Suterusu yellowpaper (V 0.1)

Dr. Lin

Co-founder and CTO of Suterusu project

Abstract. This work introduces the main technical modules of Suterusu project. We provide a detailed description of a confidential payment scheme compatible with the existing account-based smart contract platform such as Ethereum. Our proposed scheme does not require a trusted setup and both its communication and computational overhead are constant. In addition, we present a hybrid PoW/PoS mechanism to serve as Suterusu's consensus protocol.

1 Introduction

This yellowpaper will introduce technical details of the Suterusu project, including the cryptographic modules and consensus protocol we intend to use in our testnet. Our testnet will adopt a similar framework to the substrate library as the bedrock of our custom blockchain. It will implement and integrate our proposed confidential payment scheme for the smart contract platform. The confidential payment scheme will be based on groups of unknown order, more specifically class groups of imaginary quadratic order. Our proposed scheme does not require a trusted setup step and its communication and computational overhead is constant.

1.1 Roadmap

We will describe the testnet framework first and then our own design of confidential payment scheme for the smart contract platform. The hybrid proof of work/proof of stake (PoW/PoS) mining mechanism will be introduced in the following section. We will conclude with future work at the end of this paper.

2 Testnet framework

Our testnet is a fork of substrate framework, which is a library created by Parity Technologies to facilitate the fast and easy development of a custom blockchain. There are mainly two benefits to base our testnet on a fork of substrate: first, we can develop the smart contract module, proof-of-stake and more sophisticated liquid decentralized meritocracy protocol by modifying technical modules provided by the substrate runtime module library; Secondly, through compiling the code to WebAssembly and deploying it as a message on the network, forkless client updates can be ensured due to the WebAssembly fallback. Other benefits include lightweight client, interoperability etc.

3 Confidential payment for the smart contract platform

Our basic confidential payment scheme is conceptually close to the Zether [1] confidential transfer scheme, which can be viewed as an adaption of confidential payment in the UTXO model with the underlying commitment scheme in the UTXO model replaced by the Elgamal encryption, and the respective zero-knowledge proof scheme modified accordingly.

To transfer an amount b^* from a public key y to a public key $\bar{y}$, a user first generates the encryption of the balance associated with the public key y, i.e., $(C_L, C_R)=(g^b y^r, g^r)$, where b is the balance. In addition, the user also generates $(C, D)=(g^{b^*} y^r, g^r)$ and $(\bar{C}, \bar{D})=(g^{b'} y^r, g^r)$, where b^* is the deducted amount from the user's balance while b' is the remaining balance and r is the randomness used in the encryption. We use zero-knowledge proof here to prove the correctness of confidential transfer, i.e., the balance encryption is well-formed and all the aforementioned amounts are consistent and belong to the correct range. In the following sections, we will separately introduce the Σ-protocol for these two statements and then briefly introduce how they can be combined together to form the final non-interactive zero-knowledge proof for our confidential payment scheme.

Our cryptographic modules are built upon class groups of imaginary quadratic order and the interested readers are referred to [5, 6] for details on class groups. The public parameters of the following protocols include the description of the class group $\mathcal{G}$ and the generators g, h and a collision-resistant hash function $H : \{0,1\}^* \to \{0,1\}^{2\lambda}$, where λ is the security parameter.

3.1 Basic setup-free Σ-protocol for consistent balance encryption

The statement for consistent balance encryption for the case of class group is:

$$\left\{ \begin{pmatrix} C, \bar{C}, C_{L,n}, C_{R,n}, R, y, \bar{y}, \\ (sk, b^*, b', r) \end{pmatrix} : \begin{array}{l} C = g^{b^*} \cdot y^r \wedge \bar{C} = g^{b'} \cdot \bar{y}^r \wedge C_{L,n} = g^{b'} \cdot C_{R,n}^{sk} \\ \wedge R = g^r \wedge y = g^{sk} \end{array} \right\}$$

Here $C_{L,n} = C_L/C$ and $C_{R,n} = C_R/C$. This statement ensures all these ciphertexts are well-formed in the sense that the sender knows the secret key sk and randomness r used in the encryption and the encrypted balance amounts before and after deduction are consistent. The concrete Σ-protocol for consistent balance encryption can be found in Fig. 1.

