Reinforcement Learning Toward Decision-Making for Multiple Trusted-Third-Parties in PUF-Cash

Georgios Fragkos*, Cyrus Minwalla†, Jim Plusquellic*, Eirini Eleni Tsiropoulou*
*Dept. of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131-0001
†Financial Technology Research, Bank of Canada, Ottawa, ON, Canada K1A 0G9

***Abstract*—Electronic money is the digital representation of physical banknotes enabling offline and online payments. An electronic e-Cash scheme, termed PUF-Cash was proposed in prior work. PUF-Cash preserves user anonymity by leveraging the random and unique statistical properties of physically unclonable functions (PUFs). PUF-Cash is extended meaningfully in this work by the introduction of multiple trusted third parties (TTPs) for token blinding and a fractional scheme to diversify and mask Alice's spending habits from the Bank. A reinforcement learning (RL) framework based on stochastic learning automata (SLA) is proposed to efficiently select a subset of TTPs as well as the fractional amounts for blinding per TTP, based on the set of available TTPs, the computational load per TTP and network conditions. An experimental model was constructed in MATLAB with multiple TTPs to verify the learning framework. Results indicate that the RL approach guarantees fast convergence to an efficient selection of TTPs and allocation of fractional amounts in terms of perceived reward for the end-users.**

***Index Terms*—e-Cash, Physically Unclonable Functions, Reinforcement Learning, Decision Making, Trusted Third Parties.**

I. Introduction

Electronic means of payments have spawned a hive of economic and technical activity across the world. Recent advances in distributed ledger technology (DLT) have led to the emergence of cryptocurrencies and stable-coins. Their enduring popularity suggests that alternative methods to standard banking solutions are desired in the marketplace. Electronic money (e-money) schemes based on blind signatures and zero-knowledge proofs can imbue the privacy-preserving aspects of DLT while enabling bilateral, denetworked transactions. Often, such solutions rely on a combination of hardware and software security principles to protect the store of value and means of exchange during periods of partial to absent connectivity. One such variant, PUF-Cash, was conceived on the premise that physically unclonable functions can provide the foundation on which an e-Cash protocol can be constructed.

II. Related Work

An electronic version of cash (e-Cash) must contend with the two primary security challenges in digital currency, namely copy (double-spend) protection and counterfeit protection. Chaum, Fiat, and Naor (CFN) explored anonymous payments through blind signatures [1] in the first e-Cash protocol [2]. CFN blind signatures are based on RSA primitives, and a cut-and-choose shared secret protocol for the coin exchange. CFN was intended for online purchases, where the merchant could validate the coin at the time of transaction, eliminating double-spending. Batch RSA techniques were investigated by Schoenmaker [3] to reduce storage and bandwidth requirements in CFN transactions. Brands' e-Cash protocol [4] further expanded upon CFN by replacing the cut-and-choose proofs with a variant of the Sigma protocol. The Sigma protocol has gained renewed popularity as a zero-knowledge proof scheme for various applications in distributed ledger technology (DLT) [5].

CFN e-Cash and variants, as conceived, represented indivisible e-coins. By contrast, Okamato developed a divisible e-Cash scheme, although coin division was restricted to powers of two [6]. Camenisch, Hohenberger, and Lysanskaya (CHL) leveraged Chaum and Okamato to further a divisible e-Cash scheme that was more open [7] [8]. Here, a pseudo-random function embedded within Alice's device could generate coins from a single seed. CHL required the generation of zero-knowledge proofs relying on Fujisaki–Okamoto commitments [9] to ensure that generated coin values were bounded to the finite range supplied by the Bank's initial seed.

Existing e-Cash schemes rely either on factorization or the discrete logarithm problem, both of which are susceptible to Shor's algorithm [10]. Recent forays into quantum-hard electronic cash include the use of lattice-based asymmetric key exchanges to implement blind signatures [11], or tokens constructed on the super-position of quantum states [12] [13].

III. The PUF-Cash Electronic Money System

PUF-Cash is an anonymous, electronic cash protocol reliant on physically unclonable functions (PUFs) that is usable in both online and offline contexts. PUF-Cash does not rely on the properties of discrete logarithm or factorization, and is therefore quantum-hard. Transactions in PUF-Cash compare favourably to CHL: A 4096-bit recommended prime length leads to a proof size of 60 KB per signature. A complete cycle in CHL from withdrawal to deposit for a typical $10 transaction consumes 126.5 KB of data and is static, whereas a PUF-Cash transaction for the same amount consumes 75.5 KB and is linear. This

work expands the original concept of PUF-Cash by introducing the idea of multiple Trusted Third Parties (TTPs). Key contributions are the development of a reinforcement learning (RL) model [14] to optimally select both the subset of available TTPs based on network behaviour, and the fractional division of workload to obfuscate Alice's spending behaviour from the Bank.

