Divisible Electronic Cash

Special Topics in Cryptography ITC8290

Urmas Pitsi, TalTech December 2020

Plan of the talk: to dissect the following paper:

1. T. Okamoto, “An efficient divisible electronic cash scheme," Annual International Cryptology Conference - CRYPTO 1995: Advances in Cryptology - CRYPTO ‘95, pages 438-451, Springer-Verlag, New York, 1995.

But, in order to do that we need also look deeply into that paper:

2. Okamoto, T., and Ohta, K., “Universal Electronic Cash”, Proceedings of Crypto 91, pp. 324–337 (1992).

[2] is kind of integral part of [1]: includes necessary definitions.

Summary on one slide

1. Efficient, practical, secure single-term divisible electronic cash
2. Satisfies 4 security requirements under some assumptions: (1) No forging, (2) No tracing, (3) No overspending, (4) No swindling. Assumptions: RSA, Factoring, Diffie-Hellmann, Random functions.
3. Coin as Binary Tree representation: nodes representing certain denominations. The value of a coin equals Root node value equals to the sum of leave values.
4. “Coin” tree must follow 2 Rules:
   (i) Route node rule: no descendant and ancestor nodes allowed
   (ii) Same node rule: no node can be used twice
5. Bit Commitment Schemes to realize the *single-term* property (based on discrete log problem)
6. Customer U obtains an *electronic licence* from the bank B through a one-time *opening protocol* using a *bit commitment scheme*. This grants customer U permission to use electronic cash of bank B.
7. Protocols: Payment, Withdrawal, Deposit (same as Payment instruction to bank B)
   (i) Payment: coin authentication, denomination revelation
   (ii) Detection of Overspending: if any of the 2 rules are violated, then the identity of the user can be revealed. Otherwise impossible.
8. Williams integers and Quadratic Residues as number theoretic building blocks. Discrete Log Problem as the main underlying unsolvable problem.

Ideal cash system [Okamoto, Ohta, 1992][2]:

1. INDEPENDENCE: The security of electronic cash cannot depend on any physical condition. Then the cash can be transfered through networks.

2. SECURITY: The ability to copy (reuse) and forge the cash must be prevented.

3. PRIVACY (UNTRACEABILITY): The privacy of the user should be protected. That is, the relationship between the user and his purchases must be untraceable by anyone.

