Fun and games with shared multi-party entropy: the NIST casino

Philip Daian
daian2@illinois.edu

***Abstract*—In this investigation, we will examine the recent introduction of a shared timestamped source of multi-party entropy, the NIST beacon. We will explore the theory behind and implementation of the beacon, and investigate potential applications and security vulnerabilities. We will briefly discuss alternatives to the beacon, and present protocols for and an implementation of two simple and verifiably honest multi-party games of chance using the NIST beacon.**

I. Meet the NIST Beacon

The NIST beacon was introduced by the National Institute for Standards and Technology, a non-regulatory subdivision of the US Department of Commerce. It was intended to provide the first source of public, trusted, timestamped, and historied source of entropy for a varying set of cryptographic applications [1]. As the NIST beacon was being developed, many crytpographic primitives and protocols could benefit from the functionality introduced by a "beacon", broadcasting fresh randomness from a mutually trusted source [9]. Specifically, the beacon is intended to act as a trusted third party, enforcing consensus on some ground truth (the shared entropy) and allowing computational results deriving from that result to share this inherent trust [2]. In doing so, NIST is leveraging its position as a worldwide trusted standards organization to allow for the simplification of and increased transparency into complex multi-party transactions involving some level of third party trust.

The NIST beacon will source its entropy from quantum mechanics, specifically using a loophole-free quantum bell test. This test was specifically designed by NIST to be used both independently and as part of the beacon, to generate truly random data by leveraging quantum properties in ways impossible in a classical mechanics framework. The Bell Test uses the speed of light and the entanglement of two photons to share a secret as follows: two parties, Alice and Bob have separate photon detectors which receive the entangled photons, selecting the polarization of their measurement and thus affecting through quantum mechanics the photon's state. Both choices affect the final detected measurement, and because the choices are made so close to the measurement, it is impossible for the two detectors to share information about their choices even if the information travels at the speed of light. When honestly measured, the result can be interpreted as a truly random measurement [2]. Other similar efforts have used photon and quantum properties to generate high-speed truly random number sources, proving the usefulness of this new understanding of physics in crytographic applications and allowing for the field to transcend traditional pseudo-random generation that was in fact deterministic [3].

While these are the final goals of the project, due to the relative immaturity and non-availability of such entropy sources the beacon is currently using two commercially available and independent entropy sources of an unspecified nature conforming to classical RNG standards [1].

The NIST beacon uses this model for random number generation in a trusted setting entirely controlled by NIST. Broadcasting a full 512-bit entropy block every sixty seconds, each random number is timestamped and signed by NIST to prevent forgery. Additionally, the hash of previous blocks is included. This is intended to prevent modification of previous blocks by NIST, as modifying one block will require the modification of all subsequent blocks assuming collision resistant hash function properties. This system is designed to ensure that each generated value is unpredictable before generation through on-the-fly generation and the quantum properties leveraged, which are independent of other physical processes. Furthermore, a chain of blocks can be used as a proof of random number generation that can be verified offline modulo trust in NIST, checking that the hashes in the chain are valid, the blocks occur every sixty seconds, and all blocks are signed with NIST's trusted key [1]. NIST stores full historical data, allowing the future access of any block since the beginning of the

entropy generation process.

The NIST beacon is served through a simple RESTful API, accessible over TLS from anywhere on the Internet [1]. Wrappers for the beacon's API are available natively for a variety of languages, including Python and Javascript [21] [22].

More formally, the NIST beacon generates a sequence of random numbers and timestamps (R, T) such that R is unpredictable, T is accurate, and past history is protected from modification by a hash-chain using a collision resistant hash function [13].

NIST is hopeful that their beacon will enable a great deal of potential protocols that have previously been too complex to practically implement.

II. Applications of the NIST Beacon

While the NIST beacon is a relatively simple concept, its interest and novelty come from the wide variety of possible applications. The chief aim of this work is to explore these applications, and the practical implications of the use of the NIST beacon for the suggested purposes. Each of these applications stems from the timestamped source of entropy, allowing for trust that a random seed for computation has not been precomputed or adversarially selected.

