Unique Domain-specific Citizen Identification for E-Government Applications

Peter Schartner
Institute of Applied Informatics
System Security Group
Klagenfurt University
9020 Klagenfurt, Austria
Email: peter.schartner@aau.at

***Abstract*—When discussing the security of e-government applications one of the most crucial aspects is the identification of the users (aka citizens). On the one hand, the authorities and the users want to be sure beyond doubt that a certain action or record is related to the correct individual. On the other hand users do not want to have their actions or data in different domains (like health-care, taxes, register of residents, legal authorities) being linked to each other by the authorities. In this paper, we propose an efficient mechanism, which guarantees both, unique identification and inter-domain privacy protection. First of all, the proposed scheme is a replacement for the domain-specific citizen identifier defined by the Austrian authorities, but the scheme may be used as well in other scenarios, depending on unlinkable and unique identifiers.**

***Index Terms*—e-government; system-wide unique identifier; domain-specific identifier; pseudonyms; anonymity; UUIDs; GUIDs.**

I. Introduction

Concerning e-government, accountability of actions or records is one of the most important requirements. On the one hand, authorities would like to know, which user (citizen) has taken a certain actions or, which user is the owner of a certain record. On the other hand, the users do not want their actions or records being mixed up with actions or records of other users. So both groups need and want accountability, which strongly depends on unique identification of the related instances.

Despite the need for unique identification of citizens, most commonly data protection acts (or similar legal requirements) prohibit the (direct) use of unique identifiers (like passport serial numbers or social insurance numbers) outside the scope of these identifiers. Additionally, the users demand privacy protection, i.e., users do not want their actions (or records) to be linked across different domains. For example, data related to health care should not be linkable to data of social insurance and vice versa. So for both reasons, legal regulations and privacy protection, we need some sort of digital pseudonym, which uniquely identifies a citizen, but hampers the linking across domains.

The remainder of this paper is structured as follows. First we will briefly discuss related work concerning the generation of unlinkable (and unique) identifiers. After analyzing the drawbacks of the different schemes, we introduce the so called concept of collision-free numbers, which are used to generate system-wide unique domain-specific citizen identifiers. The paper will close with some modifications of the proposed scheme and open problems, which are the scope of future research.

II. Related Work

In this section, we will briefly discuss internet/industrial standards and some straightforward techniques for the generation of unlinkable unique identifiers. Besides these, we will discuss the approach of the Austrian authorities in more detail, as flaws in this approach brought up the idea of designing a replacement. Basically, all generation processes described, "try" to provide two properties for the identifiers at the same time:

- **Uniqueness:** No two (or more) citizens should be assigned the same identifier. If this happens, this could result in records or actions of different persons becoming inseparably mixed up.
- **Privacy:** Identifiers used in different domains should not be linkable to each other. In some scenarios even the linking between the person and its identifier should be impossible, which results in complete anonymity. In principle this results in the requirement that identifiers "should look" random.

A. UUIDs and GUIDs

A widely adopted approach for system-wide unique system parameters are **universally unique identifiers** (UUIDs, see [1]) and **globally unique identifiers** (GUIDs, see [2]), Microsoft's implementation of UUIDs). There exist several variants of GUIDs, but these variants either use the MAC address to guarantee uniqueness or they employ hash-functions or purely pseudo-random values. Except the first one, which violates the privacy requirement (the MAC address may be linkable to the user), none of them can guarantee uniqueness (since cryptographic hash-functions always come with the risk of duplicates).

B. National Citizen Identifier

In Austria, each individual is assigned a unique so called base number (B – Basiszahl), which is either the individual's

 ISBN: 978-1-61208-176-2

number in the central register of residents, or B is the number in the so called supplementary register, if the person is not subject to registration. Since the Austrian data protection act prohibits the direct use of the base number B, the derivation scheme for unique unlinkable domain-specific identifiers consists of two major phases (see Figure 1):

1) Disguising the base number B by use of an injective transformation, which results in the so called base identifier (bID).
2) Deriving the domain-specific citizen identifier ($dcID$) by use of the base identifier (bID) and the domain identifier (dID).

