



Coloured Ring Confidential Transactions

Felix Engelmann^(✉), Frank Kargl, and Christoph Bösch

Institute for Distributed Systems, Ulm University, Ulm, Germany
{felix.engelmann, frank.kargl, christoph.boesch}@uni-ulm.de

Abstract. Privacy in block-chains is considered second to functionality, but a vital requirement for many new applications, e.g., in the industrial environment. We propose a novel transaction type, which enables privacy preserving trading of independent assets on a common block-chain. This is achieved by extending the ring confidential transaction with an additional commitment to a colour and a publicly verifiable proof of conservation. With our coloured confidential ring signatures, new token types can be introduced and transferred by any participant using the same sized anonymity set as single-token privacy aware block-chains. Thereby, our system facilitates tracking assets on an immutable ledger without compromising the confidentiality of transactions.

Keywords: Coloured coins · Privacy · Confidential ring signature
Commitments

1 Introduction

Trading is a basic human trait that extends to the digital world. Individual trading without the need of intermediaries is enabled by block-chain technology. Participants of a block-chain reach a global consensus on which trades are valid and in which order. To achieve this, all transactions must be validated by peers and checked for violations of conservation rules, e.g., creating an asset out of thin air. The basic approach is to use plain-text transaction receipts, visible for everyone which makes validation of the transactions straightforward.

The issue with these plain-text receipts is, that trades often include valuable information for other parties, using the knowledge for their leverage. Independent research [6, 9] realised, that privacy in block-chain systems is important to support the same features as analogue trades. Monero therefore introduced ring confidential transactions to hide the sender identity (using ring signatures), the recipient identity (using one-time payment addresses), and the amount transferred (using commitments) from the public, while maintaining the possibility to verify the conservation. The real sender is indistinguishably concealed within a set of decoys. To prove ownership of an asset, which is attached to a public key,

This work was partially funded by the Federal Ministry of Economic Affairs and Energy on the basis of a decision by the German Bundestag. The authors thank Henning Kopp for the technical discussions during the sketching phase.

a ring signature is used instead of a regular digital signature. The verifier of a ring signature can check that the signer knows at least one of the corresponding private keys, but not which one. To prevent double spending of the same asset, the ring signature has to include a specific tag, which stores the identity of the signer in an encrypted form. If two signatures have the same tag, they can be linked and the second signature is invalid. Thus, no two assets can belong to the same public key. This requirement demands for one-time recipient addresses, which, in addition, serve the purpose of hiding the recipient from the public. Therefore, a one-time key is derived from the long-term recipient public key, for which only the recipient can derive the correct one-time private key. These one-time addresses prevent multiple transaction outputs to be linked to a common owner. This works well, if all transaction inputs are of equal value. However, transaction inputs with different values, still allow for deducing the real sender by comparing the transaction in- and outputs. In order to prevent this kind of sender derivation and to add privacy, the transaction value is hidden inside a commitment. Additively homomorphic Pedersen commitments [7] allow the sender to prove that the input minus the output of a transaction is zero, thereby proving the conservation without disclosing the amount. A detailed description of the techniques used is summarised by Alonso et al. [1].

The added privacy compared to fully visible transaction receipts restricts features, such as Turing complete smart contracts, which are common on non-privacy aware block-chains. Smart contracts are recipients, whose behaviour is governed by code. A prominent use-case of smart contracts is the management of tokens. These tokens can be sub-currencies or used to track assets independent of the block-chain's native currency.

In this paper, we introduce an extension to the ring confidential transaction to support sub-currencies with the benefit of privacy aware trading. Our construction features multiple coexisting asset types, also known as colours. A transaction can transfer exactly one colour, but the decoy inputs can be from any colour, having the same anonymity set as single-colour privacy aware block-chains. The colour of the transaction is only known to the interacting parties (sender and recipient of the current transaction), but not to anyone else, achieving a fully privacy aware verification of colour conservation from inputs to outputs.

With the help of our new transaction type, all participants of the block-chain can introduce new token types for their own purposes. The consensus verifies that a new colour does not yet exist to prevent unauthorised issuance of existing tokens. All the new tokens will benefit from the privacy aware transactions without the barrier of creating an independent chain per colour. A new block-chain per colour reduces the opportunities for decoys in a transaction which negatively impacts the privacy of the whole system. On top, multiple colours on a single block-chain facilitate future on-chain atomic swap operations between colours.