Fig. 1 outlines the entities involved in the PUF-Cash protocol, the protocol steps denoted by directed arrows and the sequence number captured in parentheses. The Bank is responsible for issuing and redeeming currency, represented as unitary indivisible tokens, the TTP is responsible for blinding tokens, while Alice and Bob constitute the payer and payee, respectively, in a single transaction. The TTP, Alice and Bob are devices equipped with a PUF. In PUF-Cash, the Hardware Embedded Delay PUF (HELP) [15] in particular, is used, although another PUF with suitable properties may be substituted. The HELP PUF utilizes a challenge-response sequence to authenticate, requiring a set of challenge-response pairs to be stored during enrollment at the Bank. This set is stored in two separate databases at the Bank: a labeled database (where challenges are linked to devices) and an unlabeled (anonymous) database. Note that the unlabelled database records device responses to a set of challenges common to all devices.

Enrollment is required to process transactions. The TTP is first enrolled by recording a series of challenge-response pairs for authentication within the labeled database. Then, the TTP and the Bank establish a long-lived shared secret for secure communications. Once TTP enrollment is complete, end-user devices assigned to Alice and Bob may be enrolled within the database. This is a two-stage process: Two sequences of challenge-response pairs, one set unique and linked to each device and a second set common to all devices, are first recorded in the labelled database. Then, unique responses from a set of common challenges are recorded in the unlabelled, anonymous database. Note that the labelled database serves as an authentication source for a given registered device, whereas the anonymous database establishes set membership without revealing device and user identity. A description of the enrollment process, including the underlying operations, is presented in prior work [16].

A. The PUF-Cash Protocol

Devices can execute transactions once enrolled into the system. All transactions follow the same steps, although the timing is highly variable to support a wide variety of payment options. For simplicity, it is assumed that the Bank manages accounts denominated in local currency for each end-user against which tokens are issued on a unitary basis. The protocol proceeds as follows, with steps illustrated in Fig. 1. Explicit cryptographic details underlying token creation, authentication and secure channel creation are eschewed for brevity:

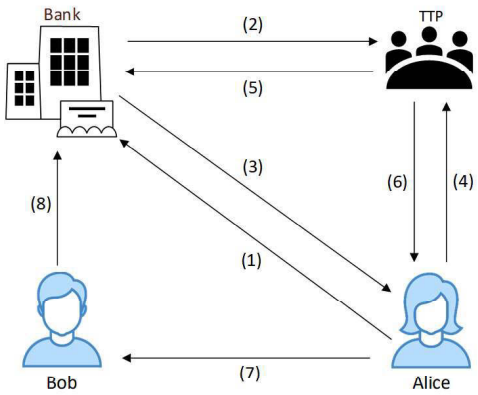

Fig. 1: Steps of the PUF-Cash Protocol

1) Alice requests funds against the balance of her account held at the Bank.
2) Upon request, the Bank issues unitary, immutable, tokens, denoted as n_1, against Alice's account. The Bank records these tokens in a list of open n_1 tokens and forwards them to the TTP.
3) Once TTP acknowledges receipt, the Bank forwards the tokens to Alice and debits her account.
4) At some point in time, Alice contacts the TTP to blind her n_1 tokens. This blinding step is necessary to protect Alice's anonymity at the time of transaction.
5) The TTP receives Alice's request and her n_1 tokens. It then confirms a match between those supplied by Alice and the ones issued by the Bank. For each matched n_1 token, the TTP issues a corresponding n_2 token, then decrements its list of open n_1 tokens and transmits the newly created n_2 tokens to the Bank. The token correspondence is not recorded.
6) Upon receipt of n_2 tokens, the Bank updates its list of blinded n_2 tokens and sends an acknowledgement to the TTP. Once acknowledged, the TTP forwards the n_2 tokens to Alice.
7) Alice identifies Bob as the payee, authenticates Bob via Transport Level Security (TLS) and transfers the requisite amount of n_2 tokens. The token count is based on the amount owed to Bob and may be different from the number blinded in Step 4.
8) Bob deposits the n_2 tokens into the Bank. Upon receipt, the Bank validates each n_2 token against the list of issued blinded tokens. Each token successfully validated is credited to Bob's account.

