This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCE.2020.3024512, IEEE Consumer Electronics Magazine

Artificially Intelligent Electronic Money

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 or e-Cash is becoming increasingly popular as the preferred strategy for making purchases, both on- and off-line. Several unique attributes of e-Cash are appealing to customers, including the convenience of always having 'cash-on-hand' without the need to periodically visit the ATM, the ability to perform peer-to-peer transactions without an intermediary, and the peace of mind associated in conducting those transactions privately. Equally important is that paper money provides customers with an anonymous method of payment, which is highly valued by many individuals. Although anonymity is implicit with fiat money, it is a difficult property to preserve within e-Cash schemes. In this paper, we investigate several artificial intelligence (AI) approaches for improving performance and privacy within a previously proposed e-Cash scheme called PUF-Cash. PUF-Cash utilizes physical unclonable functions (PUFs) for authentication and encryption operations between Alice, the Bank and multiple trusted third parties (mTTPs). The AI methods select a subset of the TTPs and distribute withdrawal amounts to maximize the performance and privacy associated with Alice's e-Cash tokens. Simulation results show the effectiveness of the various AI approaches using a large test-bed architecture.**

I. Introduction

Electronic money or cash, frequently termed e-Cash for short, refers to a system in which physical money, i.e., coins and paper bills, is substituted by digital tokens. Secure exchange, deposit and withdrawal functions are typically accomplished by a message exchange protocol. The protocol, in a potential combination with secure hardware in an offline scenario, must guarantee the security of the tokens against all threats. Customers enrolled in an e-Cash service can use their devices to purchase goods and services. To be as cash-like as possible, e-Cash must provide strong privacy guarantees such that users can remain anonymous in their transactions to both adversaries and system authorities. Beyond privacy, other challenges to implementing e-Cash include counterfeit and double-spending protection, the prevention of fraud, and the recovery of funds in the event of a theft or loss.

II. Related Work

The popularity of e-Cash will eventually proliferate to a wide range of demographics, both rural and urban, but will grow most rapidly within the context of the Smart City as the supporting infrastructure evolves to support it [1], [2]. However, privacy remains a key limiter in its wide-spread adoption, a property recognized as critical early on. For example, Chaum, Fiat, and Naor (CFN) addressed privacy in e-Cash systems by introducing blind signatures in [3] and later proposed their use in the first e-Cash protocol [4]. Online counterfeit protection enables a merchant to validate tokens at the time of transaction. Brand proposed a variant of the Sigma protocol as an alternative to replace the original cut-and-choose proof within CFN [5]. Camenisch, Hohenberger, and Lysanskaya (CHL) [6] proposed a divisible e-Cash scheme by simplifying the proof requirement and introducing a pseudo-random function embedded within Alice's device. Use of distributed ledger technologies (DLTs), in particular tokens hosted on blockchains, are proposed as viable alternatives for digital currency. While DLTs can satisfy the electronic cash requirements of counterfeit, double-spend and privacy preservation, they lack offline support as transactions are not finalized until written to the main ledger.

Existing e-Cash schemes leveraging factorization or discrete logarithm mathematical operations,

which are known to be susceptible to Shor's algorithm [7], are not secure against quantum computing attacks. Recent advances in quantum-hard alternatives to the blind-signature generation, such as lattice-based asymmetric key exchanges [8] show progress in this area. Post-quantum e-Cash schemes, such as [9], based on constructing e-Cash tokens using the superposition of quantum states have also been proposed in recent literature.

A. Contributions

The original PUF-Cash [10] was expanded by introducing multiple trusted third parties (TTPs) and reinforcement learning via stochastic learning automata (SLA) to improve performance and to introduce non-determinism in the optimal selection of TTPs [11]. To the best of our knowledge, this is the first research work in the existing literature that modifies and extends the multiple TTPs approach by investigating a wider range of reinforcement learning methods for executing the blinding process. Key contributions of this paper can be summarized as follows:

- Introduction of a Master TTP to coordinate the blinding process on behalf of the customer, e.g., Alice, and to prevent attempts by Alice to double spend. Moreover, Alice chooses the Master TTP from a set of existing TTPs through a process that better preserves her anonymity with respect to the Bank.
- Investigation of alternative reinforcement learning algorithms to autonomously optimize the selection and workload of Slave TTPs tasked with performing the blinding operation.
- Comparative evaluation and analysis of the proposed learning algorithms to SLA from a performance perspective.