One of the most unique and interesting examples provided by the authors of the beacon is the use of the beacon in random sampling. The authors propose the use of trusted shared entropy in the selection of which samples to test in manufacturing to assure quality, which industrial sites to investigate for environmental violations, and which shipments into a country to investigate for possibly illegal activity [4]. In each of these applications, there is potential for including bias in the selection of the sample, leading to a result that is unusually favorable to certain parties. By introducing a trusted source of entropy that has not been precomputed it is trivial to introduce a publicly auditable and verifiable system for fairly and randomly conducting sampling in virtually any domain. The applications for this auditable sampling technique extend to industry, medicine, and beyond. A proposal for future medical research could for example include a future timestamp and sampling function based on the NIST beacon with its list of participants, dividing them into experimental groups that can then be audited after the fact for true randomness. If the source of the entropy can indeed be trusted, this has the potential to reduce sampling error or bias in critical areas. Selection of jurors is also mentioned by NIST in their early presentations on the beacon [9].

Another application of the NIST beacon originally suggested by its authors is for use in next generation authentication methods. A common feature of authentication methods, especially those that verify via zero-knowledge proofs of knowledge, is a challenge-response protocol requiring the sharing of a random number. Previously, these protocols required multiple rounds. With the NIST beacon's trusted entropy source and its associated proof of randomness, these protocols become simple and able to be performed online in a single round [5].

The last application of the beacon initially proposed officially by NIST is the application of the beacon to secure multi-party computation. [6] This trusted random broadcast is likely suitable for use as a common reference string for zero-knowledge proofs, allowing for a trusted common reference string that is commonly used by non-interactive zero knowledge proofs to justify knowledge without requiring protocol arounds [16]. This application of the beacon the closest to the applications we will focus on the most in our original work, using the NIST beacon as a trusted source of entropy for deterministic functions sampling a random string (in this case, for gambling). We will not be concerned about maintaining zero-knowledge or privacy preserving properties, rather focusing on fair and verifiable computation when inputs are known to all participants (as in a lottery, card game, etc).

NIST also informally proposed several novel applications of the beacon in a presentation immediately preceding its release. Of particular interest to the presenter was the ability of the beacon to be used in certifying that (0, 0) is not among a set of bit commitments, discouraging adversarial commitments [9]. Also proposed are other anchoring related functionalities, used to justify that information was generated after a certain date.

One of the few actively researched proposed applications by academics outside of NIST is the use of the beacon as a timestamping server to augment the timestamping capabilities of the Bitcoin network, providing security from selfish miners who may solve a block and delay its release to the network by requiring the NIST public entropy source and timestamp to be part of the hashed Bitcoin block [13]. This proposal would decrease the potential for an attack involving a Bitcoin miner withholding blocks and releasing them at a later date to take advantage of an altered payout

schedule. Another proposed application outside of the NIST community includes geohashing, an active amateur community which involves storing objects at locations determined by hashing timestamped information sources, such as market averages [23]. The NIST beacon could potentially replace these timestamping sources, providing something easily machine checkable and globally consistent ideal as a source of entropy for randomly distributing objects in physical space.

The reaction of the amateur cryptographer community on sites like reddit and stackoverflow to the announcement of the NIST beacon was initially one of skepticism, questioning the usefulness of such a third-party source in most cryptographic contexts but acknowledging its potential usefulness as a timestamping server or experimental sampling seed in contexts where trust in NIST was acceptable [7]. Overall, practical implementations of NIST beacon applications remain extremely sparse, likely due to the relatively recent release of the beacon.