Phase 1: Disguising the base number consists of the following steps:

1) Input: base number B (12 decimal digits)
2) Binary encoding of B (5 byte)
3) Extension of B to fill two 3DES blocks (16 byte = 128 bit) by use of the following format:

$$b = B \,||\, seed \,||\, B \,||\, B,$$

where $||$ denotes the concatenation of bit strings and $seed$ is a secret constant (8 bit), only known by the authority, which holds the register of residents.
4) Encryption of the binary representation of b by use of 3DES [3] in CBC mode [4], [5] (no padding needed since the input is a multiple of the block size):

$$c = \text{3DES}_k(b),$$

where the secret key k is only known by the authority, which holds the register of residents.
5) For the ease of further usage, the result is Base64-encoded [6] to form the base identifier:

$$bID = \text{Base64}(c).$$

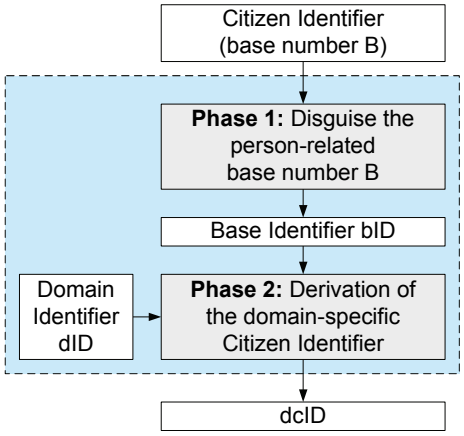

Fig. 1. Original derivation of the domain specific citizen identifier dcID

Analysis: The system-wide unique base number B is encrypted by 3DES (a block cipher) using a fixed key and seed. Hence this is an injective function and the output, the base identifier bDI, is system-wide unique as well. From the security point of view it has to be mentioned, that in case the secret key k becomes publicly known, all base identifiers can be decrypted and actions identified by use of the base identifier can be linked to persons by use of the base number B. Additionally, each individual is assigned exactly one base identifier. Hence, actions or records identified by use of the base identifier may be unlinkable to persons directly, but ar at least linkable to each other. If one of the linked actions or records provides information about its initiator or holder, all other linked actions or data sets can be linked to this specific person.

To overcome the problem of inter-domain linking discussed above, the Austrian authorities proposed to use a derivation scheme, which generates a so called domain-specific citizen identifier ($dcID$) based on the individual's base identifier (bID) and a domain identifier (dID). In order to avoid duplicates, the domain-specific citizen identifiers should be unique with high probability.

Phase 2: The derivation of the domain-specific citizen identifier $dcID$ from the base identifier bID and the domain identifier dID consists of the following steps:

1) Input: Base identifier bID (Base64-encoded) and domain identifier dID (according to the corresponding regulation [7] two to fife ISO/IEC 8859-1 [8] upper case characters)
2) Concatenation ($||$) of base identifier bID, a fixed prefix and the domain identifier dID to form the string s:

$$s = (bID \,||\, \text{“}+\text{”} \,||\, URN - prefix \,||\, dID),$$

where $URN - prefix$ is the ISO/IEC 8859-1 string “urn:publicid:gv.at:cdid+”.
3) Calculation of the SHA-1 hash [9] of s, which results in a 160 bit value h:

$$h = \text{SHA-1}(s)$$

4) Finally, h (as a binary string) may be directly used as domain-specific citizen identifier $dcID$ or may be Base64-encoded before transmission or printout.

Analysis: Since domain-specific citizen identifier are derived by the use of a hash-function, there is the risk of duplicates regardless the fact, that the input to the hash-functions are system-wide unique. Hence there is the risk of inseparable records of different individuals e.g., in E-Government databases.

C. Other Approaches

There exist at least three straight forward solutions for generating random *and* system-wide unique parameters:

- **Centralized generation and check** obviously avoids duplicates but is quite inefficient concerning storage (all previously generated parameters have to be stored for later comparison) and communication (each instance, which needs a parameter has to wait for the centralized generator to send it). Additionally, the centralized generator has full control over the generating process and knows all parameters.

 ISBN: 978-1-61208-176-2

- With **Local generation and (centralized) check**, only the generation itself is done locally, but the comparison against all previously generated parameters has to involve all other generators or a centralized service. Again, efficiency and security are quite questionable.
- **Local generation based on pseudo-random number generators** (PRNG, see [10] for details) can avoid centralized storage and comparison and is efficient in terms of memory and communications. But in order to avoid duplicates, all PRNGs have to use a common key or common secret parameters. So, if one of them is compromised, all of them become insecure. Additionally, the generated parameters are no longer random, but pseudo-random and this approach is not suitable for software implementation, because by use of software, the system-wide key (or secret parameter) cannot be protected sufficiently.