Theorem 1. *The proposed Σ-protocol in Fig. 1 is honest-verifier zero-knowledge and computationally sound in the generic group model for groups of unknown order.*

The proof of this theorem can be found in the academic version of this yellowpaper [2].

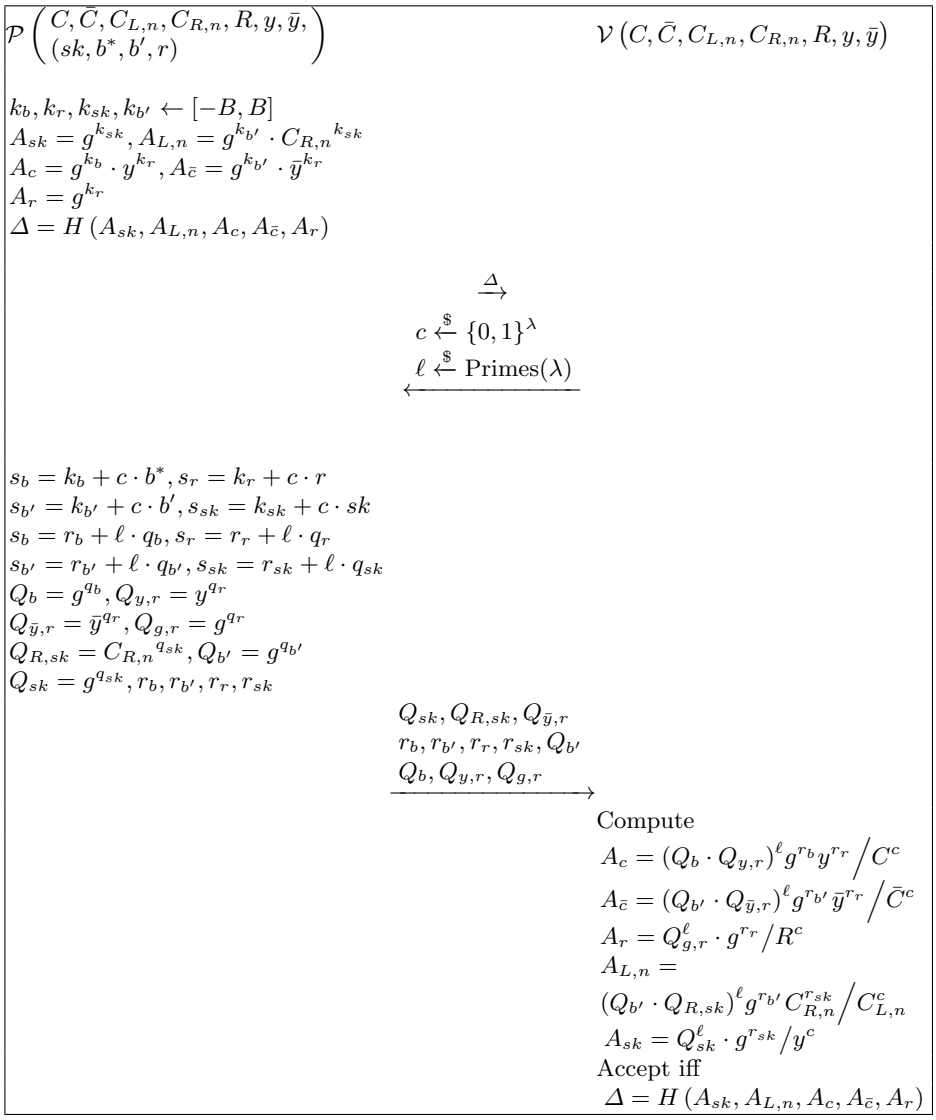

Fig. 1. Basic Σ-protocol for consistent balance encryption

3.2 Setup-free Σ-protocol with a short transcript for consistent balance encryption

The basic scheme proposed in the previous section has a relatively fast verification due to the fact that all the modular exponentiation operations only involve exponents of length smaller than the security parameter. This is achieved mainly through the usage of the prime challenge ℓ. The tradeoff for fast verification is large communication overhead as demonstrated in Fig. 1. By removing the prime challenge, this section introduces a Σ-protocol with a short transcript as shown in Fig. 2.