III. The PUF-Cash Multi-TTP Protocol

PUF-Cash is an anonymous, electronic cash protocol that defines a set of transactions between a set of entities, namely, the customers, e.g., Alice and Bob, a set of TTPs, and the Bank. PUF-Cash preserves customer anonymity using two distinct mechanisms. First, the customers and the TTPs authenticate using a privacy-preserving PUF-based authentication protocol, which prevents adversaries from tracking their transactions [12]. Second, the customers engage a set of TTPs to fractionize and convert from issued tokens to blinded tokens, reducing the Bank's ability of establishing the correct correspondence between the two sets of tokens. In this paper, we focus on improving this second mechanism. The proposed multiple TTP architecture also addresses performance bottlenecks associated with a single TTP architecture [10] by distributing system load, which in turn, maximizes throughput and automatically adjusts with dynamically changing environmental conditions.

Fig. 1 illustrates the entities and message exchange operations associated with the blinding process, which is a subset of the overall protocol. Protocol steps are ordered by the circled numbers and the arrows indicate the direction of the information exchange. The following section elaborates on this component of the protocol, namely the issuance and blinding steps.

A. Issuance and Blinding Steps

1) Alice withdraws funds from her account at the Bank. The Bank generates a unique token (128-bit random number) for each 1 cent of the withdrawal amount, then encrypts them using a PUF-generated session key and transmits them to Alice. The Bank records the issued tokens and Alice's session key in a database of open transactions.
2) Alice randomly selects a Master TTP, TTP_M, XOR encrypts her issued tokens with her session key and transmits them to TTP_M.
3) TTP_M transmits Alice's encrypted issued tokens to the Bank. The Bank validates the issued tokens by XOR decrypting them with each of the session keys stored in the database of open transactions, searching for a match to those stored in the database. All matching tokens are removed from the database to prevent double spending. If validated, the Bank sends Alice's session key and an acknowledgement to TTP_M.
4) TTP_M initiates the blinding operation: It runs an AI algorithm to select a subset of Slave TTPs from the set TTP_1 to TTP_n. A second instance of an AI algorithm runs to determine a set of fractions, which dictate how the issued tokens are partitioned among the set of Slave TTPs for blinding. The AI algorithms utilized for these operations are presented in Section

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCE.2020.3024512, IEEE Consumer Electronics Magazine

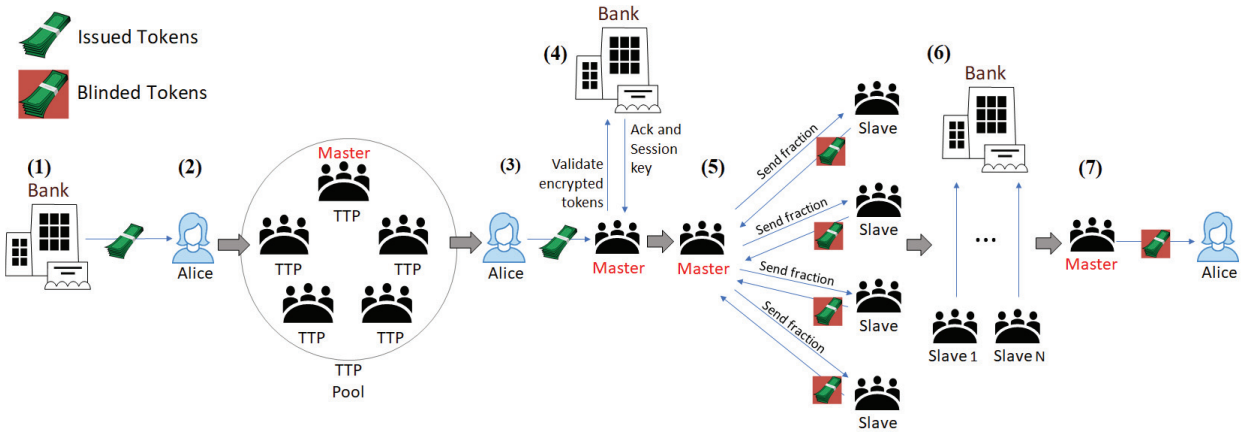

Fig. 1: Operations within the PUF-Cash Protocol responsible for exchanging issued e-Cash Tokens for blinded e-Cash Tokens

IV. Since all TTPs are enrolled in a trusted network, all messages between the Slave TTPs are transferred over encrypted channels.

5) The Slave TTPs generate a new blinded token (128-bit random number) for each 1 cent in the conversion request from TTP_M. The Slave TTPs encrypt the blinded tokens using their session keys and transmit them to the Bank. The Bank acknowledges receipt and adds them to a second blinded token database of open transactions.
6) Once acknowledged, each Slave TTP transmits the blinded tokens to TTP_M.
7) TTP_M encrypts the aggregate set of blinded tokens with Alice's session key and transmits them to Alice.