The system requires the participants to trust the following axioms: (a) That the Bank does not leak information between the labeled and unlabeled data-sets, and (b) that the TTP does not record, share or leak correlating information. If either axiom is violated, the Bank may be able to link issued n_1 tokens to blinded n_2 tokens, thereby compromising Alice's transactional anonymity. The security profile of PUF-Cash and exploration of possible attack

vectors is captured in the prior work [16]. Note that it is up to Alice to decide how many tokens were blinded in Step 4. Also note that Bob may deposit n_2s from multiple sources (not just Alice) in a single batch transfer to the Bank at the time of deposit.

B. Transaction Timelines

In PUF-Cash, the timing between issuance (n_1 creation), blinding (n_1 to n_2 exchange), transaction, and deposit/redemption, is highly variable. This property engenders great flexibility, enabling the protocol to support a wide variety of payment strategies. In particular, it was desirous from inception that the protocol support both online and offline transaction scenarios. To that end, two broad timelines are possible: An offline scenario where there is limited or no connectivity during the actual transaction (Step 7), and an online scenario where the end-user device has access to a network, and by extension, access to the Bank and the TTP.

Fig. 2 presents three timelines under consideration in this work. In each timeline, the abscissa denotes time intervals in units of days. An example time-slice of 14 days is chosen for convenience of explanation. The occurrence of protocol steps are denoted by event markers annotated by the step number in parentheses. Steps executing in rapid succession are captured in a single event with dependent steps occurring sequentially. Note that Step (0) captures enrollment, which may only occur once for the lifetime of the device. A brief textual summary presented per interval captures the action(s) pending at the end of the interval. In Fig. 2, Timeline A captures the offline scenario: Here, Alice withdraws n_1 tokens and subsequently blinds them for transactable n_2 tokens on Day 6. Then, on Day 11, Alice transacts with Bob, transferring a portion of her n_2 tokens. On Day 14, Bob deposits the n_2 tokens in the Bank. Similarly, Timeline B captures the online scenario: Alice withdraws n_1 tokens on Day 4 and holds them locally on her device. On Day 11, Alice transacts with Bob, only converting a portion of her n_1 tokens to n_2 tokens before the transfer. This approach is inherently more secure from Alice's perspective, as the n_1 tokens cannot be stolen (Alice's device is required to convert them), and are recoverable if lost, since the Bank can invalidate open n_1 tokens. However, Alice must be able to connect to the TTP at the time of transaction for her exchange with Bob to be successful. It can be observed that the primary difference between Timelines A and B is the timing of the blinding step, namely Step 4.

Let us consider Timeline C as an extreme variant of Timeline B: Here, Alice holds a trusted device that is registered at the Bank, but does not withdraw n_1 tokens, preferring to store her funds at the Bank for safe-keeping. In addition, Bob is a fully connected payee, such as a merchant terminal or an online payment processor, that deposits (and validates) n_2 tokens as soon as they are received. This scenario, although extreme on the surface,

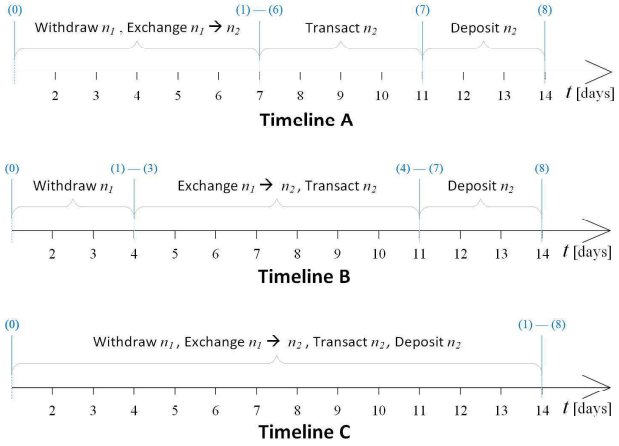

Fig. 2: Protocol Timelines

is likely to be the most common for in-store or online purchases. Alice may consider this approach to be the least risky if her PUF device is a custom IC embedded within a smartphone, where platform security may not be guaranteed. Under these conditions, all protocol steps will occur in a very short timeframe (within seconds), exposing Alice to a potential timing attack from the Bank, who may be able to correlate the issuance of n_1 with the redemption of n_2 tokens to extract her transaction history.