2 Preliminaries

Our contribution extends the ring confidential transaction (ringCT), which is prominently used in Monero for the `RCTTypeFull`, to support a colour attribute

for in- and outputs of a transaction. In this section, we describe the required building blocks, an additively homomorphic commitment scheme and a linkable ring signature, which are the same as for the ringCT. In addition to this, a full ringCT requires range proofs, but as our extension does not require an adaptation thereof, we refer the reader to the work of Noether et al. [6] for the full construction.

We use elliptic curve cryptography for our commitments and signatures. An elliptic curve is a group with the possibility to add an element, also called point, to another or itself resulting in a new point on the curve. This allows for the multiplication of a scalar x to a point. A curve standardises a base point G , which is the generator of a preferably large subgroup. Elliptic curves are suited for cryptography, as calculating x given $P = xG$ is hard, known as the discrete logarithm problem. This property can be used to generate private-public key pairs ($\text{sk} = x, \text{pk} = xG$).

2.1 Pedersen Commitments

To hide a value a in an Elliptic Curve Pedersen commitment [7] requires two points, where one can be the base point G and the second point $H = \psi G$ must be created, such that ψ is unknown to anyone. A *nothing up my sleeve* generation of H can be generated by hashing the base point with a hash function \mathcal{H} mapping from a point to another point with $H = \mathcal{H}(G)$. G and H are the public parameters of the given system. To build a commitment to a value a , a secret, random blinding factor x is generated and then combined to

$$C(a, x) = xG + aH.$$

Pedersen commitments are perfectly hiding and computationally binding under the discrete logarithm assumption. A commitment $C(0, x) = xG$ is binding but not hiding. By publishing the point $C(a, x)$, the sender commits to the value a , and can only change the choice by brute-force searching for a different pair x', a' satisfying $C(a, x) = xG + aH = x'G + a'H$ which has a negligible chance of success.

The Pederson commitment has the desirable property, that finding the value y with $C(a, x) = yG$ for $a \neq 0$ and $x \neq 0$ is difficult according to the discrete logarithm problem. However, given that $a = 0$, the commitment is reduced to $C(0, x) = xG + 0H = xG$. Then the private key to the committed point is x which is used to sign the commitment and thus proving knowledge of x .

An additional feature of the commitments are their homomorphicity in regard to addition. Three commitments $C_1(5, x_1), C_2(3, x_2), C_3(2, x_3)$ can be summed together like $C_1 - (C_2 + C_3) = x_1G + 5H - (x_2G + 3H + x_3G + 2H) = (x_1 - x_2 - x_3)G$ resulting in a commitment to zero C_0 with secret key $x_1 - x_2 - x_3$. Whoever can sign the sum of the commitments proves knowledge of all the components and proves that the sum of values is 0. For values which should be in plain-text, but which are needed to perform calculations, a commitment can be opened, by immediately disclosing a and x .

2.2 Multilayered Linkable Spontaneous Ad-Hoc Group Signature

The second building block we require for our colour extension is the Multilayered Linkable Spontaneous Ad-Hoc Group Signature (MLSAG). This is a modification of the Fujisaki-Suzuki (FS) [3] and the Liu-Wei-Wong (LWW) [4] signatures to increase their space efficiency. It provides a signature where the signer can prove knowledge of a set of private keys which are embedded in a larger set of decoys. The verifier can not deduce for which subset the signer knows the private keys. If any one of the private keys is reused, the two resulting signatures can be linked together, preventing double spending of a single output. The keygen, sign, verify and link algorithms are described according to Noether et al. [6].

$(P^j, x_j) \leftarrow \mathbf{ML.Keygen}(1^\lambda)$: Generate a vector of m private keys x_i for $i = 1, \dots, m$ with the corresponding public keys $P^i = x_i G$.