The PUF-Cash protocol addresses the core components of e-Cash. The issued token database and validation process carried out by the Bank (step 3) protects against double spending and provides a mechanism to recover lost funds. The random number generation and encrypted communication between all entities in the blinding operation guards against man in the middle and replay attacks. Use of AI algorithms for tasks distribution associated with the generation of Alice's blinded tokens across the Slave TTPs improves performance and obscures the correspondence between issued and blinded tokens to preserve Alice's transactional anonymity.

B. Transaction Timelines

The timeline associated with the PUF-Cash protocol (Fig. 1) is highly variable, as arbitrary delays can exist between Alice's withdrawal (step 1), the blinding process (steps 2-7), the value transfer operation between Alice and Bob (step 8) and Bob's deposit to the Bank (step 9). As the protocol was designed to support both online and offline transaction scenarios, it necessitated that each of these major sequences operates independently of each other. Thus, Alice can withdraw issued tokens at some point and delay blinding for days or weeks. Tokens cannot be stolen since only Alice's device is capable of carrying out the blinding operation.

More rapid timelines are possible, and it follows that a timeline that minimizes delays between the initial and final components of the exchange represents the weakest level of privacy for the protocol. This scenario is highly probable in scenarios where Alice engages in an online purchase and the payee (Bob) is a fully connected merchant, e.g., a merchant terminal or an online payment processor. Here, Alice delays the exchange operation after withdrawing (to guard against theft) and then carries out the exchange and value transfer operations in rapid succession. The expanded multi-TTP architecture first proposed in [11] adds significantly to the privacy over the single TTP version proposed in the original PUF-Cash protocol [10]. However, the exploration of an AI approach to obscure the relationship between issued and blinded tokens was limited to one algorithm in [11], and is expanded here to include a wider range of AI algorithms.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCE.2020.3024512, IEEE Consumer Electronics Magazine

IV. Artificial Intelligent Transactions

In this section, we explore AI algorithms to automate the performance and anonymity-enhancing tasks associated with the TTP exchange operation, namely, (1) the selection of a set of TTPs and (2) the partitioning of Alice's withdrawal amount across the selected TTPs. From Fig. 1, Alice begins the issued-to-blinded-token exchange operation by randomly selecting a Master TTP, TTP_M. TTP_M then selects a subset of TTPs based on the theory of the Learning Automata (LA). Once selected, TTP_M determines the Slave offloaded fractional amounts of issued tokens using a reinforcement learning (RL) approach. We investigate a variety of RL approaches, including Linear Reward Inaction, Binary and Max log linear, and Optimistic Q-learning with Upper Bound Confidence (Q_{UC}) and compare their drawbacks and benefits.

A. Slave TTP Selection

Let $U = \{1, \ldots, u, \ldots, |U|\}$ and $T = \{1, \ldots, t, \ldots, |T|+1\}$ denote the sets of customers and TTPs, respectively. Given that customers immediately choose a Master TTP, we use u in the following to refer to a customer's TTP_M. Each TTP t has a computation capability $F_t[\frac{\textit{CPU cycles}}{\textit{unit operation}}]$. Each Master TTP u is physically separated from a Slave TTP t at a distance of $d_{u,t}[m]$. Each TTP_M selected by a customer has M_u issued tokens and can offload $f_{u,t}\%$ of them to a Slave TTP t. TTP_M can select $N \leq |T|$ Slave TTPs, thus, the TTP_M's selection strategy is given by $\mathbf{s} = [1, \ldots, t, \ldots, N]$ and its strategy space is $\mathcal{S} = \{\mathbf{1}, \ldots, \mathbf{s}, \ldots, \mathbf{S}\}$. TTP_M wants to minimize its communication and computing delay, thus, it selects Slave TTPs in its close proximity which have high available computation capacity. The reward that TTP_M experiences by choosing a strategy $\mathbf{s}$ is defined below,

$$r_{u,\mathbf{s}}^{(ite)} = \frac{\sum\limits_{t \in \mathbf{s}} \frac{F_t}{|U_t|_{(ite-1)}}}{\frac{\sum\limits_{t \in \mathbf{s}} d_{u,t}}{\sum\limits_{t \in T} d_{u,t}}} \tag{1}$$

where $|U_t|$ is the number of users engaging TTP t. TTP_M acts as an LA making probabilistic iterative decisions (the iteration is denoted as (ite)) until converging to a stable decision. Eq. 1 is normalized as $\hat{r}_{u,\mathbf{s}}^{(ite)} = \frac{r_{u,\mathbf{s}}^{(ite)}}{\sum\limits_{u \in U} r_{u,\mathbf{s}}^{(ite)}}$ to reflect the reward probability. TTP_M's probability to select the same (Eq. 2a) or different strategy (Eq. 2b) is given below.