Much of the risk in Timeline C stems from the fact that a singular TTP is responsible for the token blinding step. To thwart this attack and anonymize Alice's transaction history, multiple TTPs are proposed, such that each TTP only processes a fraction of Alice's tokens at any given point in time. In this scenario, when Alice requests a token exchange, the TTP will first request all of Alice's n_1 tokens from the Bank. Alice will then supply a fractional set of n_1 tokens of her choosing for exchange, against which the TTP will generate the equivalent n_2 tokens. In addition, each TTP will have sufficient computational capacity to serve multiple users concurrently, and furthermore, n_2 updates to the Bank will be batched. Under these conditions, if sufficient users are interacting with the system, individual exchanges will be obfuscated, making it impossible for the Bank to discern Alice's behaviour. The optimal selection of TTPs will depend on a combination of the per-TTP available computational capability and prevailing network conditions.

IV. Reinforcement Learning-based PUF-Cash Transaction

Towards diversifying and masking Alice's spending habits from the Bank, a distributed and autonomous reinforcement learning-based mechanism is proposed, enabling Alice to select the TTPs and the corresponding fractional amounts for a successful transaction. Alice (and respectively each end-user in the PUF-Cash environment) acts as a stochastic learning automaton (SLA) [14] mak-

ing optimal decisions about herself, i.e., to which TTPs to offload fractions of her total amount, based on the reaction, i.e., state, of the PUF-Cash environment [17]. Alice's decision-making problem consists of two decision-making layers. In the first layer, Alice decides to which TTPs she will offload fractions of her total amount, while in the second layer she decides the optimal fractions of the total amount to be offloaded to each one of the selected TTPs.

Let us denote as $C = \{1, \ldots, c, \ldots, |C|\}$ the set of end-users in the PUF-Cash environment. The set of the available TTPs is $T = \{TTP_1, \ldots, TTP_t, \ldots, TTP_{|T|}\}$. Each TTP_t has a computation capability $F_{TTP_t}[\frac{CPU\ cycles}{unit\ operation}]$, which was derived from our prototype system using a set of Xilinx 7010 FPGAs on Digilent Cora boards [18]. The "CPU cycles/unit operation" is the effort required to generate each random 16-byte n_2 token. The communication distance of each end-user c with each TTP_t is denoted as $d_{c,TTP_t}[m]$. Furthermore, each end-user possesses a finite quantity of tokens, denoted as M_c, and can offload a fraction of them to a TTP, i.e., $f_{c,TTP_t}\%$. Alice can select N TTPs, i.e., $N \leq |T|$ to distribute her n_1 tokens for conversion, thus her strategy space at the first decision-making layer consists of vectors $\mathbf{STR_s} = [TTP_A, TTP_B, \ldots, TTP_N]$, where $s \in S = \{1, \ldots, s, \ldots, |S|\}$ and $|S|$ is the total number of distinct subsets of the N TTPs. Alice aims to minimize her communication delay and therefore chooses one of these TTP subsets that best meets this goal, likely achieved by choosing TTPs that are in close proximity to her. She also takes into consideration the current workload associated with each TTP, choosing a subset that has sufficient computational capacity to process each of the fractional workloads delivered to them. Based on the communication delay (denominator of Eq.1) and the computational congestion (numerator of Eq.1), both of which are broadcast by the TTPs to the end-users, Alice determines a corresponding personalized reward $r^{(ite)}_{c,\mathbf{STR_s}}$ at the ite iteration of the SLA algorithm in the first decision-making layer as follows.

$$r^{(ite)}_{c,\mathbf{STR_s}} = \frac{\sum\limits_{TTP_t \in \mathbf{STR_s}} \frac{F_{TTP_t}}{|C_{TTP_t}|_{(ite-1)}}}{\frac{\sum\limits_{TTP_t \in \mathbf{STR_s}} d_{c,TTP_t}}{\sum\limits_{\forall TTP_t \in T} d_{c,TTP_t}}} \quad (1)$$