III. NIST Beacon Vulnerabilities

An obvious requirement for the NIST beacon if it is to be used a trusted timestamping server is the integrity and consistency of the values it provides. One of the most obvious security issues in the NIST beacon is its existence as a trusted third party. In a cryptographic context, the existence of a trusted third party is often considered a security risk in and of itself, with some researchers claiming that "[w]hen a protocol designer invokes or assumes a TTP, (s)he is creating the need for a novel organization to try to solve an unsolved security problem via traditional security and control methods. Especially in a digital context these methods require continuing high expenditures by the TTP and the TTP creates a bottleneck which imposes continuing high costs and risks on the end user", proceeding to claim that the design of systems that inherently do not require trusted third parties is categorically less vulnerable to manipulation and more secure than systems involving trusted third parties [8].

While this argument is convincing and seemingly devastating to the NIST beacon, it acknowledges the possibility of trusted third party based cryptosystems to provide increased assurance and integrity over legacy systems despite their inherent vulnerability to manipulation. Because the NIST beacon uses a trusted third party by design, these security concerns are not amenable to improvement in a beacon model, but it is likely still possible to gain utility from the use of the NIST beacon over traditional timestamping or shared entropy techniques.

Amateur cryptographers on StackOverflow are also involved in brainstorming risks of the NIST beacon, concluding that an attack in which the NIST pre-generates and withholds beacon blocks or generated blocks without true entropy is possible under NIST's model [7]. These attacks would theoretically be able to influence the output value of the NIST beacon, and thus target any application that uses it in a deterministic fashion. Because this is a relatively severe security hole similar to the trusted third party hole argued by Szabo, it is important to consider the threat model of a particular application before making use of the NIST beacon. Any application vulnerable to targetted manipulation by large-scale intelligence agencies can realistically be targetted through the NIST beacon.

Despite this, NIST argues that it is possible to increase the quality of the entropy from the NIST beacon by performing further computations on it, including bit-flipping or repeated hashing [9]. The widespread use of such techniques would reduce the need for trust in NIST, as manipulations of the beacon could only target a singe application at a time, making them likely not cost effective when compared to the continued honest operation of the beacon.

When using the NIST beacon, it is important to consider the trust placed in NIST in its organizational context. Previous violations of the integrity of NIST services occured when the NIST beacon itself was shut down for several days as a result of the US government budget shutdown, in which many key services were suspended [11]. This downtime exemplifies the social complications introduced to cryptosystems by the need for trust in a third party organization.

Even moreso, previous NIST transgressions have been far less innocuous. Security experts have in the past raised important questions about several NIST standards, and the possibility that these standards were directly weakened for cryptographic applications in order to reveal information to certain parties (in the case of NIST generally the US government and the NSA intelligence agency). One such allegation comes from Bruce Scheiner, who recently publicly called into doubt the selection of constants used in NSA cryptographic ellyptic curves, arguing that their selection did not comply with the procedure expected for random curve selection and

instead used an opaque and ad-hoc process which, without explanation and in the context of poorly understood cryptography, could likely indicate a backdoor [18].

More solidly substantiated allegations include the introduction of serious bugs into the Dual_EC_DBRG PRNG, used to generate private keys. This system, targetted as part of a US federal operation called Bull-Run intended to weaken cryptosystems by, among other things, inserting bugs into software implementations and standards to reduce private key generation entropy in ways predictable by the NSA [19] [20].

These extremely disturbing allegations have been repeatedly substantiated by document leaks, including those leaked by NSA insider Edward Snowden [24]. These allegations have serious security implications in the context of the NIST beacon, which was originally designed to perform a similar service to the known backdoored cryptosystems.

Fortunately, some assurance can be gained by noting that the NSA does not suggest using the Beacon entropy for generating secrets (such as public keys) due to the public accessibility of the Beacon [1]. This significantly reduces incentives to backdoor the beacon on behalf of the government, however, it is important when using such cryptosystems to consider the threat model and trust model available, one intimiately tied with the integrity of the NIST organization itself due to the inherent design of the beacon allowing for manipulable results.