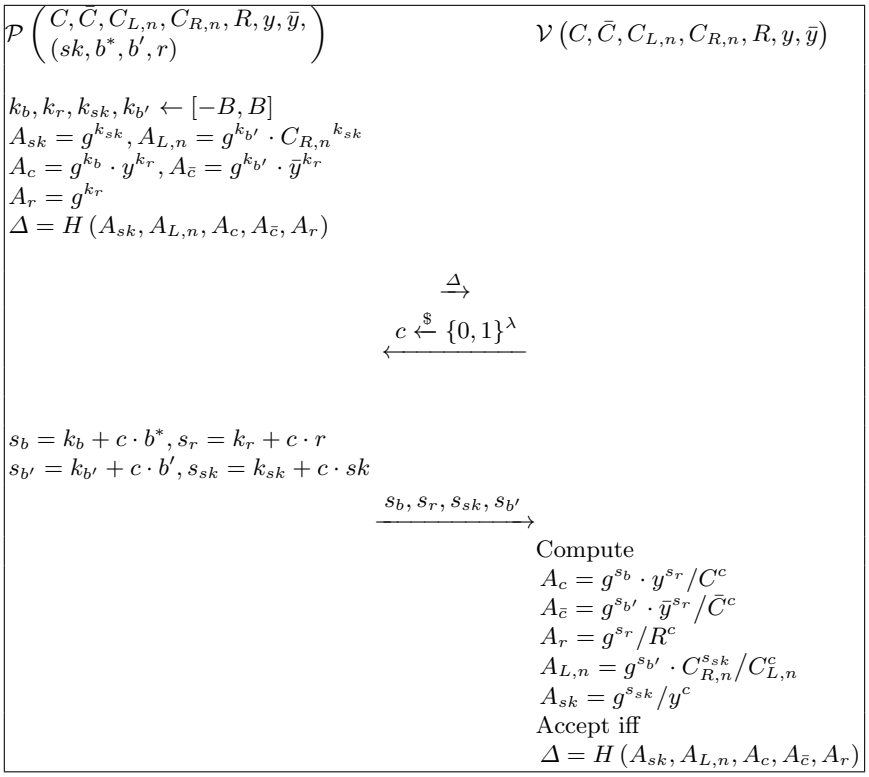

Fig. 2. Σ-protocol with a short transcript for consistent balance encryption

Theorem 2. *The proposed Σ-protocol in Fig. 2 is honest-verifier zero-knowledge and computationally sound in the generic group model for groups of unknown order.*

The proof of this theorem can be found in the academic version of this yellowpaper [2].

3.3 Basic Σ-protocol for setup-free range proof with free base

Zero-knowledge Range proof is used for ensuring there is no overflow attack in confidential payment. This section provides a zero-knowledge range proof for a secret committed in a Pedersen commitment over Class groups. Our proposed scheme can be viewed as a variant of the scheme introduced in [3] but over class groups and with free base chosen by the prover. The original scheme requires a trusted setup while our proposed construction removes this requirement.

Similar to that of [3, 4], to prove $x \in [a, b]$ the prover needs to prove $4(x - a)(b - x) + 1$ is a sum of three squares. The prover will run a variant of *Rabin-Shallit* algorithm [4] as a subroutine to compute the integer representations of $4(x - a)(b - x) + 1$, i.e., $\{x_i\}_{i=1}^{3}$ satisfying $4(x - a)(b - x) + 1 = \sum_{i=1}^{3} x_i^2$. The protocol presented in Fig. 3 is similar to the one proposed in [3] except the bases of the commitment are chosen freely by the prover. The public parameters of the following protocol include the description of the class group $\mathcal{G}$, two random generators of the group g and h and a collision-resistant hash function $H : \{0,1\}^* \to \{0,1\}^{2\kappa}$. In the following description, both u and v are two public random bases chosen by the prover $\mathcal{P}$. B is an integer larger than $2^{2\lambda}|\mathcal{G}|$.