$(P_i^j, I_j) \leftarrow \mathbf{ML.Keyselect}(P^j)$: Select a set of $n - 1$ vectors, each containing m public keys $\{P_i^j\}_{i=1, \dots, n}^{j=1, \dots, m}$ from other users. For a secret index π , corresponding to the signer, all the secret keys x_j must be known, such that $x_j G = P_\pi^j$ and let $I_j = x_j \mathcal{H}(P_\pi^j)$ for $j = 1, \dots, m$.

$\sigma \leftarrow \mathbf{ML.Sign}(m, P_i^j, x_j, I_j)$: Let m be the message to sign. For $j = 1, \dots, m$ and $i = 1, \dots, \pi - 1, \pi + 1, \dots, n$ draw s_i^j and α_j as secure, random scalars. With a hash function $\mathfrak{h} : \{0, 1\}^* \rightarrow \mathbb{Z}_q$, compute $L_\pi^j = \alpha_j G$ and $R_\pi^j = \alpha_j \mathcal{H}(P_\pi^j)$ for $j = 1, \dots, m$. Continue with the vector $i = \pi + 1$ as

$$\begin{aligned} c_{\pi+1} &= \mathfrak{h}(m, L_\pi^1, R_\pi^1, \dots, L_\pi^m, R_\pi^m) \\ L_{\pi+1}^j &= s_{\pi+1}^j G + c_{\pi+1} P_{\pi+1}^j \quad \text{and} \quad R_{\pi+1}^j = s_{\pi+1}^j \mathcal{H}(P_{\pi+1}^j) + c_{\pi+1} I_j \end{aligned}$$

and calculate L and R for each increment of $i \pmod n$ until $i = \pi - 1$ like

$$\begin{aligned} c_{\pi-1} &= \mathfrak{h}(m, L_{\pi-2}^1, R_{\pi-2}^1, \dots, L_{\pi-2}^m, R_{\pi-2}^m) \\ L_{\pi-1}^j &= s_{\pi-1}^j G + c_{\pi-1} P_{\pi-1}^j \quad \text{and} \quad R_{\pi-1}^j = s_{\pi-1}^j \mathcal{H}(P_{\pi-1}^j) + c_{\pi-1} I_j. \end{aligned}$$

Given $c_\pi = \mathfrak{h}(m, L_{\pi-1}^1, R_{\pi-1}^1, \dots, L_{\pi-1}^m, R_{\pi-1}^m)$, we calculate s_π^j with $\alpha_j = s_\pi^j + c_\pi x_j \pmod l$ (modulus curve order l) and the output consists of

$$\sigma = (c_1, s_1^1, \dots, s_1^m, s_2^1, \dots, s_2^m, \dots, s_n^1, \dots, s_n^m, I_1, \dots, I_m). \quad (1)$$

0/1 $\leftarrow \mathbf{ML.Verify}(m, \sigma, P_i^j)$: Starting with $i = 1$ and c_1 , calculate L_i^j and R_i^j for all i and j . If $c_{n+1} = c_1$, the signature is valid and 1 is returned, 0 otherwise.

0/1 $\leftarrow \mathbf{ML.Link}(\sigma, \sigma')$: If the signatures σ and σ' share an I_j , they used the same private key x_j in the signing process and 1 is returned, 0 otherwise.

The MLSAG signature scheme must satisfy the the following correctness conditions: For every $\lambda, m \in \mathbb{N}$, every $n \in \mathbb{N} \setminus \{1\}$, every $(P^j, x_j) \leftarrow \mathbf{ML.Keygen}(1^\lambda)$, and every m , it holds with high probability that

$$\mathbf{ML.Verify}(m, \sigma \leftarrow \mathbf{ML.Sign}(m, P_i^j \leftarrow \mathbf{ML.Keyselect}(P^j), x_j, I_j), P_i^j) = 1.$$

The MLSAG satisfies the following security properties which are proven in the original LWW signature description [4] and the construction by Noether et al. [6]:

- *Unforgeability*: negligible probability of producing a valid signature without knowledge of all private keys in one vector.
- *Linkability*: negligible probability of being able to produce two different signatures using the same private key in both.
- *Signer Ambiguity*: negligible additional probability of guessing the secret index, even by knowing private keys of decoy inputs.

3 Our Coloured Ring Confidential Transaction

Having explained the necessary building blocks, we proceed with a detailed description of the `RCTTypeFull` ringCT and highlighted the additional elements required for our extension in `red`.