where $|C_{TTP_t}|$ is the number of end-users offloading a fraction of their amount to the TTP_t. The personalized reward $r^{(ite)}_{c,\mathbf{STR_s}}$ of each end-user is normalized as $\hat{r}^{(ite)}_{c,\mathbf{STR_s}} = \frac{r^{(ite)}_{c,\mathbf{STR_s}}}{\sum\limits_{c=1}^{|C|} r^{(ite)}_{c,\mathbf{STR_s}}}$ to reflect the reward probability $0 \leq \hat{r}^{(ite)}_{c,\mathbf{STR_s}} \leq 1$ of end-user c to select the strategy $\mathbf{STR_s}$, i.e., the subset of TTPs to process her transaction. Given the personalized reward probability, each end-user acts as an SLA and determines her action probability vector $\mathbf{Pr^{(ite)}_c} = [Pr^{(ite)}_{c,1}, \ldots, Pr^{(ite)}_{c,\mathbf{STR_s}}, \ldots, Pr^{(ite)}_{c,\mathbf{STR_{|S|}}}]$. Similarly, the end-users' action probabilities are determined based on the SLA update rule [17], as follows.

$$Pr^{(ite+1)}_{s,\mathbf{STR_s}} = Pr^{(ite)}_{s,\mathbf{STR_s}} + b\hat{r}^{(ite)}_{c,\mathbf{STR_s}}(1 - Pr^{(ite)}_{s,\mathbf{STR_s}}),\ ^{(ite+1)}_{\mathbf{STR_s}} = ^{(ite)}_{\mathbf{STR_s}} \quad (2a)$$

$$Pr^{(ite+1)}_{s,\mathbf{STR_s}} = Pr^{(ite)}_{s,\mathbf{STR_s}} - b\hat{r}^{(ite)}_{c,\mathbf{STR_s}} Pr^{(ite)}_{s,\mathbf{STR_s}},\ ^{(ite+1)}_{\mathbf{STR_s}} \neq ^{(ite)}_{\mathbf{STR_s}} \quad (2b)$$

where $0 < b \leq 1$ is the learning parameter (for smaller values of b the end-user explores more of the available strategies). Eq.2a expresses the probability of selecting the same strategy $\mathbf{STR_s}$ in iteration ite, while Eq.2b represents the probability of selecting a different strategy. At each iteration of the SLA approach, the end-users select a strategy, i.e. specific subsets of TTPs to process their transaction amounts, and subsequently experience the environment's reaction, i.e. the communication delay and computation congestion, both of which are captured in the personalized reward probabilities. Over time, this type of iterative learning process enables each end-user to converge to the optimal selection of TTPs. It is noted that the learning process associated with each end-user is initialized such that all strategies $\mathbf{STR_s}$ are equiprobable of being selected.

After the convergence of the first decision-making layer, each end-user c has selected the most efficient combination of TTPs, i.e., $\mathbf{STR^*_s|_c} = [TTP^*_A, TTP^*_B, \ldots, TTP^*_N]$. Then, each end-user determines the fraction $f_{c,TTP^*_t}\%$ of her n_1s that should be offloaded to each of the TTPs as a means of achieving two goals, obfuscation and speed-up. As per Timeline C in Section III-B, the TTPs in Alice's subset request all of Alice's n_1s from the Bank at the point of sale, and then each of the TTPs immediately converts a fraction of them to n_2s, which are then transmitted back to the Bank. Other end-users will make similar requests concurrently, therefore, obfuscation occurs by virtue of each TTP processing multiple fractional requests for a group of end-users. Alice also aims to leverage the available parallelism associated with multiple TTPs to accelerate her overall transaction. Here, it is again possible to use SLA to derive a solution by defining a reward $r^{(ite')}_{c,\mathbf{STR^*_s|_c}}$ that each end-user experiences by offloading a fractional amount $f_{c,TTP^*_t} \cdot M_c$ to the $TTP^*_t, \forall TTP^*_t \in \mathbf{STR^*_s|_c}$ in the iteration ite' of the second decision-making layer, which is defined as follows.

$$r^{(ite')}_{c,\mathbf{STR^*_s|_c}} = \frac{\sum\limits_{\forall TTP^*_t \in \mathbf{STR^*_s|_c}} (f^{(ite')}_{c,TTP^*_t} M_c F_{TTP^*_t})}{\sum\limits_{\substack{\forall c' \in |C_{TTP^*_t}| \\ \forall TTP^*_t \in \mathbf{STR^*_s|_c}}} (f^{(itc')}_{c',TTP^*_t} M_{c'}) \sum\limits_{\forall TTP^*_t \in \mathbf{STR^*_s|_c}} F_{TTP^*_t}} \quad (3)$$