$$P_{u,\mathbf{s}}^{(ite+1)} = P_{u,\mathbf{s}}^{(ite)} + \lambda_1(1 - P_{u,\mathbf{s}}^{(ite)}) - \lambda_2(1 - \hat{r}_{u,\mathbf{s}}^{(ite)})P_{u,\mathbf{s}}^{(ite)}, \quad \mathbf{s}^{(ite+1)} = \mathbf{s}^{(ite)} \tag{2a}$$

$$P_{u,\mathbf{s}}^{(ite+1)} = P_{u,\mathbf{s}}^{(ite)} - \lambda_1 \hat{r}_{u,\mathbf{s}}^{(ite)} P_{u,\mathbf{s}}^{(ite)} + \lambda_2(1 - \hat{r}_{u,\mathbf{s}}^{(ite)})(\frac{1}{|\mathcal{S}| - 1} - P_{u,\mathbf{s}}^{(ite)}), \mathbf{s}^{(ite+1)} \neq \mathbf{s}^{(ite)} \tag{2b}$$

where $\lambda_1, \lambda_2 \in [0, 1]$ are the learning parameters, and based on them, three different LA approaches are possible: (1) Linear Reward Penalty (LRP), where $\lambda_1 = \lambda_2$; (2) Linear Reward-ϵ Penalty (LR-ϵP), where $\lambda_1 >> \lambda_2$; (3) Linear Reward Inaction (LRI), where $\lambda_2 = 0$. Here, each TTP_M performs a thorough, a less thorough, and a very limited exploration of its available strategies, respectively. Following convergence, each TTP_M has selected the most efficient combination of Slave TTPs $\mathbf{s}^*$. A detailed comparative evaluation of the LRP, LR-ϵP, and LRI TTPs selection algorithms is presented in Section V-A.

B. Autonomous Offloading of Fractional Amounts to Slave TTPs

Following the selection process, TTP_M then determines the fraction $f_{u,t}\%$ that it will offload to each selected Slave TTP $t^* \in \mathbf{s}^*$, with the goal of maximizing performance and privacy. Regarding the worst case privacy scenario outlined in Section III-B, TTP_M validates the issued tokens and immediately sends requests to the Slave TTPs to generate blinded tokens, which are then transmitted to the Bank. Other TTP_M will make requests concurrently, thereby obscuring Alice's issued tokens among the multiple fractional requests for other customers. The parallel processing capability provided by multiple Slave TTPs also reduces the turn-around time for Alice's exchange operation.

TTP_M's strategy is $\mathbf{F_{u,i}} = [f_{u,1^*}, \ldots, f_{u,t^*}, \ldots, f_{u,N^*}]$, $\sum\limits_{t^*=1}^{N^*} f_{u,t^*} = 1$, $i \in I = \{1, \ldots, |I|\}$, where I is the number of offloading combinations. By selecting strategy $\mathbf{F_{u,i}}$, TTP_M experiences the following reward,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCE.2020.3024512, IEEE Consumer Electronics Magazine

$$r_{u,\mathbf{F_{u,i}}}^{(ite)'} = \frac{\sum_{t^* \in \mathbf{S}^*} (f_{u,t^*}^{(ite')} M_u F_{t^*})}{\sum_{u' \in |U_{t^*}|, t^* \in \mathbf{S}^*} (f_{u',t^*}^{(ite')} M_{u'}) \sum_{t^* \in \mathbf{S}^*} F_{t^*}} \quad (3)$$

which can be normalized as $\hat{r}_{u,\mathbf{F_{u,i}}}^{(ite')} = \frac{r_{u,\mathbf{F_{u,i}}}^{(ite')}}{\sum_{u \in U} r_{u,\mathbf{F_{u,i}}}^{(ite')}} \in [0,1]$ to reflect the reward probability. Four different artificial intelligent RL algorithms are investigated to determine Alice's stable offloading decision, namely: (1) LRI, which belongs to the category of gradient ascent algorithms; (2) Binary log linear learning (BLLL); (3) Max log linear learning (MLLL); and (4) Optimistic Q-learning with Upper bound Confidence action selection (Q_{UC}). The probability of selecting a strategy $\mathbf{F_{u,i}}$ in iteration ite' of each algorithm is denoted as $P_{u,\mathbf{F_{u,i}}}^{(ite')}$. The probabilistic selection of an offloading strategy considering the LRI gradient ascent algorithm follows the rule of Eq. 2a,2b by utilizing the reward probability $\hat{r}_{u,\mathbf{F_{u,i}}}^{(ite')}$ and all offloading combinations $|I|$ in Eq. 2b.

