Papers summary

Martinez Anthony

February 2022

Contents

1 Introduction

This report contains all summary for my reading on the different SNARK papers. I'm currently reading oher papers and check what I understood from the previous one, so things will change.

2 Cryptographic definition

Verifiable decryption: a verifiable decryption [CS03] is a primitive which can convince the verifier that the decrypted message is indeed from the corresponding ciphertext.
Rerandomizable encryption: a rerandomizable encryption [PR07] is a public-key encryption scheme where the ciphertext can be rerandomized, which can be viewed as a newly-encrypted ciphertext.
Additively-homomorphic encryption: an additively-homomorphic encryption is a primitive that allows computations on ciphertexts.

3 SAVER

[KO] give the opportunity to detach the encryption of our snark circuit. It's based on ElGamal Encryption.
With this scheme we can convince a verifier that a clear message M is indeed from a corresponding ciphertext C.
So it allow us to prove the correctness of the message without revealing the secret key.

Here are the different primitive of their scheme :
CRS $\leftarrow$ Setup(R) : takes an arbitrary relation R as an input, and outputs the corresponding common reference string CRS.
SK, P K, V K $\leftarrow$ KeyGen(CRS) : takes a CRS as an input, and outputs the corresponding secret key SK, public key P K, verification key V K.
π, CT $\leftarrow$ Enc(CRS, P K, M, $a_{n+1,l}$, $a_{l+1,m}$) : takes CRS, a public key P K, a message M = $m_1, ..., m_n$, a zk-SNARK statement $a_{n+1,l}$, and a witness $a_{l+1,m}$ as inputs, and outputs a proof π and a ciphertext CT = $(c_0, ..., c_n, \phi)$.
π' , $C'_t \leftarrow$ Rerandomize(PK, π, CT) : takes a public key PK, a proof π, a ciphertext CT as inputs, and outputs a new proof π' and a new ciphertext CT' with fresh randomness.
0/1 $\leftarrow$ Verify_Enc(CRS, π, CT , $a_{n+1,l}$) : take a CRS, a proof π, a ciphertext CT and a statement $a_{n+1,l}$ as inputs, and outputs 1 if CT , $a_{n+1,l}$ is in the relation R, or 0 otherwise.
M, v $\leftarrow$ Dec(CRS, SK, VK, CT) : takes CRS, a secret key SK, a verification key VK, and a ciphertext CT = $(c_0, ..., c_n, \phi)$ as inputs, and outputs a plaintext M = $m_1, ..., m_n$ and a decryption proof v.
0/1 $\leftarrow$ Verify_Dec(CRS, VK, M, v, CT) : takes CRS, a verification key VK, a

message M , a decryption proof v, and a ciphertext CT as inputs, and outputs 1 if M is a valid decryption of CT , or 0 otherwise.

4 COCO

[Kos+] is designed to prove that someone else knows a knowledge. The knowledge is already knows by you. And it's supposed to provide a high level of snark circuit implementation, to be more user friendly.
I didn't find any repository for now..
By the way to achieve this they cipher the witness inside the circuit in order to put a backdoor. With that if someone knows the witness it can prove it. And this encryption is interesting in our case.
For example with an RSA encryption the essential challenge is that the arithmetic operations are over integers mod n, where n is larger than the SNARK field order p. They represent integers mod n as $\lceil \frac{\log_2 n}{m} \rceil$ m-bit elements. To multiply a pair of such integers z=xy mod n, they construct a circuit that verifies xy = qn + z, where q and z are $\lceil \frac{\log_2 n}{m} \rceil$ m-bit elements provided as witnesses by the prover. Their current implementation for big integers uses m = 64.
But as they said it's not efficient enough so they proposed a Diffie-Hellman key exchange via a SNARK-friendly field extension. Instead of relying on RSA as the main PKE scheme, they investigate another scheme based on the Discrete-Logarithm problem in Extension Fields, and use it for symmetric key exchange. Since p is only 254-bit prime, the DL problem in F_p will not be hard, therefore an extension F_{p^μ} will be used instead. The key exchange circuit has two generators in that case g, h $\leftarrow F_{p^\mu}$, where $\langle g \rangle = \langle h \rangle$ is a large multiplicative subgroup of order $q|p^\mu - 1$. They follow Lentra's guidelines for selecting q to be a factor of the μ-th cyclotomic polynomial $\theta_\mu(x)$ when evaluated at x = p [Len97] (I'm currently reading it, update will come).

5 Lenstra

[Len97] from what I understand, they said that we can optimize calculations through our finite field by choosing a right basis. And they give a method to select this basis.

Articles

[Len97] A. K. Lenstra. "Using cyclotomic polynomials to construct efficient discrete logarithm cryptosystems over finite fields". In: *Information Security and Privacy* (1997), pp. 126–138.

[CS03] Jan Camenisch and Victor Shoup. "Practical verifiable encryption and decryption of discrete logarithms". In: *Annual International Cryptology Conference* (2003), pp. 126–144.

[PR07] Manoj Prabhakaran and Mike Rosulek. "Rerandomizable rcca encryption". In: *Annual International Cryptology Conference* (2007), pp. 517–534.

[KO] Jiwon Lee Jaekyoung Choi Jihye Kim and Hyunok Oh. "SAVER : SNARK-friendly, Additively-homomorphic, and Verifiable Encryption and decryption with Rerandomization". In: (). DOI: `https://docs.zkproof.org/pages/standards/accepted-workshop3/proposal-saver.pdf`.

[Kos+] JAhmed Kosba et al. "COCO: A Framework for Building Composable Zero-Knowledge Proofs". In: (). DOI: `https://eprint.iacr.org/2015/1093.pdf`.