Other implementation concerns have been raised about the NIST beacon, including its original use of HTTP transport security that was outdated and did not provide adequate tamper resistance from third party services. Furthermore, the same user noted that assuming the correctnes of NIST beacon claims requires at the very least trust in "the security of SHA-512, the methods used to measure the unpredictability of quantum behavior, and the methods described in SP800-90A for turning those measurements into random bits", calling SP800-90A into question as the standard which described the selection processes used in the known to be backdoored Dual_EC DBRG syste, described above [7].

IV. Timestamped Random Oracle Lottery

One possible novel application of the NIST beacon is to electronic lotteries. In the past, auditable e-lotteries have struggled from the challenge of generating the entropy required to determine a winner, often resulting to multi-round protocols or deferring trust to a third party to deal with such issues [12] [14] [15].

Previous surveys of several dozen existing electronic lottery schemes have considered several important security criteria for e-lottery schemes: the winning result is (1) randomly generated and (2) publicly verifiable, the total revenue is (3) publicly verifiable, the winning ticket is impossible to forge (4), the user need not be online for the winning value generation (5), player anonymity (6), confidentiality of ticket values (7), fair purchasing of tickets (8), and no early registration required (9) [15]. In analyzing previous schemes under such a criteria, the authors found that no schemes were able to meet all of the requirements without requiring a third-party. Furthermore, the authors propose a scheme based on verifiable random functions, functions taking random input and producing publicly verifiable output and a function that enforces time delays (delaying function) [15].

While the original proposers of this approach stick attempt strictly to avoid introducing third parties, there is a good case to be made for the use of the NIST beacon in such applications. Firstly, the NIST beacon can provide timestamped entropy, guaranteeing modulo trust in NIST that winning values are fairly and unpredictably generated without requiring an additional third party. Secondly, as an organization, NIST is likely relatively indifferent to gambling applications. It is extremely unlikely that a targetted attack on such applications through the NIST beacon would occur, as the resulting cost to NIST's reputations far outweighs any possible dishonest game of chance gains or sabotage. While there is still risk inherent in this deference to a third party that researchers into electronic lotteries are smart to avoid, it is also possible to fairly and simply satisfy many of the above e-lottery criteria through simple applications of the NIST beacon and its related guarantees.

Consider the players having access to a trusted, shared random oracle providing similar functionality to the NIST beacon, providing trusted timestamps and entropy of the form (R, T) resistant to modifications in past history and proved to be random by a zero-knowledge proof (in the practical case the signature of NIST on each block). Each player can now participate in the lottery by generating a ticket T signed by a private key they hold. Sending such a signed transaction to the server, the server can include several of these transactions in a "block". These blocks are groupings of several transactions and

an associated random oracle value/timestamp pair. We enforce the property that these blocks are authenticated by the server using RSA signatures with a trusted master server key. Inclusion of a transaction in such a block is intended to act as a proof of purchase, and thus should only be done after payment settlement. Once included in a block, a user can claim a valid ticket signed by the server and checkpointed by the oracle as proof of their entry into the competition.

While the authors of the scheme requiring no third party use a delaying function to prevent of their tickets, we simply checkpoint our blocks to the random oracle. We issue one block for every new random value and timestamp issued by the shared oracle, checkable for timestamps by all with access to the oracle.

We enforce the property that these blocks are fresh by including data from the random oracle source in the blocks, comprising a timestamped query usable to prove that the blocks are no older than a certain date. We further prevent the modification or removal of old blocks by including the hash of the previous block in each block, a strategy similar to that used by the NIST beacon itself. This imples that any retroactive changes to previous blocks in the chain requires regeneration of all subsequent blocks if a collision is computationally difficult to find.

The parameters of the game (eg - end random oracle time, payout, etc.) are specified as part of the first block, a special block issued by the server to define game settings and like other blocks not modifiable after the fact due to the hash chain built.

The end winner is computed using a determinstic function of the chain, taking as input the parameters in the first blocks and all subsequent blocks, aggregating available tickets, and determining a winner using a common random reference string provided by the oracle. The last reference string provided by the oracle is used for entropy, with all transactions being released in blocks before the unveilling of that value. The chain thus, as in the NIST beacon, provides the full proof of authenticity and purchase for the lottery, showing every ticket sold and able to be verified with the winner computed by all parties posessing a copy of the chain.