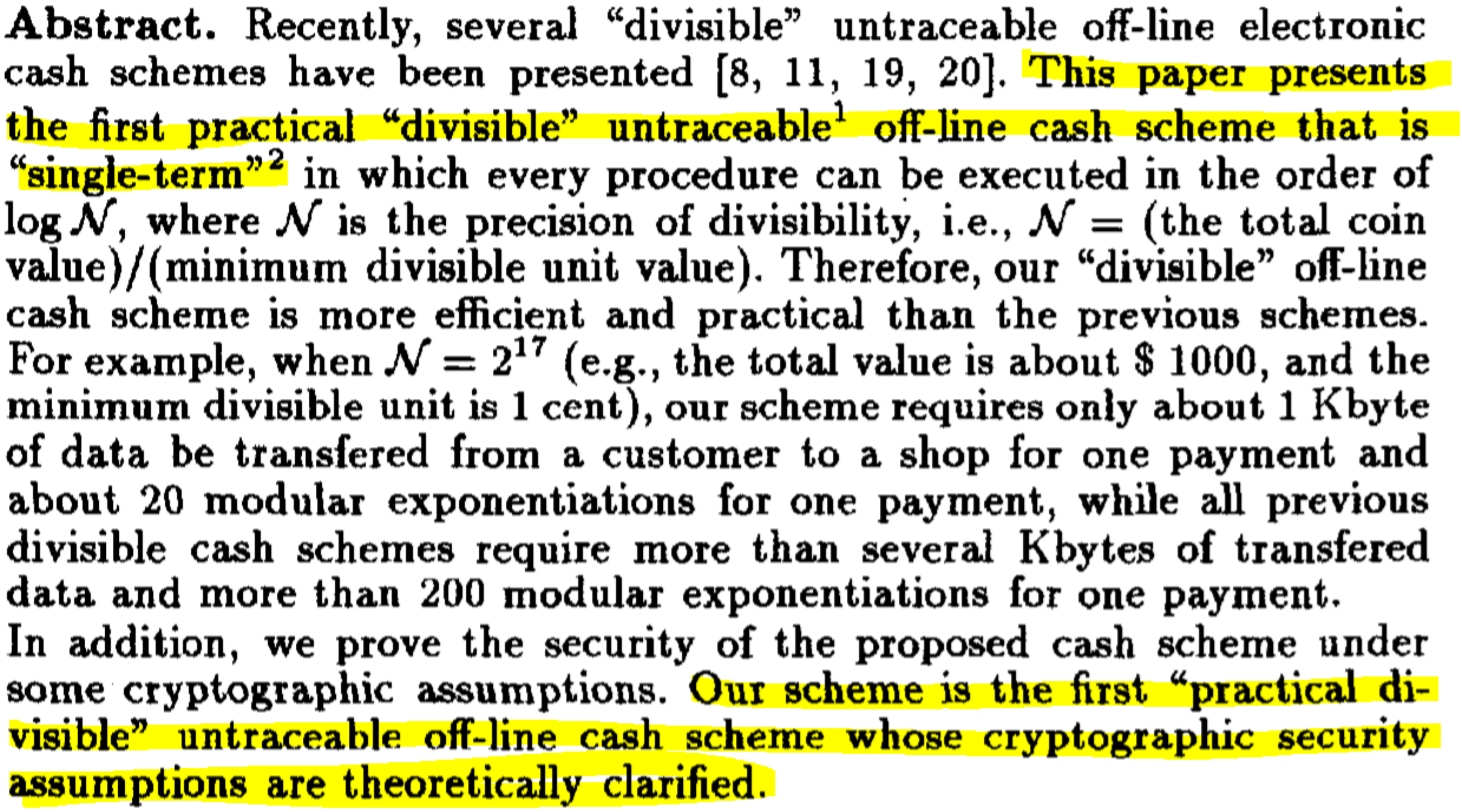
4. OFF-LINE PAYMENT: When a user pay the elcctronic cash to a shop, the procedure between the user and the shop should be executed in an off-line manner. That is, the shop does not need to be linked to the host in user's payment procedure.

5. TRANSFERABILITY: The cash can be transfered to other users.

6. DIVIDABILITY: One issued piece of cash worth value *C* (dollars) can be subdivided into many pieces such that each subdivided piece is worth any desired values less than *C* and the total value of all pieces is equivalent to C.

* dividability is apparently an uncommon way of saying divisibility.

Tatsuaki Okamoto, 1995: “An Efficient Divisible Electronic Cash Scheme”



Abstract. Recently, several “divisible” untraceable off-line electronic cash schemes have been presented [8, 11, 19, 20]. This paper presents the first practical “divisible” untraceable[1] off-line cash scheme that is “single-term”[2] in which every procedure can be executed in the order of $\log \mathcal{N}$, where $\mathcal{N}$ is the precision of divisibility, i.e., $\mathcal{N} =$ (the total coin value)/(minimum divisible unit value). Therefore, our “divisible” off-line cash scheme is more efficient and practical than the previous schemes. For example, when $\mathcal{N} = 2^{17}$ (e.g., the total value is about $ 1000, and the minimum divisible unit is 1 cent), our scheme requires only about 1 Kbyte of data be transfered from a customer to a shop for one payment and about 20 modular exponentiations for one payment, while all previous divisible cash schemes require more than several Kbytes of transfered data and more than 200 modular exponentiations for one payment.
In addition, we prove the security of the proposed cash scheme under some cryptographic assumptions. Our scheme is the first “practical divisible” untraceable off-line cash scheme whose cryptographic security assumptions are theoretically clarified.

1. “untraceable” = anonymous: transactions cannot be linked to users’ identities. However coins divided from the same coin can be linked: this scheme is “linkable”.
2. “single-term” : a practical cash scheme in which the cut-and-choose method is not used and cash consists of a single term (as described in [3], [4]).

Table of Contents of the paper:

Tatsuaki Okamoto, 1995: “An Efficient Divisible Electronic Cash Scheme”

1. Introduction
2. Number Theoretic Conventions (Lemma 1, Lemma 2)
3. Binary Tree Approach (coin data structure, root/same node rules, Γ, Λ values)
4. Bit Commitment Schemes (basic protocols, basis for electronic license)
5. Efficient Divisible Cash Scheme (protocols: opening, payment, withdrawal)
6. Security (incl. assumptions RSA factoring, discrete log problem)
7. Efficiency
8. Conclusion

1. Introduction

- A “divisible” coin worth some amount of money, say $x, is a coin that can be spent many times as long as the sum total of all its the transactions does not exceed $x.