A more sophisticated approach is the so called **location- and time-based generation**, which simply uses location and time provided by a GPS receiver to derive a unique seed for the generation process. The idea behind this concept: two generation processes cannot take place at the same place *and* the same time. Besides the fact that the GPS signal will not be available at all locations, the according paper does not specify, how (pseudo-) randomness and uniqueness are maintained (see [11] for details).

D. Summary of Related Work

Summarizing the related work, we see that none of them fulfills both requirements at the same time: system-wide uniqueness and privacy protection (full or inter-domain unlinkability).

III. Preliminaries

After briefly revisiting basic cryptographic algorithms used in this paper, we will present the core building block of unique domain-specific identifiers: so called collision free number generators (CFNG, introduced in [12], [13]).

A. Cryptography

We assume that the reader is familiar with **Symmetric Encryption** (like DES [14], 3DES [3], or AES [15]) and **Hash-functions** (SHA-1 [9] or RIPEMD160 [16]), and refer to [10] for further details.

In order to keep the output of symmetric encryption as short as possible, we will employ **Ciphertext Stealing**. Let l_B be the block-length of a symmetric encryption function E. Let u be a plaintext, where $l_B < l_u \leq 2l_B$. If u is encrypted straightforwardly by padding u up to $2l_B$ bits and then encrypting two blocks, the length of the corresponding ciphertext c is $l_c = 2l_B$. Using the CBC mode [4], [5] with ciphertext stealing [17], c can be generated such that $l_c = l_u$. This works as follows: First u is cut into the blocks u_1 and u_2, where $l_{u_1} = l_B$ and $l_{u_2} = l_u - l_B$. Then u_1 is encrypted by use of E and a properly chosen key k resulting in a block $c_1||c_2$, where $l_{c_1} = l_u - l_B$ and $l_{c_2} = l_B - l_{c_1}$. Then the block $c_2||u_2$ is encrypted by use of E and the same key k resulting in the block c_3. This works, since $l_{c_2} + l_{u_2} = l_B$. The ciphertext of u is then $c_1||c_3$ and contains sufficient information to compute u, if k is available. The length of c is $l_c = l_{c_1} + l_{c_3} = l_u$. An example for a 64 bit block cipher (like DES) encrypting an 79 bit input can be found in Figure 2.

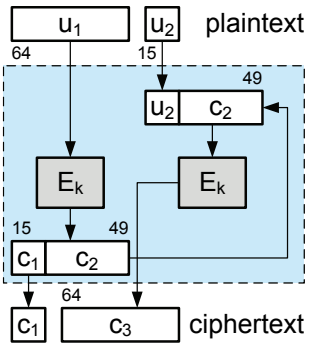

Fig. 2. CBC mode with ciphertext stealing

Details on **Elliptic Curve Cryptography (ECC)** can be found in [18]). For the ease of reading this paper we will just define the basics of ECC.

Definition: Let $E(\mathbb{Z}_p)$ be an elliptic curve group, where p is an odd prime. Let $P \in E(\mathbb{Z}_p)$ be a point of prime order q, where $q|\#E(\mathbb{Z}_p)$. The *Elliptic Curve Discrete Logarithm Problem (ECDLP)* is the following: Given a (random) point $Q \in \langle P \rangle$ and P, find $k \in \mathbb{Z}_q$ such that $Q = kP$.

By $SM(k, P)$ we henceforth denote the **Scalar Multiplication** kP in $E(\mathbb{Z}_p)$. It is believed that the ECDLP using $l_p \approx l_q \approx 160$ is secure against powerful attacks like Pollard's rho algorithm [18].

Point Compression [19]: A point on an elliptic curve consists of two coordinates and so requires $2l_p$ bits of space. It is clear that for every x-value there exist at most two possible y-values. Since they only differ in the algebraic sign, it suffices to store only one bit instead of the whole y-value. A point (x, y) can hence be stored as $x||b$, where $b = y$ MOD 2, and then only requires $l_p + 1$ bits of space.

This has the only drawback that if we want to include this point in some computations, we first have to compute the two possible y-values and then decide by b, which of them is correct. In our case, we are only interested in saving space. There is no necessity to compute y here.