In a lottery, the winner is determined by assigning each ticket a unique range of discrete integers based on the value of the ticket. The winning integer is then computed using the total number of covered integers and the shared entropy source, and the ticket whose range covers the winning integer is then the winner of the lottery. This is verifiable by all participants. The winner then claims the ticket by signing a challenge message using the RSA private key originally used to sign his ticket to sign a challenge (potentially also based on the random oracle).

Let us consider the security properties of such a scheme in the context of the desired e-lottery guarantees. Firstly, the winning result is randomly generated. If we do not allow any transactions to be included in blocks released after the timestamping of the final random value provided by the oracle, the winner is determined exclusively by the random oracle output (as the win function is deterministic with all other inputs fixed). Because we will design a scheme where every player is equally likely to win given a random reference string, this means that every player must be equally likely to win given that the provided oracle string is random, implying random winner generation.

The winner is also publicly verifiable. Any user posessing the whole chain can verify the integrity of the chain, giving them assurance that blocks were not retroactively modified by the lottery agent and that adversarial blocks were not preselected before the opening of the lottery. Users are also able to calculate, given the chain, the final random entropy used in the winning computation and thus the winner themselves. This allows any user with whom the programmatic rules of the game are shared to publicly verify the outcome of each chain.

The total revenue is similarly publicly verifiable by virtue of the public visibility of each ticket, the winning ticket is computationally impossible to forge without knowledge of the winning player's RSA key (by the properties of RSA), and the user need not be online for generation of the winning player, which requires only the chain to compute and a value from the NIST beacon to be included.

Fair purchasing is enforced by giving each user a proof of purchase, the inclusion of their transaction in a block. Once their transaction is included, any modifications to alter or exclude their transactions can be refuted by the user, who provides an alternate chain digitally signed by the lottery authority proving their purchase and substantiating any allegations of foul play by holding the lottery authority accountable.

No registration is required to submit transactions to the server, simply an RSA key and appropriate payment settlement.

The only two security properties not entirely guaranteed by our scheme are user anonymity and ticket price hiding. Each ticket in our model bears all the information needed to compute the final game outcome, containing its full dollar value as a result. This is mitigable by simply issuing a separate ticket for each base unit (eg - dollar) spent to a different RSA key, making it impossible for an outside observer to know the number of tickets controlled by any subset of the participatns. Similarly, we do not provide user anonymity, allowing users to identify themselves pseudonmyously using RSA key signatures. This is likely sufficient as a zero-knowledge proof of ownership for a ticket, as the generation of the RSA key is unlinked entirely with the physical identity of its owner.

Thus, a shared timestamping random oracle can be used to greatly simplify protocols useful in online lotteries. In addition to lotteries, we can generalize the above scheme to many other games - each transaction is simply an action in-game that acts as an input to the final outcome of the game. The outcome of the game is determined by a determinstic function of all of these transactions seeded with a common random reference string, providing desirable verifiable entropy and accountability in the generation of the game's final result.

V. Shared Random Oracle Coin Tossing

Assuming a shared random oracle as above, there is also an obvious scheme for coin tossing inherent in this random oracle model. It is trivial to use the shared entropy used in the previous scheme to determine the lottery winners to instead seed an RNG, using the RNG to deterministically output coin toss results to participating users. Similarly to the lottery, knowledge of any of these outputs requires knowledge of the input used to generate them in the PRNG, as randomly seeding the PRNG will produce an output string that appears random by properties of a PRNG.

Thus we apply the NIST beacon similarly to a coin-tossing scheme, allowing users to seed a deterministic PRNG in a fashion that is equivalent to the random and unpredictable output of a random oracle, given the assumptions made about the NIST beacon's entropic unpredictability hold.