- If a coin is not divisible, the customer must withdraw a coin whenever he spends it, or withdraw many coins of various values and store them in his electronic wallet.

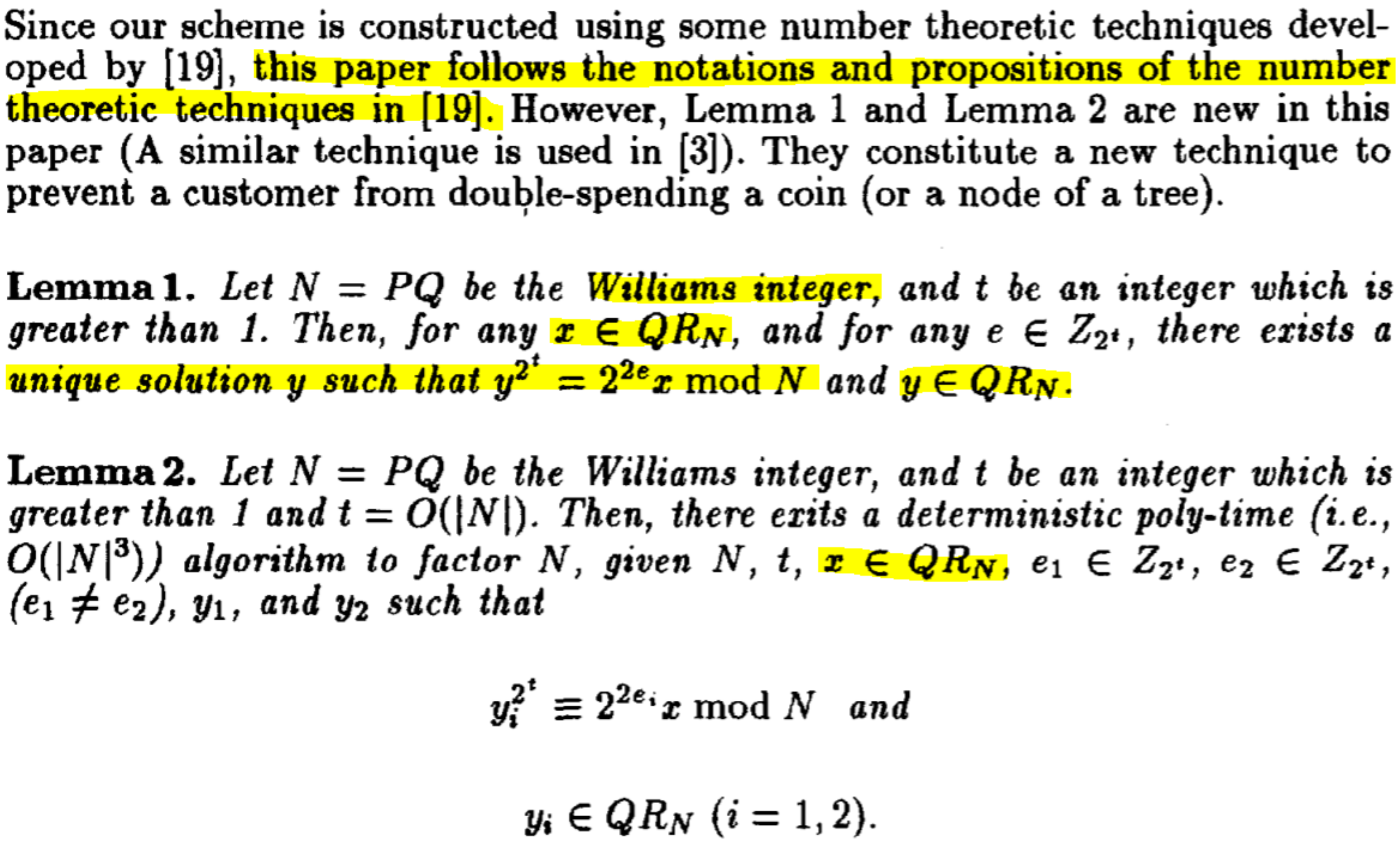
- Real cash is not “divisible”: we must use many various bills and coins in daily life.