B. Collision-free Number Generators

In [12], we proposed so called collision-free number generators (CFNGs) as a mechanism for generating random but system-wide unique (cryptographic) parameters. Basically, these generators disguise a unique (eventually publicly known) parameter by use of a randomizer. In the scope of e-government identifiers, the information being disguised will be the digital identity of the citizen. The resulting parameter will be a system-wide unique domain-specific digital pseudonym.

 ISBN: 978-1-61208-176-2

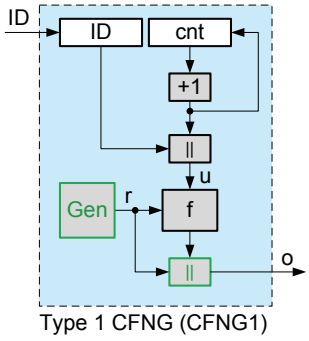

Fig. 3. Basic construction of Collision-free Number Generators (CFNGs)

The output o of a basic – type 1 – CFNG (denoted as CFNG 1 in the remainder of this article, also see Figure 3) is of the form

$$o = f(u, r)||r = f_r(u)||r = \text{CFNG1}(),$$

with f being an injective mixing transformation for an arbitrary but fixed randomizer r and u, r defined as above. We suggest to either use an injective one-way mixing-transformation for f_r according to Shannon [20] (e.g., symmetric encryption) or an injective probabilistic one-way function, based on an intractable problem (e.g., the discrete logarithm problem [10]).

In this paper, we will just revisit the proofs of uniqueness. For a detailed discussion of randomness, efficiency and privacy protection, we refer the reader to [12].

Theorem: *Outputs of Type 1 CFNGs are unique during their lifetime.*

Proof: Consider two outputs of two arbitrary type 1 CFNGs: $o_1 = \text{CFNG1}_1() = f_{r_1}(u_1)||r_1$ and $o_1 = \text{CFNG1}_2() = f_{r_2}(u_2)||r_2$, with r_1, r_2 being random and $u_1 = ID_1||cnt_1$ and $u_2 = ID_2||cnt_2$. With respect to the randomizers r_1 and r_2, there are two cases:

1) $r_1 \neq r_2$: This directly means that $o_1 \neq o_2$.
2) $r_1 = r_2 = r$: Now, both calls of the generators employ the same randomizer and f_r becomes injective. Hence $f_r(u_1)$ and $f_r(u_2)$ will be different if and only if $u_1 = ID_1||cnt_1$ and $u_2 = ID_2||cnt_2$ differ in at least one bit. This is always true, because
   a) different generators use different identifiers ($ID_1 \neq ID_2$), and
   b) if we call the same generator twice (i.e., $ID_1 = ID_2$), the values cnt_1 and cnt_2 will differ, because the counter is incremented at each call of the generator.

   Hence the outputs o_1 and o_2 will be different again.

□

When analyzing CFNGs, which employ a block cipher E (CBC mode with ciphertext stealing) for f ($o = \text{E}_r(ID||cnt)||r = c||r$), it is obvious that the identity of the generator is not protected sufficiently. Everybody who gets hold of an output o can retrieve the identifier ID of the according generator by simply decrypting c by use of r: $ID||cnt = \text{D}_r(c)$.

We will see that this may not be a problem in certain application scenarios; but, in order to guarantee the protection of the generators ID we have either to change our requirements on f, or we have to slightly change the design of CFNGs.

- To provide privacy, f has to be a cryptographic one-way function. Candidates include injective probabilistic one-way functions based on an intractable problem like the (ECC) discrete logarithm problem [10].
- In the case that f is a (bijective) symmetric encryption function, we can employ an additional (injective) one-way-function g to the output or to the randomizer of the original CFNG, which results in the variants depicted in Figure 4 (CFNG 2 and CFNG 3):
  1) The first variant simply hides the output of a type 1 CFNG by use of function g:
  $$o = g(\text{CFNG1}()) = \text{CFNG2}(),$$
  2) The second variant only hides parameter r (which is needed to invert function f) by use of function g:
  $$o = f(u, r)||g(r) = f_r(u)||g(r) = \text{CFNG3}(),$$

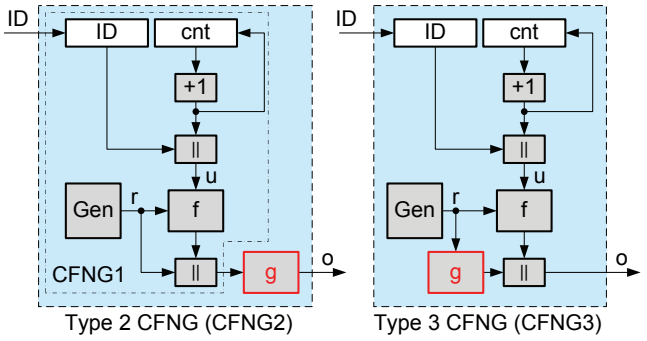

Fig. 4. Variants of Collision-free Number Generators

Corollary: *Outputs of type 2 and type 3 CFNGs are unique during their lifetime.*

Proof: Function g, applied to the unique inputs in type 2 and type 3 CFNGs is a injective one-way-function. Hence g applying g cannot destroy the uniqueness of the outputs.