In- and Outputs. Our transaction requires inputs, which are outputs of previous transactions. The sender selects m inputs to be used. Depending on how many decoys per input ($n - 1$) the sender wants to include in the transaction, additional $m \cdot (n - 1)$ inputs are selected. Each input contains a public key P_i^j , which was generated as a one-time payment address. The amount a each input holds is stored in a Pedersen commitment $C_i^j(a, b)$ with blinding factors b . The sender only knows $a_{j,in}$ and $b_{j,in}$ and x_j for the inputs ($P_\pi^j = x_j G, C_\pi^j(a_{j,in}, b_{j,in})$) under its control. All real inputs make up one vector

$$\{(P_\pi^1, C_\pi^1(a_{1,in}, b_{1,in})), \dots, (P_\pi^m, C_\pi^m(a_{m,in}, b_{m,in}))\}$$

at the secret index π . The decoy vectors at $i = 1, \dots, \pi - 1, \pi + 1, \dots, n$ are assembled equally, with neither knowledge of the private keys for P_i^j nor of the blinding factors and amounts of the commitments C_i^j .

We introduce the colour property as an additional commitment in each input. Colours are defined as scalars f_i . Each input gets an additional commitment F_i^j to a colour. For the sender owned inputs, the colours $f_{j,in}$ and blinding factors $u_{j,in}$ of the commitments $F_\pi^j(f_{j,in}, u_{j,in})$ are known. An input ($P_\pi^j, C_\pi^j(a_{j,in}, b_{j,in}), F_\pi^j(f_{j,in}, u_{j,in})$) is now composed of the recipient one-time key and two commitments. The q outputs of a transaction are also represented as a tuple of three elements ($P_k, C_k(a_{k,out}, b_{k,out}), F_k(f_{out}, u_{k,out})$) for $k = 1, \dots, q$ with the blinding factors $b_{k,out}$ and $u_{k,out}$ randomly drawn and secret.

Conservation. The sum of amounts of all inputs into a transaction must always be greater or equal to the sum of all output amounts, so the plain-text equation $\sum_{j=1}^m a_{j,in} - \sum_{k=1}^q a_{k,out} = 0$ translates to a commitment equation

$$\sum_{j=1}^m C_i^j - \sum_{k=1}^q C_k = C_0^i \quad (2)$$

resulting in a commitment C_0^i to zero. For $i = \pi$ in Eq. (2), the signer knows all amounts $a_{j,in}$ and blinding factors $b_{j,in}$, which make up the private key to C_0^π . This conservation ensures, that no asset is created in a transaction.

To ensure, that the real inputs are all from the same colour, the colour commitments F_i^j are checked in pairs to the colour commitment of the first output F_1 . Again we can use a commitment to zero $F_i^j - F_1 = F_0^{i,j}$, which does not disclose the colour. Unlike the summation of the amounts, comparing aggregate commitments utilising the homomorphic property is not secure and could lead to the following attack. An attacker creates a transaction with two input colours $f_{in} - \epsilon$ and $f_{in} + \epsilon$ and an output f_{out} . If we only verify that $f_{in} - \epsilon + f_{in} + \epsilon = 2f_{out}$ the inputs are not necessarily from the same colour. This conservation rule only supports one colour per transaction. Transactions with multiple colours involved, maintaining the signer ambiguity is supported by an extended version of our scheme currently in development.

Signature. The n commitments from the amounts and $n \cdot m$ commitments from the colour checks can now be signed by an MLSAG from Sect. 2.2. To bind a zero commitment to the originating spend key, and to have independent link tags, the public key is added to the commitment. As the sender knows the private key x_j to the spend key P_π^j and the components of the commitments to colour and value, it can still sign the sum of commitment and P_π^j with $x_{m+1+j} = x_j + f_{j,in} - f_{1,out}$. The following set of vectors is used as key input P_i^j into the **ML.sign**(m, P_i^j, x_j, I_j) algorithm with I_j from the **ML.Keyselect** algorithm:

$$P := \left[\left\{ P_1^1, \dots, P_1^m, \sum_{j=1}^m (P_1^j + C_1^j) - \sum_k C_k, P_1^1 + F_1^1 - F_1, \dots, P_1^m + F_1^m - F_1 \right\}, \right. \\ \dots, \left\{ P_\pi^1, \dots, P_\pi^m, \sum_{j=1}^m (P_\pi^j + C_\pi^j) - \sum_k C_k, P_\pi^1 + F_\pi^1 - F_1, \dots, P_\pi^m + F_\pi^m - F_1 \right\}, \\ \left. \dots, \left\{ P_n^1, \dots, P_n^m, \sum_{j=1}^m (P_n^j + C_n^j) - \sum_k C_k, P_n^1 + F_n^1 - F_1, \dots, P_n^m + F_n^m - F_1 \right\} \right].$$

Output Proofs. The amounts are values modulus the curve order l , so overflows can be used to create new assets in a transaction. To counter this, the ringCT uses range proofs [2, 5] to confine the output amounts to the interval $[0, 2^{64}]$. Our extension has to make sure, that all outputs are commitments to the same colour. We achieve this by appending $q - 1$ signatures for the zero commitments $F_1 - F_k = F_0^k$ for $k = 2, \dots, q$.

Public Verification. The complete transaction with the references of the inputs and outputs and the ring signature σ is broadcast and anyone is able to verify the transaction and the conservation of assets. Therefore the vectors of public keys P_i^j are read from the referenced inputs together with the

amount and colour commitments. The points for checking the conservation are calculated. The transaction is accepted if $\mathbf{ML.Verify}(\sigma, \mathbf{m}, P_i^j) = 1$ and $\mathbf{ML.Link}(\sigma, \sigma') = 0$ for all other transactions σ' .

4 Discussion

In this section we evaluate the theoretical impact of our extension and discuss its implications on the privacy of the whole system.

Correctness. The correctness of the proposed scheme, is satisfied by the availability of a rightful owner of an output and its corresponding key to transfer the funds of one colour to another address. This is given under the correctness of the non-colour aware ringCT. The restriction of real in- and outputs being of the same colour only separates the transactions into different asset types, but within each of them, funds can be transferred.

Size and Performance Overhead. The MLSAG signature size increases significantly compared to a `RCTTypeFull` transaction. The current MLSAG signature (Eq. (1)) requires $(n(m+1) + 1 + m)32 + \epsilon$ Bytes, with ϵ being the size of variable length encoded positions of the ring members. In addition to this, the q outputs require $q(1 + 64 \cdot 2 + 64)32$ Bytes for the Borromean range proofs [5] including signatures and commitments proving a range of 64 bit.

Our extension depends on longer vectors because of the colour equivalency proofs. The signature size then increases by $n \cdot m$ additional random values $s_{1,m+2}, \dots, s_{1,m+1+m}, \dots, s_{n,m+2}, \dots, s_{n,m+1+m}$ to $(n(m+1) + 1 + m)32 + \epsilon$ Bytes. The range proofs for the amount stay exactly the same. To prove that the colours of the outputs are all the same, we need additional $q - 1$ signatures for pairwise commitments to zero.

The range proofs use most of the space of the current transactions, so that our increase in signature size is quite negligible. Only for a high number of inputs, the impact is significant. Comparing only the signature sizes, our new approach requires approximately twice the space. With the introduction of bullet proofs [2] the range proof size will no longer increase linearly, but logarithmically leading to a greater influence of the colour overhead.

Security Analysis. Our construction uses the MLSAG and Pedersen commitments in a unmodified version as black-boxes and can therefore rely on the guarantees provided by these primitives.

The addition of a token colour provides a second attribute with which transactions can be related. In a transaction with multiple inputs, an attacker who knows that referenced inputs have different colours can discard these from the anonymity set. Assuming a worst-case uniform distribution over colours of transactions the probability of selecting a complete decoy vector with the same colour is $\frac{1}{\chi^m}$ and vanishing with the number of inputs m and a total of χ colours.

For a more likely distribution of transaction frequencies modelled by a power law, with most outputs in the native colour, the probability to find a one-colour decoy is higher. A Zipf distribution results in a probability of approximately one in each 20 decoy vectors having 2 equal colours for a transaction with two inputs and a reasonable 200 colours in total.