A benefit provided by the log linear algorithms is that they enable convergence to the best equilibrium compared to the LRI algorithm (and in general the gradient ascent algorithms) by allowing a more thorough exploration of their strategy space. In the BLLL and MLLL algorithms, a random TTP_M selects an alternative strategy $\mathbf{F'_{u,i}}^{(ite')}$ with equal probability (while the other TTP_M keep the same actions) and receives the corresponding normalized reward $\hat{r'}_{u,\mathbf{F'_{u,i}}^{(ite')}}^{(ite')}$ during the exploration phase. In the learning phase, the TTP_M selects different ($\mathbf{F_{u,i}}^{(ite'+1)} = \mathbf{F'_{u,i}}^{(ite')}$) or the same ($\mathbf{F_{u,i}}^{(ite'+1)} = \mathbf{F_{u,i}}^{(ite')}$) offloading strategy based on Eq. 4a,4c and Eq. 4b,4d for the BLLL and MLLL algorithms, respectively.

$$P_{u,\mathbf{F_{u,i}}}^{(ite'+1)} = \frac{e^{\hat{r'}_{u,\mathbf{F'_{u,i}}^{(ite')}}^{(ite')} \cdot \beta}}{e^{\hat{r'}_{u,\mathbf{F'_{u,i}}^{(ite')}}^{(ite')} \cdot \beta} + e^{\hat{r}_{u,\mathbf{F_{u,i}}^{(ite')}}^{(ite')} \cdot \beta}} \quad (4a)$$

$$P_{u,\mathbf{F_{u,i}}}^{(ite'+1)} = \frac{e^{\hat{r}_{u,\mathbf{F_{u,i}}^{(ite')}}^{(ite')} \cdot \beta}}{e^{\hat{r'}_{u,\mathbf{F'_{u,i}}^{(ite')}}^{(ite')} \cdot \beta} + e^{\hat{r}_{u,\mathbf{F_{u,i}}^{(ite')}}^{(ite')} \cdot \beta}} \quad (4b)$$

$$P_{u,\mathbf{F_{u,i}}}^{(ite'+1)} = \frac{e^{\hat{r'}_{u,\mathbf{F'_{u,i}}^{(ite')}}^{(ite')} \cdot \beta}}{\max\{e^{\hat{r'}_{u,\mathbf{F'_{u,i}}^{(ite')}}^{(ite')} \cdot \beta}, e^{\hat{r}_{u,\mathbf{F_{u,i}}^{(ite')}}^{(ite')} \cdot \beta}\}} \quad (4c)$$

$$P_{u,\mathbf{F_{u,i}}}^{(ite'+1)} = \frac{e^{\hat{r}_{u,\mathbf{F_{u,i}}^{(ite')}}^{(ite')} \cdot \beta}}{\max\{e^{\hat{r'}_{u,\mathbf{F'_{u,i}}^{(ite')}}^{(ite')} \cdot \beta}, e^{\hat{r}_{u,\mathbf{F_{u,i}}^{(ite')}}^{(ite')} \cdot \beta}\}} \quad (4d)$$

Here, $\beta \in \mathbb{R}^+$ controls the degree to which the TTP_M can explore alternative strategies. The nature of the MLLL probabilistic update rule (Eq. 4c,4d) enables faster convergence to larger (and better) normalized rewards (see Section V-B,V-C).

An alternative RL-based decision making model for offloading is based on the Q_{UC} algorithm [13]. Each TTP_M determines its Q-value (experienced reward) in ite' iterations following the update rule $Q_{\mathbf{F_{u,i}}^{(ite')}}^{(ite')} = Q_{\mathbf{F_{u,i}}^{(ite')}}^{(ite')} + \gamma \cdot (\hat{r}_{u,\mathbf{F_{u,i}}^{(ite')}}^{(ite')} - Q_{\mathbf{F_{u,i}}^{(ite')}}^{(ite')})$ and selects a strategy as follows:

$$\mathbf{F_{u,i}}^{(ite')} = \arg\max_{\mathbf{F_{u,i}}^{(ite')}} [Q_{\mathbf{F_{u,i}}^{(ite')}}^{(ite')} + c \cdot \sqrt{\frac{\ln(ite')}{N_{(ite')}(\mathbf{F_{u,i}}^{(ite')})}}] \quad (5)$$

Here, $N_{(ite')}(\mathbf{F_{u,i}}^{(ite')})$ denotes the number of times the strategy $\mathbf{F_{u,i}}^{(ite')}$ has been selected prior to iteration ite'. The physical meaning of Eq. 5 is that a TTP_M selects a strategy that not only maximizes its Q-value, but also considers the upper confidence bound as expressed by the square-root component. The upper confidence bound measures the uncertainty in the estimate of $\mathbf{F_{u,i}}^{(ite')}$'s value, where $c \in \mathbb{R}^+$ captures the confidence level. It can be observed that if a strategy $\mathbf{F_{u,i}}^{(ite')}$ is selected, the uncertainly decreases, while the opposite holds true if the strategy is not selected. Finally, it should be noted that the term $\ln(ite')$ causes the increases to get smaller over time. Thus, the strategies that have been explored frequently will be selected with decreasing frequency over time, ensuring that all strategies will eventually be explored.