- Prepaid cards are “divisible” and this is the major merit of such cards over real cash.

- This paper presents a divisible untraceable efficient off-line electronic cash scheme.

- Our scheme is the first practical “single-term divisible” cash scheme in which every procedure can be executed in the order of log(N), and every transfered data sizes are of the order of log(N). The amount of the required computation and communication for a payment is much less compared with previous schemes.

2. Number Theoretic Conventions



Since our scheme is constructed using some number theoretic techniques developed by [19], this paper follows the notations and propositions of the number theoretic techniques in [19]. However, Lemma 1 and Lemma 2 are new in this paper (A similar technique is used in [3]). They constitute a new technique to prevent a customer from double-spending a coin (or a node of a tree).

Lemma 1. *Let $N = PQ$ be the Williams integer, and t be an integer which is greater than 1. Then, for any $x \in QR_N$, and for any $e \in Z_{2^t}$, there exists a unique solution y such that $y^{2^t} = 2^{2e}x \bmod N$ and $y \in QR_N$.*

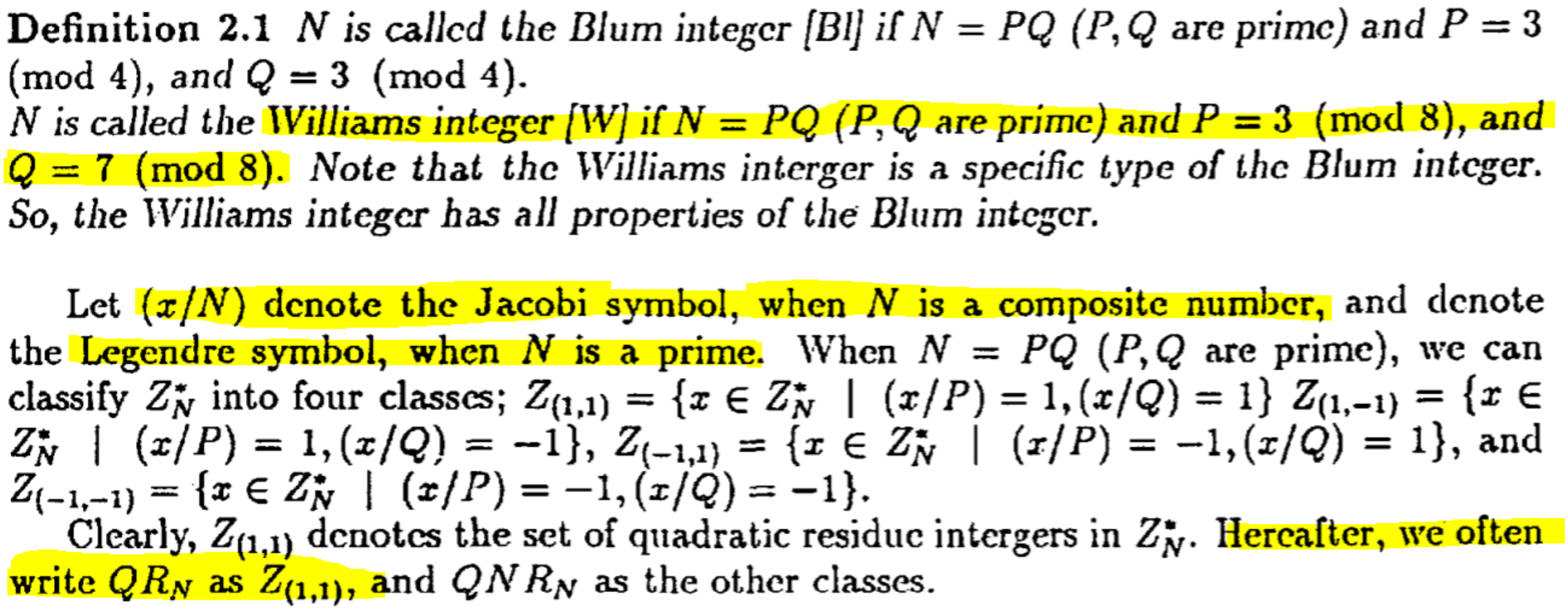
Lemma 2. *Let $N = PQ$ be the Williams integer, and t be an integer which is greater than 1 and $t = O(|N|)$. Then, there exits a deterministic poly-time (i.e., $O(|N|^3)$) algorithm to factor N, given N, t, $x \in QR_N$, $e_1 \in Z_{2^t}$, $e_2 \in Z_{2^t}$, $(e_1 \neq e_2)$, y_1, and y_2 such that*