The soundness of the original zero-knowledge argument of range proof relies on the five facts as specified in Proposition 1 of [3]. All these five facts can be derived from the generic group model for groups of unknown order. Therefore, one can prove the following theorems on the security of the above protocol:

Theorem 3. *The proposed Σ-protocol in Fig. 3 is honest-verifier zero-knowledge assuming Decisional DH assumption holds in the underlying group.*

Theorem 4. *The proposed Σ-protocol in Fig. 3 is sound in the generic group model for groups of unknown order.*

The proof of these theorems can be found in the academic version of this paper [2].

3.4 Σ-protocol for short range proof with free base

One could also further shrink the range proof size by removing the prime challenge. The improved Σ-protocol can be found in Fig. 4.

one can prove the following theorems on the security of this protocol:

Theorem 5. *The proposed Σ-protocol in Fig. 4 is honest-verifier zero-knowledge assuming Decisional DH assumption holds in the underlying group.*

Theorem 6. *The proposed Σ-protocol in Fig. 4 is sound assuming both Discrete Logarithm problem and RSA problem are hard in the underlying group.*

The proof of these theorems can be found in the academic version of this paper [2].

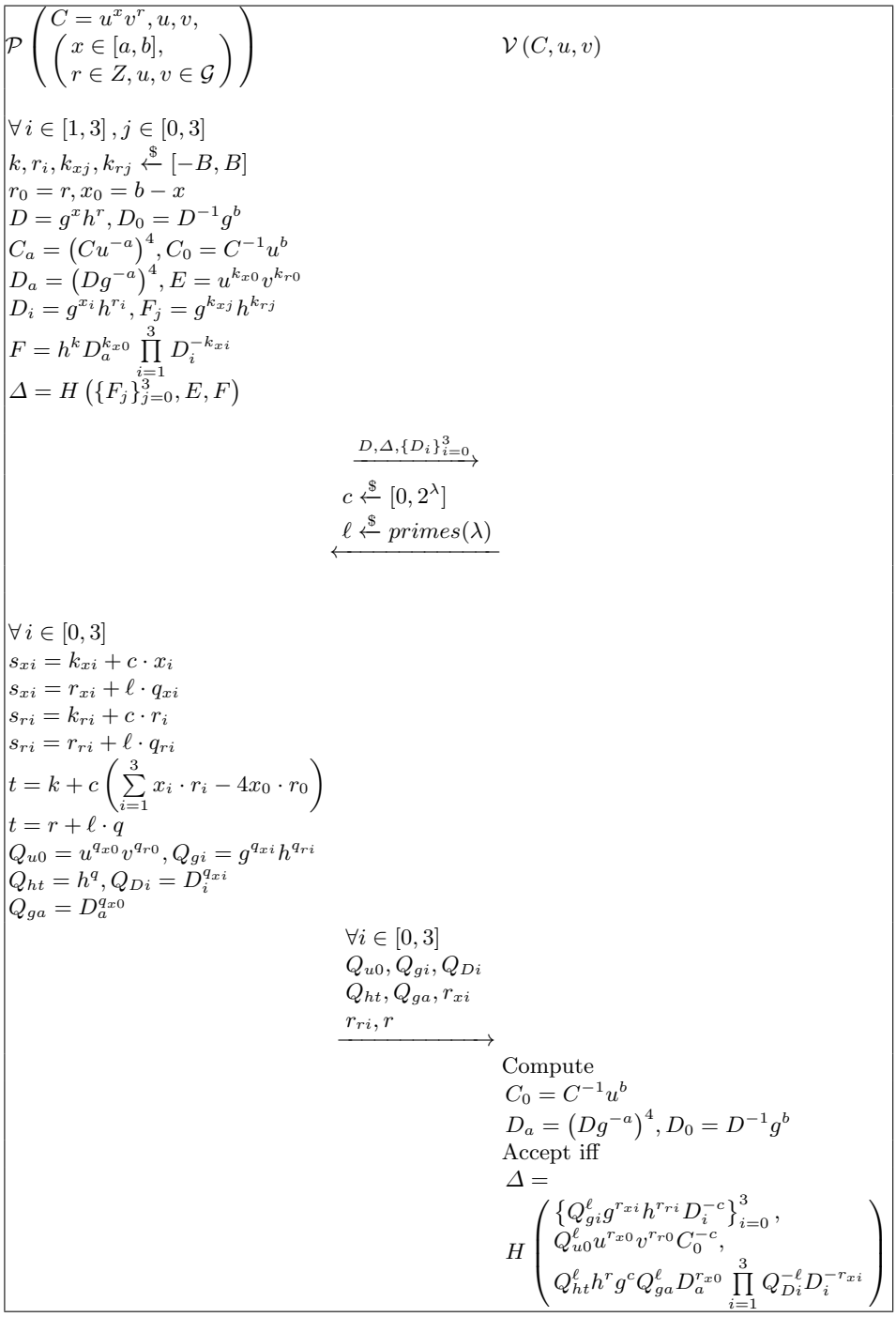

Fig. 3. Basic Σ-protocol for range proof with free base

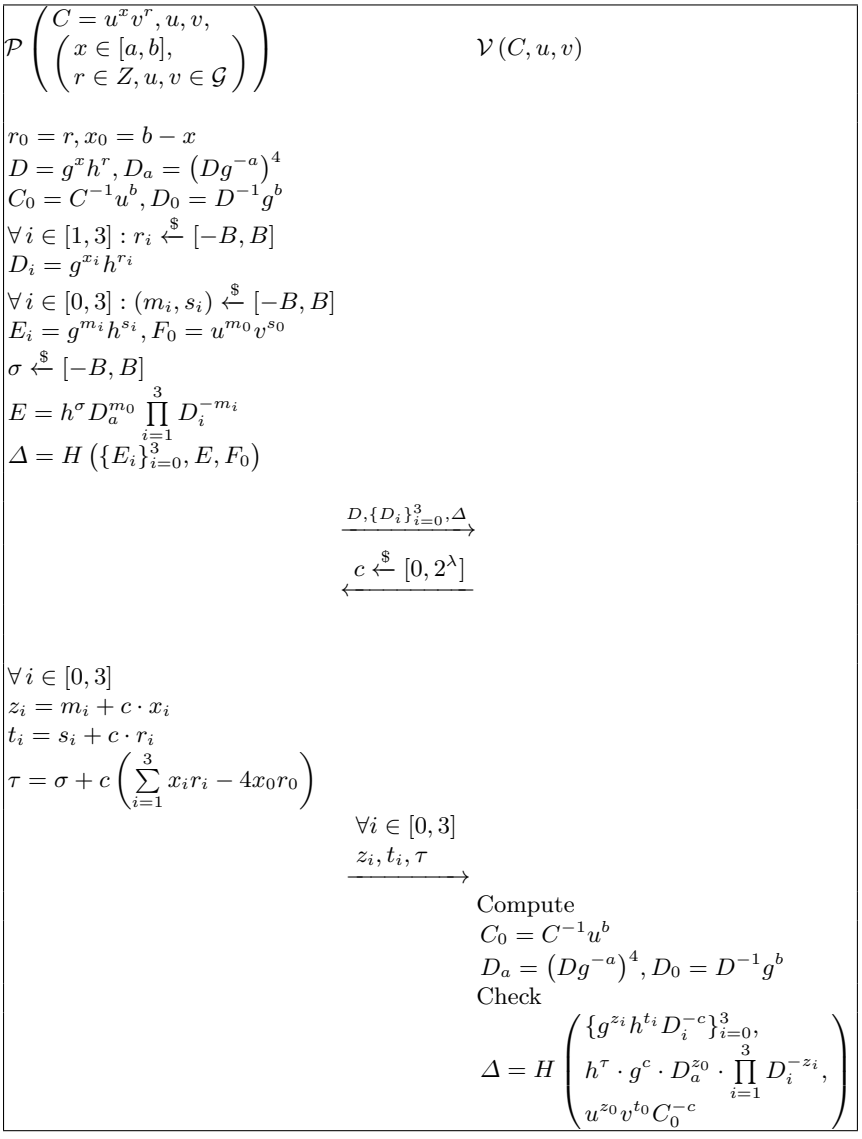

Fig. 4. Σ-protocol for short range proof

3.5 ZK-ConSNARK for confidential transfer in smart contract

The two modules presented in the previous sections can be combined into a single non-interactive zero-knowledge argument scheme with constant communication overhead, i.e., ZK-ConSNARK mentioned in our whitepaper. The respective statement is:

$$\left\{ \left(\begin{array}{c} C, \bar{C}, C_{L,n}, \\ C_{R,n}, R, y, \bar{y}, \\ (sk, b^*, b', r) \end{array} \right) \middle| \begin{array}{l} C = g^{b^*} \cdot y^r \wedge \bar{C} = g^{b^*} \cdot \bar{y}^r \wedge C_{L,n} = g^{b'} \cdot C_{R,n}^{sk} \\ \wedge R = g^r \wedge y = g^{sk} \wedge \\ b^* \in [0, MAX] \wedge b' \in [0, MAX] \end{array} \right\}$$

This statement is essentially a conjunction of the statements proposed in the previous two sections. By applying the Fiat-Shamir transformation, one could easily transform the proposed schemes into their non-interactive versions. The concrete scheme and proof can be found in the academic version of this work.

3.6 Discussion

There are mainly two kinds of potential attacks against a confidential payment scheme for the smart contract platforms as noted in the Zether paper [1]: front-running and replay attacks. The proposed protection mechanism in Zether other than the confidential transfer scheme, such as proof of burn, locking accounts to other smart contracts, etc. can be easily transferred to our setting albeit the security of the modified scheme will be reduced to assumptions over groups of unknown order. We will provide a more detailed description of the revised Zether framework over class group in the future version of this yellowpaper.

4 Consensus protocol: hybrid PoW/PoS

Our testnet will employ a hybrid PoW/PoS protocol as the consensus protocol. The validator nodes are required to invest a certain amount of hardware equipment and pay for the maintenance cost to maintain its infrastructure to ensure the security of the Suter network. There are mainly two responsibilities of validators: 1. Transaction validation and mining; 2. Participating in the Suter network on-chain governance.

More specifically, the validator nodes need to invest in hardware in order to join the network and listen for new blocks. When a new block is proposed, it will validate the block. The block validation of a Suter payment transaction mainly consists of verifying zero-knowledge proof to make sure there is no double-spending attack. The validator nodes are also responsible for assembling new blocks and solving our proof-of-work puzzle to collect the mining reward. Our preliminary choice of PoW hash function will be memory-hard, similar to Ethash used by the Ethereum network. Ethash is ASIC-resistant, which would guarantee the decentralization degree of the Suter network.

In the future, the Suter network will adopt a hybrid PoW/PoS system. A random beacon will be applied to select a voter from the stakeholders. The voter

will be responsible for generating a signature to approve the blocks mined by the PoW miners. This provides additional checks and balances mechanism on the PoW miners.

The miners are also required to participate in voting whenever there is a necessity for protocol change. For instance, there will be a transition period from our current PoS governance to the hybrid PoW/PoS mechanism. The initial block rewards will goes to the PoW miners and stakeholders according to the amount of contributed mining power. The Suter community will decide how to adjust this proportion by running our on-chain governing mechanism in the later development stage.

5 Future work

This yellowpaper describes the technical modules used in the Suterusu project, including the confidential payment scheme and hybrid PoW/PoS consensus protocol. Note that the proposed cryptographic modules is tentative and might be subject to change during our future development. We will implement our proposed schemes and compare them to the schemes with logarithmic complexity to choose the one with more practical performance parameters as the underlying cryptographic modules for our final implementation.

References

1. B. Bünz, S. Agrawal, M. Zamani, and D. Boneh. Zether: Towards privacy in a smart contract world.
2. G. Couteau and H. Lin. Confidential balance transfer for smart contract platform. Cryptology ePrint Archive, Report 2019, 2019.
3. G. Couteau, T. Peters, and D. Pointcheval. Removing the strong rsa assumption from arguments over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 321–350. Springer, 2017.
4. J. Groth. Non-interactive zero-knowledge arguments for voting. In *International Conference on Applied Cryptography and Network Security*, pages 467–482. Springer, 2005.
5. Russell W.F. L. and Giulio M. Subvector commitments with application to succinct arguments, 2019. https://eprint.iacr.org/2018/705.
6. H. Lipmaa. Secure accumulators from euclidean rings without trusted setup. In *International Conference on Applied Cryptography and Network Security*, pages 224–240. Springer, 2012.