V. Results and Discussion

The effectiveness of the artificial intelligence assisted processes within the PUF-cash protocol are evaluated in this section. In particular, we evaluate TTP_M processes associated with: (1) The selection of Slave TTPs (Section V-A); (2) The fractionalization of issued tokens to the selected subset of Slave TTPs (Section V-B). Finally, in Section V-C, we compare the results from the various AI algorithmic approaches. The evaluation is carried out with the parameters assigned as follows: $|U| = 1000$, $|T| = 6$, $F_t = 356[\frac{CPU\ cycles}{unit\ operation}]$, $d_{c,t} \in [100, 1000]m$,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCE.2020.3024512, IEEE Consumer Electronics Magazine

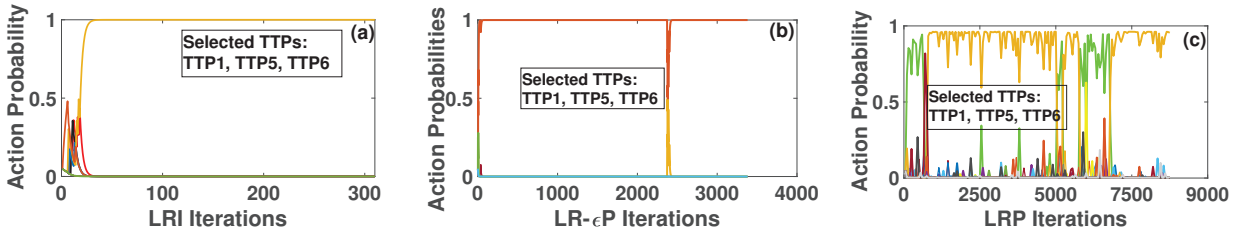

Fig. 2: Gradient Ascent Learning Algorithms – Convergence

$N = 3$, $M_u \in [100, 20000]$ tokens, $c = 2$, $\gamma = 0.6$, $f_{u,t} \in [10\%, 20\%, \ldots, 100\%]$. Algorithmic specific parameters are assigned as follows: for the gradient ascent RL algorithms $\lambda_1 = 0.7$, for the LRI approach $\lambda_2 = 0$, for the LR-ϵP approach $\lambda_2 = 0.001$ and for the LRP approach $\lambda_2 = 0.7$.

A. Reinforcement Learning-based TTPs Selection

In this section, a comparative evaluation of the gradient ascent LRP, LR-ϵP, and LRI RL algorithms is carried out to show the drawbacks and benefits with respect to Slave TTP selection process. Fig. 2 shows the convergence to a stable selection for a typical TTP_M using each of the three gradient ascent algorithms. The graphs reflect the expected result that more iterations are needed for convergence when the solution space is thoroughly explored as is true for LRP in Fig. 2c, while fewer iterations are required for LR-ϵP - Fig. 2b, and even fewer for LRI - Fig. 2a, for lightly explored solution spaces. The real execution time of the three gradient ascent RL algorithms is presented in Fig. 3b, while the corresponding average reward of the TTP_M is shown in Fig. 3a. Thus, we conclude that more thorough exploration leads to higher rewards at the expense of longer execution times. The trade-off between the reward and the execution time is also shown in Fig. 3b which plots the ratio of the normalized reward to normalized real execution time, where normalization is performed by dividing the sum of the rewards by time. The results reveal that the LRI algorithm outperforms the LRP and LR-ϵP algorithms due to the extremely low real execution time and achieved rewards.