VI. Applying the Beacon - A Demonstration

To demonstrate the schemes we have discussed above, a prototype web casino was created allowing users to engage (without financial incentives) in the above games using the NIST beacon as a random oracle backend. This casino is available online at pdaian.com/nistcasino, and allows for fully functional lottery and cointossing games. A user is simply required to provide a ticket request conforming to the given rules and signed by their RSA key, and then to verify that it was properly included in the checkpointed chain of blocks described above. Once the entry is included in the transaction chain, due to the properties of previous hash function pointers, as in the NIST beacon it is impossible for the NIST casino to retroactively modify or remove the entry.

Verification of the chain can then be done offline using the provided tools, checking that the structure of the chain follows certain integrity characteristics (proper hashes of previous blocks, proper block signatures, only valid transactions in each block, etc.) Once the chain is verified, the provided tools can be used to deterministically compute a winner offline, proving the fairness of the result assuming trust in the entropy of the NIST beacon.

This provided for a practical and fun online crypto lottery experience that is provably fair and satisfied a significant portion of our targetted e-lottery security guarantees as described above.

While the security arguments of our scheme follow from a random oracle setting, it is reasonable to assume based on the constructions and claims made by the NIST beacon authors that new values provided by the NIST beacon are both timestamped and unpredictable [1]. These properties together make the NIST beacon sufficient to cover the security assumptions we extracted from the random oracle model. However, organizational and technical trust in NIST and their implementations is still crucial, providing possible practical security holes and attack vectors not existing in models not relying on third parties and introducing inherent security vulnerabilities which are impossible to mitigate.

VII. NIST Beacon Alternatives

As with any system, one of the important considerations when deciding whether to use the NIST beacon in new constructions is the availability and comparability of its alternatives, especially in terms of security guarantees and computational advantages. Two-party and multi-party RNG is a well-established problem featuring some solutions that are mostly multi-round [10]. In such a

scheme, guarantees against both active and possible adversaries are demonstrable. Despite this, such simplistic schemes lack an important property of the NIST beacon, namely its historical data and ability to be verified as proof of fresh entropy.

One of the earliest proposed schemes to serve as such an entropy beacon with no need for a trusted third party leveraged the Bitcoin blockchain to gather entropy, using the low bits of the SHA-256 hashes resulting from Bitcoin transactions. This informal analysis concluding that even a small percentage of network hash power may be problematic in giving an adversary an opportunity to probabilistically select favorable hashes [10].

Yet another approach is the elimination of entropy in applications like non-interactive zero-knowledge proofs. Previous schemes have leveraged discrete log properties and homomorphic encryption to remove the requirement for the random oracle model and the assumptions and implementation challenges it carries [17].

VIII. Conclusions

The NIST beacon is a novel and useful piece of technology that has extensive applications in improving modern systems willing to accept NIST as trusted. Despite some practical security issues and the theoretical promise of third-party free systems, the beacon remains a clever project and a good choice for many.

The creation of a gaming site based on the NIST beacon, while not one of the approved or recommended uses, also presents a novel use case for such a trusted oracle. With relatively simple implementation and some extremely consequential but palatable assumptions, we obtain rigorous assurance in the fairness of a wide variety of games, a marked improvement over today's entirely nontransparent lottery and game systems.

References

[1] Peralata, Rene(2014), *NIST Randomness Beacon*. NIST published online at http://www.nist.gov/itl/csd/ct/nist_beacon.cfm (accessed 12/4/2014)

[2] Bienfang, Joshua(2012), *Truly Random Numbers – But Not by Chance*. NIST published online at http://www.nist.gov/pml/div684/random_numbers_bell_test.cfm (accessed 12/4/2014)

[3] Wayne, Michael and Kwiat, Paul(2010), *Low-bias high-speed quantum random number generator via shaped optical pulses*. Optics Express 18(9):9351-9357

[4] Peralata, Rene(2014), *Unpredictable Sampling*. NIST published online at http://www.nist.gov/itl/csd/ct/beacon-unpredict-sampling.cfm (accessed 12/4/2014)