□

IV. Our Proposal: Unique Domain-Specific Citizen Identifiers

In this section we present three methods to generate unique and unlinkable domain-specific Citizen Identifier:

- **Method 1** (the basic principle) may be directly used as a replacement for the scheme described in Section II-B as it uses the same inputs (and inputs lengths) and generates outputs of equal length.
- **Method 2** uses slightly different (shorter) inputs, but employs more randomness to disguise the inputs. Nevertheless it may also be used as a replacement for the old citizen identifiers.

 ISBN: 978-1-61208-176-2

- **Method 3** uses the base number (60 bit) as the source of uniqueness instead of the base identifier (128 bit) as methods 1 and 2 do. As with method 2, shorter inputs to the encryption function allow more randomness.

A. Basic Principle

Based on a type 2 CFNG employing elliptic curve cryptography (ECC – see [18] for details), elliptic curve scalar multiplication (SM) and point compression (PC) we will now present a generator for system-wide unique and inter-domain unlinkable identifiers. As in the original scheme, our replacement (see Figure 5) generates 160 bit identifiers. But in contrast to the original scheme, these outputs are provably system-wide unique, as we employ type 2 CFNGs (see Figure 4 left) parameterized as follows:

1) Inputs: Base identifier bID (128 bit) and domain identifier dID (five uppercase letters encoded in 24 bit).
2) Starting from the output length of 160 bit we have to subtract one bit to encode the y-coordinate of the ECC-point, 24 bit to encode the domain identifier and 128 bit to store the base identifier. This results in 7 bits remaining for the randomizer.
3) The unique and inter-domain unlinkable identifiers $dcID$ is of the following form:

$$dcID = \text{PC}(\text{SM}((\text{DES}(u, k) \,||\, r), P)),$$

where where P is a so-called generator point of the elliptic curve, $|r| = 7$ and $k = \text{msb}_{56}(H(r))$. In order to reduce redundancy and the bit length of the input of the encryption function, we omit the constant URN-prefix.

Since we employ DES to encrypt the base identifier, we need to expand the randomizer r (7 bit) to 56 bit. This can easily be achieved by use of a hash-function H (e.g., RIPEMD160 [16] or SHA-1 [9]) and a trimming function msb, which extracts the 56 most significant bits: $k = \text{msb}_{56}(\text{H}(r))$. Note that the low entropy of key k is not a severe problem here, because the only purpose of k (based on randomizer r) is to hamper brute force attacks (by a factor of $2^7 = 128$ in this setting).

B. Variant 1

Up to now, the Austrian e-government act [21] and the corresponding domain regulation [7] define just 35 different domain identifiers (see Table I).

So spending 24 bit to store the domain identifier dID is a massive overhead. A more practical solution is reducing the bit length of dID by half (i.e., to 12 bit) and using a binary encoding instead of the text encoding. By this, the length of the randomizer r can be enlarged by 12 bit, which results in $|r| = 19$ bit.

C. Variant 2

This variant directly uses the base number B (5 byte = 40 bit) instead of the base identifier bID (128 bit) and hence shortens the input of the encryption function by 88 bit. We will use some of the bits to embed additional data X, which might hold a counter in order to provide different identifiers within the same domain. The remainder of the bits will be used to enlarge the randomizer to 80 bit and replace DES with SKIPJACK [22] (block length 64 bit and key length 80 bit) in CBC mode with ciphertext stealing. This finally results in unique and unlinkable domain-specific identifiers of the form:

$$dcID = \text{PC}(\text{SM}((\text{SKIPJACK}_r(B \,||\, X \,||\, dID) \,||\, r), P)),$$

with $|B| = 40$ bit, $|X| = 15$ bit, $|dID| = 24$ bit and $|r| = 80$ bit.