B. Artificial Intelligence-enabled Slave TTPs Offloading

A comparative evaluation among the gradient ascent, log linear, and Q_{UC} RL algorithms is

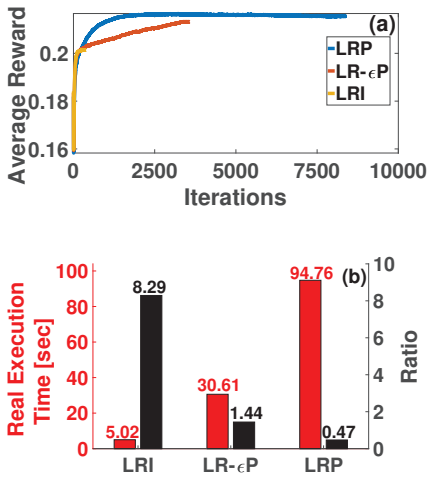

Fig. 3: Gradient Ascent RL Algorithms - Comparative Evaluation

carried out here to determine the effectiveness of Slave TTP offloading strategies. Fig. 4 presents the convergence of the LRI, BLLL, MLLL, and Q_{UC} algorithms. The results reveal that the log linear (BLLL, MLLL) and Q_{UC} algorithms require more iterations than the LRI algorithm to converge to a stable decision, as they explore more thoroughly the available strategies or converge to the best equilibrium, respectively, while achieving higher TTP_Ms' average rewards. Also, it is clear that the MLLL algorithm converges faster when compared to BLLL given the form of the probabilistic update rule (Eq. 4c,4d). A higher value of the learning parameter β for the log linear algorithms results in higher TTP_Ms' average rewards at the expense of longer execution times.

Fig. 6 shows the results for the Q_{UC} algorithm. Fig. 6a illustrates the uncertainty associated with a TTP_M offloading strategy, showing two strategies where the TTP_M converged, i.e., strategy #3 and #9, as well as the number of times that each strategy is selected before convergence occurs. The results reveal that strategies which are selected more often produce lower uncertainties, as the TTP_M has

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCE.2020.3024512, IEEE Consumer Electronics Magazine

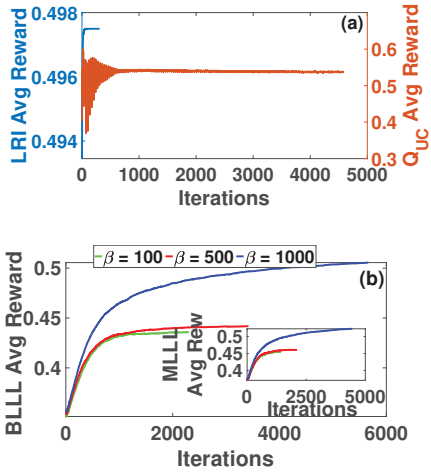

Fig. 4: Gradient Ascent, Log Linear, Q-Learning RL Algorithms – Average Rewards

explored the solution space more thoroughly. This is confirmed by the corresponding Q-value for these two strategies (Fig. 6b), which shows the strategy that the TTP_M finally selects generates a greater Q-value (reward).

C. Comparative Evaluation

In this section, the gradient ascent, log linear, and Q_{UC} RL algorithms are compared with respect to the TTP_Ms' average reward, real execution time, and the ratio of the normalized average reward over the normalized real execution time (Fig. 5) when tasked with selecting the fractional amounts to offload to the Slave TTPs. The results reveal that the Q_{UC} algorithm outperforms the other algorithms in terms of average achieved reward, at the expense of higher real execution time. The MLLL algorithm ranks second and performs better than the BLLL algorithm when using the same learning parameter β – with respect to the achieved reward and execution time. In contrast, the LRI algorithm has similar rewards but with significantly lower real execution times. To determine the trade-off between the average reward and the execution time, we calculate the ratio of the normalized average reward over the normalized real execution time. The results show that the MLLL algorithm with a low learning parameter value represents the best trade-off due to the significantly lower real execution time while achieving similar rewards when compared to the other algorithms. The superiority of the

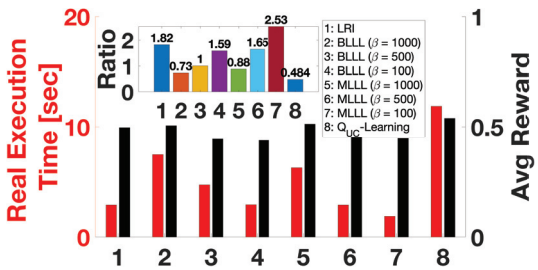

Fig. 5: TTP_M offloading - Comparative Evaluation

MLLL algorithm is further supported by its intrinsic characteristic to converge to the best equilibrium.

D. Transaction Time Speedup with Multiple TTPs

The performance benefit associated with introducing multiple TTPs over the one TTP protocol proposed in [10] are reported on here. Only the MLLL algorithm from the previous section is used in the hardware validation experiments to measure the speed-up, but the performance of the other algorithms can be extrapolated from the results reported above. Our testbed is composed of 5 TTPs and 9 customers. The customers were configured to continuously carry out the blinding process, which effectively emulates a much larger set of customers. The average transaction time, measured from the time instant when a customer selects a Master TTP to the time instant when the customer receives the blinded tokens from the Master TTP improves from approximately 9 seconds with one TTP to 3.7 seconds with 5 TTPs, i.e., a speedup of approximately 2.43.