To simplify the problem, we define F discrete levels of fractions, i.e., $[f_{c,TTP^*_t}|_1, \ldots, f_{c,TTP^*_t}|_f, \ldots, f_{c,TTP^*_t}|_F]$ (e.g., $[10\%, 20\%, \ldots, 100\%]$). The strategy vector of end-user c is denoted as $\mathbf{FR_{c,fr}} = [f_{c,TTP^*_1}|_f, \ldots, f_{c,TTP^*_N}|_f]$,

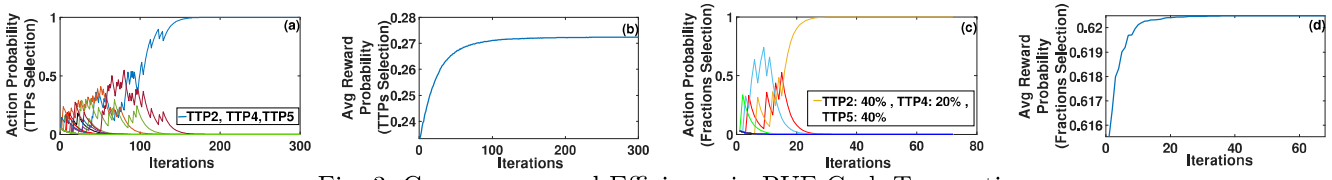

Fig. 3: Convergence and Efficiency in PUF-Cash Transactions

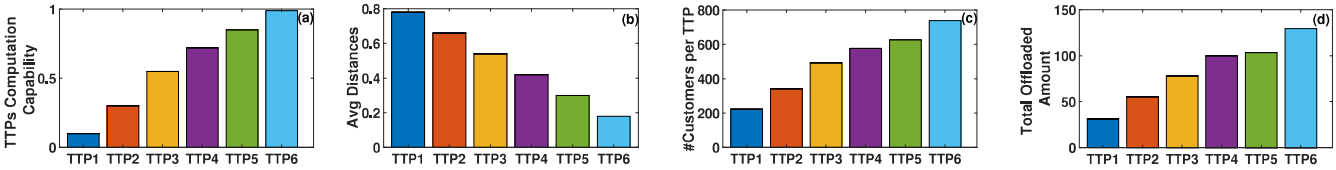

Fig. 4: Overall Operation of the Reinforcement Learning Approach

where $f \in \{1,\ldots,F\}, f_{c,TTP_1^*|_f} + \cdots + f_{c,TTP_N^*|_f} = 1$ and $\mathbf{FR_{c,fr}}$ represents one of the possible combination of fractions of the total amount M_c offloaded to the TTPs included in the strategy $\mathbf{STR_s^*|_c}$ that the end-user has selected in the first layer and $|FR|$ is the total combinations of fractions. Thus, each end-user acting again as a stochastic learning automaton and given the reaction of the PUF-Cash environment captured by the reward probability $\hat{r}^{(ite')}_{c,\mathbf{STR_s^*|_c}} = \frac{r^{(ite')}_{c,\mathbf{STR_s^*|_c}}}{\sum_{c=1}^{|C|}(r^{(ite')}_{c,\mathbf{STR_s^*|_c}})}$, determines the action probability vector $\mathbf{Pr^{(ite')}_{c,fr}} = [Pr^{(ite')}_{c,1},\ldots,Pr^{(ite')}_{c,|FR|}]$, which represents the probability of selecting a specific fractions' combination action $\mathbf{Fr_{c,fr}}, \forall fr \in \{1,\ldots,|FR|\}$ to offload her total amount to the pre-selected subset of TTPs. The end-user's action probability to select the same fractions' combination in the next iteration of the decision-making process is defined in Eq.4a, while the probability to select a different strategy is given by Eq.4b.

$$Pr^{(ite'+1)}_{c,fr} = Pr^{(ite')}_{c,fr} + b'\hat{r}^{(ite')}_{c,\mathbf{STR_s^*|_c}}(1 - Pr^{(ite')}_{c,fr}), \quad ^{(ite'+1)}_{fr} = ^{(ite')}_{fr} \tag{4a}$$

$$Pr^{(ite'+1)}_{c,fr} = Pr^{(ite')}_{c,fr} - b' \cdot \hat{r}^{(ite')}_{c,\mathbf{STR_s^*|_c}} Pr^{(ite')}_{c,fr}, f \; ^{(ite'+1)}_{fr} \neq ^{(ite')}_{fr} \tag{4b}$$

where $0 < b' \leq 1$ is the learning parameter. Again, each end-user is initialized with equal probability of selecting a combination of fractions from the available ones.