[5] Peralata, Rene(2014), *New Secure Authentication Mechanisms*. NIST published online at http://www.nist.gov/itl/csd/ct/beacon-new-secure-auth-mechanisms.cfm (accessed 12/4/2014)

[6] Peralata, Rene(2014), *Secure Multi-party Computation*. NIST published online at http://www.nist.gov/itl/csd/ct/beacon-secure-multi-party-computation.cfm (accessed 12/4/2014)

[7] Frame, Richie(2014), *How useful is NIST's Randomness Beacon for cryptographic use?*. via StackExchange, published online at http://crypto.stackexchange.com/a/15231 (accessed 12/4/2014)

[8] Szabo, Nick(2001), *Trusted Third Parties Are Security Holes*. Self published online at http://szabo.best.vwh.net/ttps.html)

[9] Peralata, Rene(2014), *Secure Multi-party Computation*. Presented at RSA Conference 2014, San Francisco, published online at http://www.rsaconference.com/writable/presentations/file_upload/asec-t07b-the-nist-randomness-beacon_final.pdf (accessed 12/4/2014)

[10] Isaacson, Andy(2014), *[cryptography] NIST Randomness Beacon*. via the cryptography@randombit.net mailing list, published online at http://lists.randombit.net/pipermail/cryptography/2013-November/005762.html (accessed 12/3/2014)

[11] Bonneau, Joseph(2013), *NIST randomness beacon off due to the #shutdown*. via Twitter, published online at https://twitter.com/josephbonneau/status/385117096239050752 (accessed 12/4/2014)

[12] Demer, Rickie(2014), *Trustless, Multiparty Random Number Generation*. via StackExchange, published online at http://crypto.stackexchange.com/a/12778 (accessed 12/4/2014)

[13] Heilman, Ethan(2014), *One Weird Trick to Stop Selfish Miners*. International Association for Cryptologic Research, published online at http://crypto.stackexchange.com/a/12778 (accessed 12/4/2014)

[14] Xiao-han, Sun(2014), *Scheme of Instant-open E-lottery Based on Trusted-third-party*. Journal of Weinan Teachers University, published online at http://en.cnki.com.cn/Article_en/CJFDTotal-TXBM200908040.htm (accessed 12/4/2014)

[15] Chow, Sherman S M, Hui, Lucas C K, Yiu, S M, Chow, K P(2010), *An e-Lottery Scheme Using Verifiable Random Function*. Computational Science and Its Applications-ICCSA, published in LNCS 3482:651-660

[16] Haralambiev, Kristiyan(2011), *Efficient Cryptographic Primitives for Non-Interactive Zero-Knowledge Proofs and Applications*. Doctoral dissertation.

[17] Damgrd, Ivan, Fazio, Nelly, and Nicolosi, Antonio (2006), *Non-interactive Zero-Knowledge from Homomorphic Encryption*. Theory of Cryptography, published in LNCS 3876:41-59

[18] Schneier, Bruce(2013), *The NSA Is Breaking Most Encryption on the Internet*. Published online at https://www.schneier.com/blog/archives/2013/09/the_nsa_is_brea.html#c167 (accessed 11/25/2014)

[19] Lochter, M. and Merkle, J.(2010), *Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation*. Internet Engineering Task Force, published as independent submission for review/comment at http://tools.ietf.org/html/rfc5639

[20] Schneier, Bruce(2007), *Did NSA Put a Secret Backdoor in New Encryption Standard?*. Wired Magazine, published online at http://archive.wired.com/politics/security/commentary/securitymatters/2007 (accessed 12/4/2014)

[21] https://pypi.python.org/pypi/randomness$_beacon/0.0.1$

[22] https://github.com/charlescharles/nist-beacon

[23] http://xkcd.com/426/

[24] Ball, James, Borger, Julian, and Greenwald, Glenn (2013). ”US and UK spy agencies defeat privacy and security on the internet”. The Guardian, published online at http://www.theguardian.com/world/2013/sep/05/nsa-gchq-encryption-codes-security (accessed 12/4/2014)