Note that the direct use of the base number (which can also be the passport or social insurance number) may be prohibited by law. In this case, variant 1 or the basic scheme have to be used.

TABLE I
CURRENT LIST OF DOMAIN IDENTIFIERS

e-government domain regulation – appendix to § 3 – part 1					
AR	AS	BF	BW	EA	EF
GH	GS	GS-RE	JR	KL	KU
LF	LV	RT	SA	SF	SO
SO-VR	SR-RG	SV	UW	VT	VV
WT	ZP				
e-government domain regulation – appendix to § 3 – part 2					
BR	HR	KI	OI	PV	RD
VS	VS-RG	ZU			

V. CONCLUSION AND FUTURE WORK

We are aware of the fact that the proposed scheme is first of all a replacement for a national standard for generating unlinkable domain-specific identifiers (which does not completely fulfil its own requirements). But nevertheless, provably system-wide unique unlinkable and domain specific identifiers based on collision-free number generators (CFNGs), parameterized as defined in Section IV-A, may be employed in other application scenarios as well. These scenarios include identifiers in the context of e-business, the replacement of UUIDs and GUIDs [12], temporary MACs for untraceable network devices [23], and digital pseudonyms [24], [25].

REFERENCES

[1] P. Leach, M. Mealling, and R. Salz, "RFC 4122 – A Universally Unique IDentifier (UUID) URN Namespace," 2005, (retrieved: 12/2011). [Online]. Available: http://www.ietf.org/rfc/rfc4122.txt
[2] Microsoft Developer Network, "Globally Unique Identifiers (GUIDs)," http://msdn.microsoft.com/en-us/library/cc246025.aspx, 2008, (retrieved: 12/2011).
[3] National Institute of Standards and Technology (NIST), "FIPS Special Publication 800-67: Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher," 2008.
[4] ISO/IEC, "ISO/IEC 10116: Modes of Operation of an n-bit Block Cipher," ISO/IEC, 1991.
[5] National Institute of Standards and Technology (NIST), "FIPS Special Publication 800-38A: Recommendation for Block Cipher Modes of Operation – Methods and Techniques," 2001.
[6] S. Josefsson, "RFC 4648 – The Base16, Base32, and Base64 Data Encodings," 2006, (retrieved: 12/2011). [Online]. Available: http://www.ietf.org/rfc/rfc4648.txt