To summarize, end-users make autonomous decisions, hidden from the Bank, by sensing the state of the PUF-Cash environment. They first decide which subset of TTPs will service their request, and then decide the fraction of their n_1s to offload to each TTP in the subset.

V. Results and Discussion

This section provides a detailed numerical evaluation of the proposed reinforcement learning-based strategy in terms of the overall framework's operational efficiency and performance. We consider $C = 1000$ end-users performing transactions per timeslot, i.e., $1sec$, $|T| = 6$ TTPs in the system, where the TTPs' computation capability is $F_{TTP_t} = 356[\frac{CPU\ cycles}{unit\ operation}]$. The end-users' distance d_{c,TTP_t} from the TTPs takes random values in the interval $[100, 1000]m$. For presentation purposes only and without loss of generality, each end-user can select $N = 3$ TTPs to offload fractions of her total number M_c of n_1s, where M_c takes random values in the interval [100-20000] tokens, and the learning parameters are $b = b' = 0.7$. Also, each end-user can offload discrete fractions $f_{c,TTPt^*|_f} \in [10\%, 20\%, \ldots, 100\%]$ of her total amount. In the following evaluation, we normalize the end-users' distance from the TTPs and the TTPs' computation capability.

Fig.3 presents the convergence of the overall reinforcement learning-based framework to the most efficient selection of strategies by the end-users. Specifically, Fig.3a presents the convergence of the action probabilities (Eq.2a,2b) of an typical end-user when selecting a subset of TTPs from the available ones. The results reveal that the probability of selecting one strategy converges to one, while the probabilities of selecting any other strategy converge to zero in 200 iterations (7.21 sec). Additionally, Fig.3b presents the average of the end-user's reward probabilities as a function of the number of iterations performed by the SLA algorithm when selecting the end-users' TTPs. The curve shows that the end-users average reward probability increases as the end-users learn to select the most efficient TTP subset, and converges to a large value showing the efficiency of the end-users' choice. Moreover, Fig.3c shows the convergence rate of selecting the fraction of the total amount to offload to each of the TTPs in the selected subset. The results show that the end-users converge quickly, i.e., 60 iterations (1.2 sec) to their most efficient choice of fractions, while the average reward probability of all the end-users converges to a large value (Fig.3d).

Fig.4 illustrates the overall operation of the reinforcement learning-based approach. Specifically, it is observed that more end-users (Fig.4c) select the TTPs with higher computation capability (Fig.4a) - in order to experience decreased computation congestion - and smaller average distance (Fig.4b) - in order to experience decreased communication delay - while they proportionally offload fractions of their total amount (Fig.4d).

Finally, Fig.5 presents a comparative evaluation of the proposed framework to other approaches. Fig. 5a compares

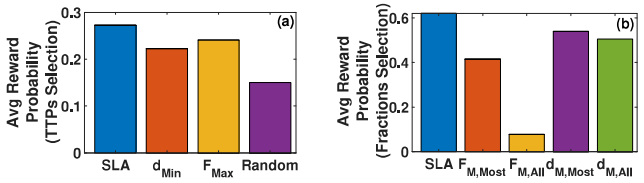

Fig. 5: Comparative Evaluation

the SLA-based TTPs selection to three alternatives, i.e., the end-users select N TTPs based on: (i) closer to them (d_{Min}), (ii) that have the greatest computation capability (F_{Max}), and (iii) randomly. The results illustrate that the SLA-based selection of TTPs provides the greatest average reward probability to the end-users across the two-tier optimization. It can be observed that although (ii) is second-highest from a rewards perspective, it is roughly identical to a single TTP scenario and thus, offers the least amount of obfuscation to Alice. The random selection of TTPs appears to be the least successful, producing persistently low rewards for end-users.

Fig.5b presents a comparative evaluation among the SLA-based selection of fractions of the total amount to offload to the selected TTPs and four alternative scenarios, where (i) the end-user offloads a random fraction greater than 50% of the total amount to the TTP that has the greatest computation capability ($F_{M,Most}$) or (ii) is less distant ($d_{M,Most}$) among the selected ones, while the remaining amount is randomly allocated to the other TTPs in the subset, (iii) the end-user offloads her total amount to the TTP with the greatest computation capability ($F_{M,All}$) or (iv) to the closest TTP ($d_{M,All}$) among the selected ones. The results reveal that the SLA-based selection of fractions of the total amount to be offloaded to the selected TTPs provides the highest average reward probability to the end-users. Also, the communication delay, which is expressed through the end-users' distance from the TTPs, becomes the dominant factor to the end-users' perceived satisfaction when deciding on the fractions to distribute to the TTPs, while the TTPs' computation capability plays a smaller role.