VI. Conclusion

In this paper, multiple trusted-third-parties (mTTPs) are utilized within the PUF-Cash protocol for improving performance and privacy. A set of artificial intelligence (AI) algorithms are evaluated to determine their effectiveness in selecting an optimal subset of Slave TTPs for offloading the task of exchanging issued e-Cash tokens for blinded tokens during value transfer operations requested by Alice. The combination of AI algorithms and mTTPs provides a significant performance benefit to PUF-Cash, while simultaneously reducing the Bank's ability to correlate Alice's issued tokens with her corresponding blinded tokens. The latter property increases the level of privacy for Alice,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCE.2020.3024512, IEEE Consumer Electronics Magazine

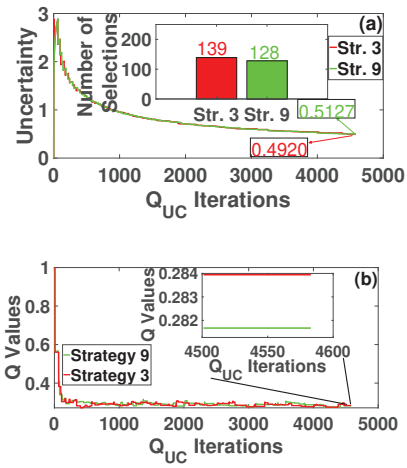

Fig. 6: Q_{UC}-Learning – Evaluation

making PUF-Cash an attractive solution for implementing e-Cash systems that are private and scale with system demand.

ACKNOWLEDGMENT

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

REFERENCES

[1] C. U. Mohanty, Saraju P. and E. Kougianos, "Everything you wanted to know about smart cities, the internet of things is the backbone," *Consumer Electronics Magazine*, vol. 5, no. 3, pp. 60–70, July 2016.

[2] S. Rathore and J. H. Park, "Cognitive science-based security framework in consumer electronics," *Consumer Electronics Mag.*, vol. 9, no. 1, pp. 83–87, Jan/Feb 2020.

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

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

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

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

[7] 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.

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

[9] S. Aaronson and P. Christiano, "Quantum money from hidden subspaces," in *ACM Symp. on Th. of Comp.*, 2012, pp. 41–60.

[10] 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, p. 21, 2019.

[11] G. Fragkos, C. Minwalla, J. Plusquellic, and E. E. Tsiropoulou, "Reinforcement learning toward decision-making for multiple trusted-third-parties in puf-cash," in *IEEE WF-IoT*, 2020.

[12] W. Che, M. Martin, G. Pocklassery, V. K. Kajuluri, F. Saqib, and J. Plusquellic, "A privacy-preserving, mutual PUF-based authentication protocol," *Cryptography*, vol. 1, no. 1, 2016.

[13] R. S. Sutton, A. G. Barto *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135.

Georgios Fragkos (gfragkos@unm.edu) is a Ph.D. student at the Department of Electrical and Computer Engineering, University of New Mexico. He received his Diploma in Electrical and Computer Engineering from National Technical University of Athens in 2018. His main research interests include reinforcement learning and game theory. He received the IEEE Outstanding Graduate Engineering Student Award in 2020.

Cyrus Minwalla (cminwalla@bankofcanada.ca) is a Technical Researcher and Security Lead within the Financial Technology group at the Bank of Canada. He holds a PhD in Computer Engineering from York University and received the NRC's Top Scientist Under 40 Award in 2017. His research interests include digital currencies, cryptography, embedded devices and Internet-of-Things (IoT).

Jim Plusquellic (jimp@ece.unm.edu) is a Professor in Electrical and Computer Engineering, University of New Mexico. His research interests include security and trust in embedded systems, supply chain and IoT. He received an "Outstanding Contribution Award" from IEEE Computer Society for "Co-Founder of and providing Outstanding Contributions to the IEEE International Symposium on Hardware-Oriented Security and Trust (HOST) for the Past Ten Years 2008-2017".

Eirini Eleni Tsiropoulou (eirini@unm.edu) is an Assistant Professor at the Department of Electrical and Computer Engineering, University of New Mexico. Her research interests include reinforcement learning, game theory, and network economics. Four of her papers received the Best Paper Award at IEEE WCNC 2012, ADHOCNETS 2015, IEEE WMNC 2019, and INFOCOM 2019 by IEEE ComSoc Technical Committee on Communications Systems Integration and Modeling.