 ISBN: 978-1-61208-176-2

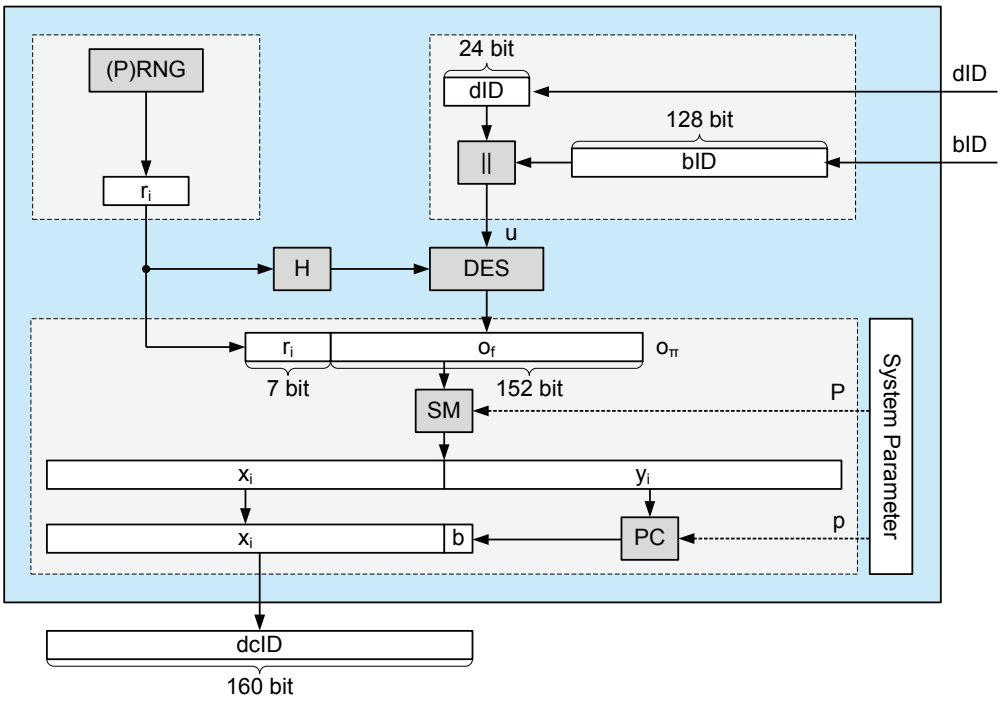

Fig. 5. New implementation of the domain specific citizen identifier $dcID$

[7] Republik Österreich, "Verordnung des Bundeskanzlers, mit der staatliche Tätigkeitsbereiche für Zwecke der Identifikation in E-Government-Kommunikationen abgegrenzt werden (E-Government-Bereichsabgrenzungsverordnung – E-Gov-BerAbgrV) StF: BGBl. II Nr. 289/2004, (Fassung vom 14.9.2011)," 2004, (retrieved: 12/2011). [Online]. Available: http://www.ris.bka.gv.at

[8] ISO/IEC, "ISO/IEC 8859-1:1998, Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1," ISO/IEC, 1998.

[9] National Institute of Standards and Technology (NIST), "FIPS Publication 180-2: Secure Hash Standard," 2002.

[10] A. Menezes, S. Vanstone, and P. V. Oorschot, *Handbook of Applied Cryptography*. CRC Press, Inc., 1996.

[11] IPCOM, "Method of generating unique quasi-random numbers as a function of time and space. PriorArtDatbase, IPCOM#000007118D," 2002, http://priorartdatabase.com/IPCOM/000007118 (retrieved: 12/2011).

[12] M. Schaffer, P. Schartner, and S. Rass, "Universally Unique Identifiers: How To Ensure Uniqueness While Protecting The Issuer's Privacy," in *Security and Management*, S. Aissi and H. Arabnia, Eds. CSREA Press, 2007, pp. 198–204.

[13] P. Schartner, "Random but system-wide unique unlinkable parameters," *JIS – Journal of Information Security*, vol. 3, no. 1, January 2012, ISSN Print: 2153-1234, ISSN Online: 2153-1242, in print. [Online]. Available: http://www.scirp.org/journal/jis

[14] National Institute of Standards and Technology (NIST), "FIPS Publication 46-3: Data Encryption Standard (DES)," 1999.

[15] ——, "FIPS Publication 197 – Advanced Encryption Standard (AES)," 2001.

[16] H. Dobbertin, A. Bosselaers, and B. Preneel, "Ripemd-160: A strengthened version of ripemd," in *Proceedings of Fast Software Encryption (FSE)*, ser. LNCS, D. Gollmann, Ed., vol. 1039. Springer, 1996, pp. 71–82.

[17] C. Meyer and S. Matyas, *Cryptography: A New Dimension in Computer Data Security*. John Wiley & Sons Inc, 1982.

[18] D. Hankerson, A. J. Menezes, and S. A. Vanstone, *Guide to Elliptic Curve Cryptography*. Springer-Verlag, 2004.

[19] IEEE, "Std 1363-2000: IEEE Standard Specifications for Public-Key Cryptography," 2000.

[20] C. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28(4), pp. 656–715, 1949.

[21] Republik Österreich, "Bundesgesetz über Regelungen zur Erleichterung des elektronischen Verkehrs mit öffentlichen Stellen (E-Government-Gesetz – E-GovG), BGBl. I 10/2004, (Fassung vom 3.3.2011)," 2010, (retrieved: 12/2011). [Online]. Available: http://www.ris.bka.gv.at

[22] National Institute of Standards and Technology (NIST), "SKIPJACK and KEA Algorithm Specifications, ver. 2, 29," 1998.

[23] M. Schaffer and P. Schartner, "Untraceable Network Devices," Klagenfurt University (Austria) – System Security Group (syssec), Tech. Rep. TR-syssec-06-04, November 2006.

[24] P. Schartner and M. Schaffer, "Unique User-Generated Digital Pseudonyms," in *MMM-ACNS*, ser. Lecture Notes in Computer Science, V. Gorodetsky, I. Kotenko, and V. Skormin, Eds., vol. 3685. Springer, 2005, pp. 194–205.

[25] ——, "Efficient privacy-enhancing techniques for medical databases," in *BIOSTEC (Selected Papers)*, ser. Communications in Computer and Information Science, A.L.N.Fred, J. Filipe, and H. Gamboa, Eds., vol. 25. Springer, 2008, pp. 467–478.

 ISBN: 978-1-61208-176-2