VI. Conclusion

In this paper, we proposed an improvement to PUF-Cash by introducing multiple trusted-third-parties (TTP) for obfuscating Alice's transaction history from the Bank in rapid withdrawal and spend scenarios. A reinforcement learning approach was proposed to optimize TTP selection and subsequent fractional splitting based on network conditions and computational load. Both decision-making layers are realized using stochastic learning automata, where end-user devices make local autonomous decisions. A numerical evaluation of the proposed strategy is presented in terms of the overall framework's operational efficiency and performance. Results suggest that the two-tier RL SLA strategy, under representative conditions, converges rapidly and guarantees convergence. Furthermore, the technique demonstrates the greatest average reward probability per end-user, both in the selection of TTPs and the fractional split per chosen TTP.

Acknowledgment

The research of Mr. Fragkos and Dr. Tsiropoulou was conducted as part of the NSF CRII-1849739.

References

[1] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology*. Boston, MA: Springer US, 1983, pp. 199–203.

[2] D. Chaum, A. Fiat, and M. Naor, "Untraceable electronic cash," in *Advances in Cryptology*, S. Goldwasser, Ed. Springer New York, 1990, pp. 319–327.

[3] B. Schoenmakers, *Security Aspects of the Ecash™ Payment System*. Springer Berlin Heidelberg, 1998, pp. 338–352.

[4] S. Brands, "Untraceable off-line cash in wallet with observers," in *Advances in Cryptology - CRYPTO' 93*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 302–318.

[5] J. Groth and M. Kohlweiss, "One-out-of-many proofs: Or how to leak a secret and spend a coin," in *Advances in Cryptology - EUROCRYPT 2015*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 253–280.

[6] T. Okamoto, "An efficient divisible electronic cash scheme," in *Advances in Cryptology - CRYPT0' 95*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 438–451.

[7] J. Camenisch, S. Hohenberger, and A. Lysyanskaya, "Compact e-cash," in *Advances in Cryptology - EUROCRYPT 2005*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 302–321.

[8] J. Camenisch, A. Lysyanskaya, and M. Belenkiy, "Endorsed e-cash," in *Proc. IEEE Symposium on Security and Privacy*. IEEE Computer Society, June 2007.

[9] E. Fujisaki and T. Okamoto, "Statistical zero knowledge protocols to prove modular polynomial relations," in *Advances in Cryptology - CRYPTO '97*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 16–30.

[10] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, Jan 1999.

[11] M. Rückert, "Lattice-based blind signatures," in *Advances in Cryptology - ASIACRYPT 2010*, M. Abe, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 413–430.

[12] S. Aaronson, "Quantum copy-protection and quantum money," in *Proc. IEEE Conference on Computational Complexity*, Jul 2009.

[13] S. Aaronson and P. Christiano, "Quantum money from hidden subspaces," in *Proc. ACM Symposium on Theory of Computing*, 2012, pp. 41–60.

[14] P. Mars, *Learning algorithms: theory and applications in signal processing, control and communications*. CRC press, 2018.

[15] J. Aarestad, J. Plusquellic, and D. Acharyya, "Error-tolerant bit generation techniques for use with a hardware-embedded path delay PUF," in *Proc. IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, June 2013, pp. 151–158.

[16] J. Calhoun, C. Minwalla, C. Helmich, F. Saqib, W. Che, and J. Plusquellic, "Physical unclonable function (PUF)-based e-cash transaction protocol (PUF-Cash)," *Cryptography*, vol. 3, no. 3, 2019.

[17] Y. Xu, J. Wang, and Q. Wu, "Distributed learning of equilibria with incomplete, dynamic, and uncertain information in wireless communication networks," in *Game Theory Framework Applied to Wireless Com. Net.* IGI Global, 2016, pp. 63–86.

[18] *Cora Z7 Reference Manual*, Digilent Corporation, Pullman, WA, May 2018. [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/Digilent%20PDFs/Cora_Z7_RM_Web.pdf