$$y_i^{2^t} \equiv 2^{2e_i}x \bmod N \quad \textit{and}$$

$$y_i \in QR_N \ (i = 1, 2).$$

Reference [19] above is in this presentation: [2] Okamoto, Ohta 1992 “Universal Electronic Cash”:
Williams integer: N = P*Q (P, Q are prime) and P = 3 (mod 8) and Q = 7 (mod 8).
QRN : quadratic residue (mod N)

2. Number theoretic conventions continued, page 2: conventions from [2]:

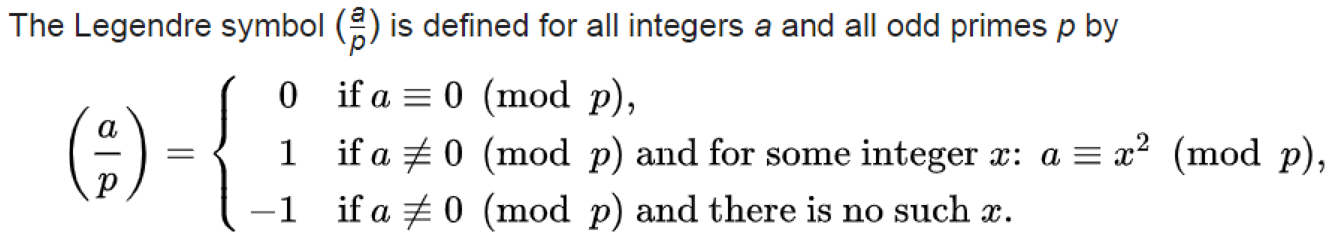


Definition 2.1 *N is called the Blum integer [Bl] if $N = PQ$ (P, Q are prime) and $P = 3 \pmod 4$, and $Q = 3 \pmod 4$.*
N is called the Williams integer [W] if $N = PQ$ (P, Q are prime) and $P = 3 \pmod 8$, and $Q = 7 \pmod 8$. Note that the Williams interger is a specific type of the Blum integer. So, the Williams integer has all properties of the Blum integer.

Let (x/N) denote the Jacobi symbol, when N is a composite number, and denote the Legendre symbol, when N is a prime. When $N = PQ$ (P, Q are prime), we can classify Z_N^* into four classes; $Z_{(1,1)} = \{x \in Z_N^* \mid (x/P) = 1, (x/Q) = 1\}$ $Z_{(1,-1)} = \{x \in Z_N^* \mid (x/P) = 1, (x/Q) = -1\}$, $Z_{(-1,1)} = \{x \in Z_N^* \mid (x/P) = -1, (x/Q) = 1\}$, and $Z_{(-1,-1)} = \{x \in Z_N^* \mid (x/P) = -1, (x/Q) = -1\}$.

Clearly, $Z_{(1,1)}$ denotes the set of quadratic residue intergers in Z_N^*. Hereafter, we often write QR_N as $Z_{(1,1)}$, and QNR_N as the other classes.

Jacobi / Legendre symbols [6]:

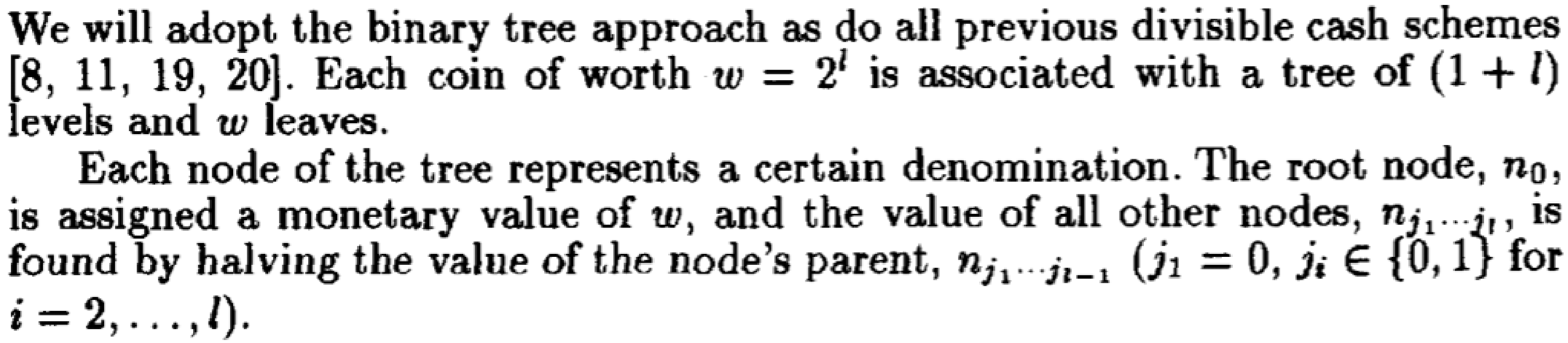


The Legendre symbol $(\frac{a}{p})$ is defined for all integers *a* and all odd primes *p* by

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{if } a \equiv 0 \pmod{p}, \\ 1 & \text{if } a \not\equiv 0 \pmod{p} \text{ and for some integer } x\text{: } a \equiv x^2 \pmod{p}, \\ -1 & \text{if } a \not\equiv 0 \pmod{p} \text{ and there is no such } x. \end{cases}$$

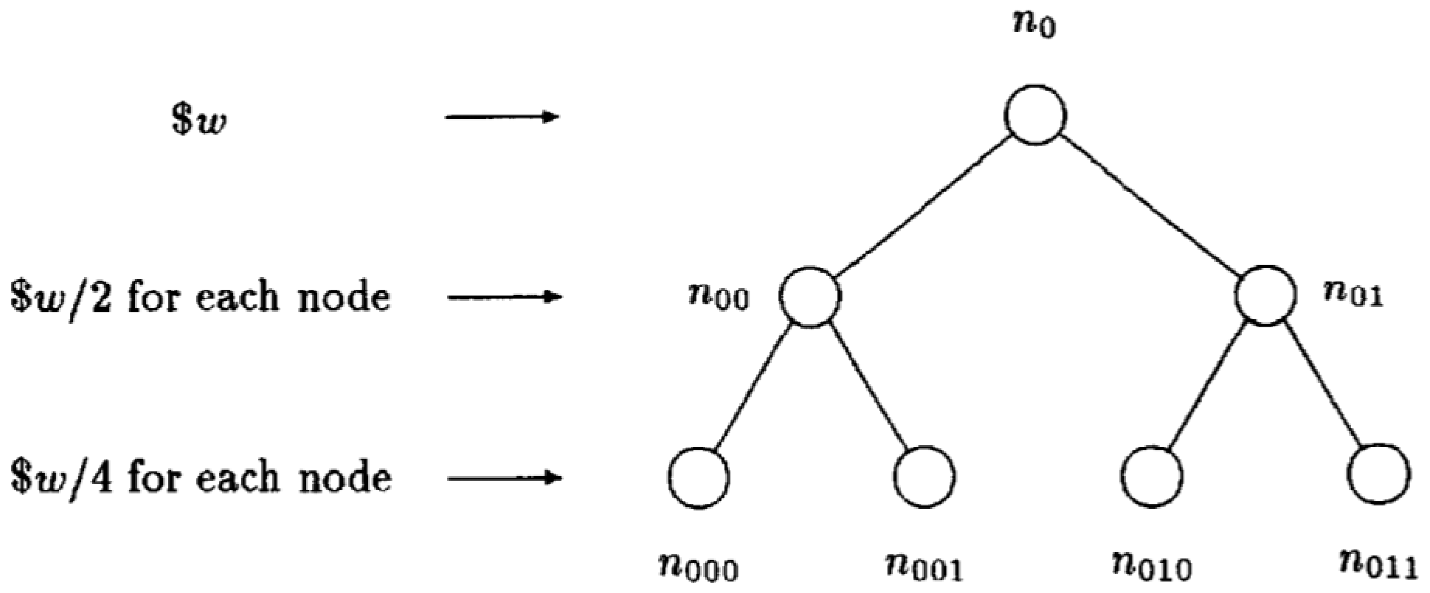
Jacobi symbol is defined as the product of Legendre symbols corresponding to the prime factors of N.