Initial Colour Creation. The ability to transfer privacy aware coloured tokens requires a token issuing protocol to begin with. A simple way is to allow an additional output in a transaction granting the output address a defined number of tokens in a new colour. With an open colour commitment, the transaction is only valid, if the colour is new with respect to all previous colour initiation transactions. Depending on the usage of the new token type, the amount can be an open commitment, to publicly announce the total supply of the token, or, if not needed, be confidential.

5 Related Work

Confidential Assets. Poelstra et al. [8] created a protocol to hide transaction values and allow the transaction of multiple assets on the same block-chain. They use Pedersen commitments to store the amount of each UTXO and because of it's homomorphic properties, can publicly verify the conservation. To mark different assets, a *asset tag* in the form of a commitment to a curve point is added to each output. These asset tags must be generated, such that no factor in any pair of assets is known to anyone. By the discrete logarithm assumption it is hard to verify that no such factors exist for newly introduced assets. While this is no limitation of block-chains with a predefined number of different assets, the dynamic addition of new asset types by untrusted participants can introduce asset which might have a nontrivial factor to an existing one. Moreover their scheme does not support sender set anonymity.

Hidden in Plain Sight: Transacting Privately on a Blockchain. Oleg Andreev at Chain Inc. also proposed a multi asset transaction¹ with the same fundamental techniques as the confidential assets. They also represent different assets as orthogonal curve points and need to verify that the factors between assets are not known to anyone. As the work before, sender anonymity can not be satisfied by hiding the real transaction input in an anonymity set.

6 Conclusion

We introduced an extension to the ring confidential transaction to support multiple colours of tokens to coexist on one block-chain. The transaction is publicly

¹ <https://blog.chain.com/hidden-in-plain-sight-transacting-privately-on-a-blockchain-835ab75c01cb>.

verifiable to transfer only assets in a single colour, without disclosing it. To achieve a high grade of anonymity, the decoy inputs can be of any token colour. Thereby, we allow for an easy issuance of privacy preserving tokens which benefit from each other by disguising themselves with each other. On top, our approach can use all the existing privacy preserving mechanisms in place, such as an anonymous peer to peer network, to maintain the privacy of the participants. Compared to contending solutions, we only require a small adaptation of an existing protocol.

References

1. Alonso, K.M., Joancomartí, J.H.: Monero-privacy in the blockchain (2018). https://urldefense.proofpoint.com/v2/url?u=https-3A__eprint.iacr.org_2018_535&d=DwICAg&c=vh6FgFnduejNhPPD0fl_yRaSfZy8CWbWnIf4XJhSqx8&r=8pdeajTECKGg7IiCE2coz4KdRwElEr3FQX_jYRdcVprH4Ev8v5ZWCYnpPJWwDKNa&m=02zPFAFc2LLmq8CEh5Rp9WWgx7_hdZiLtoX1FNI.gG8&s=CZn8ZZRRRD_abxthfDUHPJcZLzx6aPb8EUoNY3RuVw&e=
2. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: efficient range proofs for confidential transactions. Technical report, Cryptology ePrint Archive, Report 2017/1066 (2017). <https://eprint.iacr.org/2017/1066>
3. Fujisaki, E., Suzuki, K.: Traceable ring signature. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 181–200. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71677-8_13
4. Liu, J.K., Wei, V.K., Wong, D.S.: Linkable spontaneous anonymous group signature for ad hoc groups. In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) ACISP 2004. LNCS, vol. 3108, pp. 325–335. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-27800-9_28
5. Maxwell, G., Poelstra, A.: Borromean ring signatures (2015)
6. Noether, S., Mackenzie, A.: Ring confidential transactions. *Ledger* **1**, 1–18 (2016)
7. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_9
8. Poelstra, A., Back, A., Friedenbach, M., Maxwell, G., Wuille, P.: Confidential assets. In: Financial Cryptography Bitcoin Workshop (2017). <https://blockstream.com/bitcoin17-final41.pdf>
9. Sasson, E.B., et al.: Zerocash: decentralized anonymous payments from bitcoin. In: 2014 IEEE Symposium on Security and Privacy (SP), pp. 459–474. IEEE (2014)