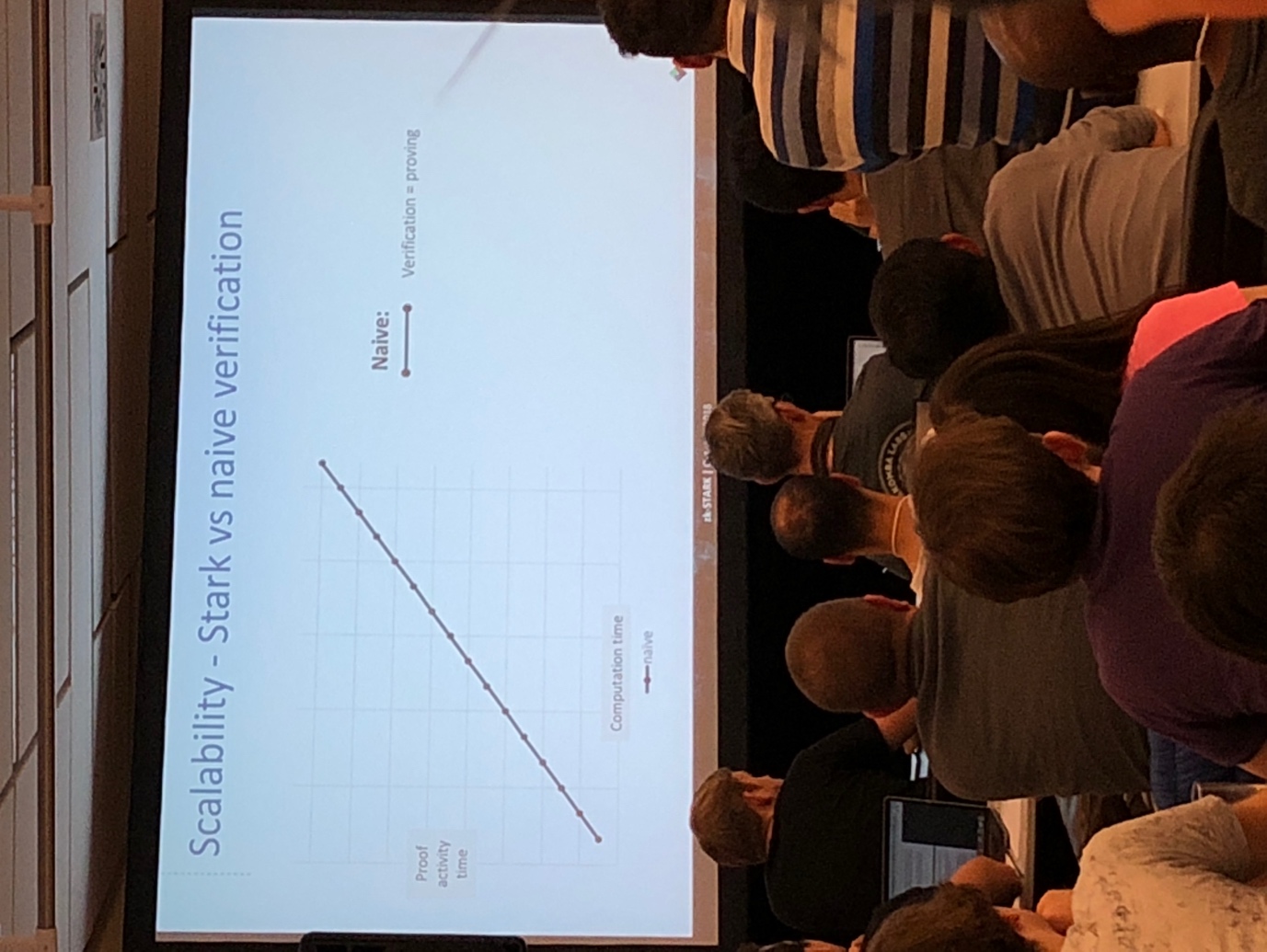
Any system that satisfies these conditions can be considered a zk stark.

Cost of generating a proof is linear to cycles. Time to verify is exponentially faster than t. All messages sent on behalf of the prover must be accounted for and must be trusted random bits.

Knows it was generated correctly. And can fully extract a witness. This is the last attribute.

Fiat shamir heuristic, random coins. Some functions you trust from oracle random function.

SCI-POC is the only library that works. Does not have zero knowledge. Recursive version of snarks in coda.



Use naiive verification, the cerifer runs same inputs as proved. Time to generate is same to verify.