3. Binary Tree Approach

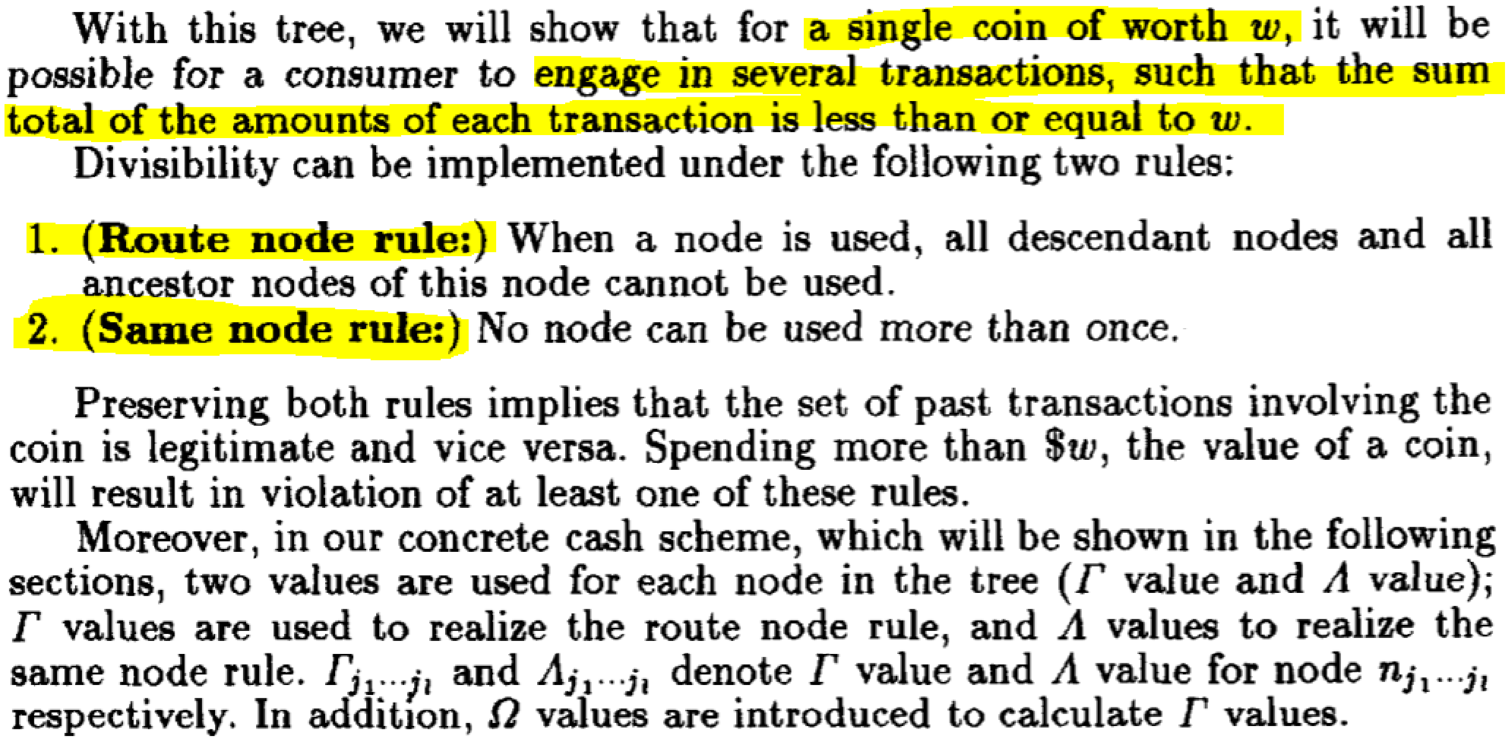


We will adopt the binary tree approach as do all previous divisible cash schemes [8, 11, 19, 20]. Each coin of worth $w = 2^l$ is associated with a tree of $(1 + l)$ levels and w leaves.

Each node of the tree represents a certain denomination. The root node, n_0, is assigned a monetary value of w, and the value of all other nodes, $n_{j_1 \cdots j_l}$, is found by halving the value of the node's parent, $n_{j_1 \cdots j_{l-1}}$ ($j_1 = 0$, $j_i \in \{0, 1\}$ for $i = 2, \ldots, l$).



3. Binary Tree Approach continued, page 2.



With this tree, we will show that for a single coin of worth w, it will be possible for a consumer to engage in several transactions, such that the sum total of the amounts of each transaction is less than or equal to w.

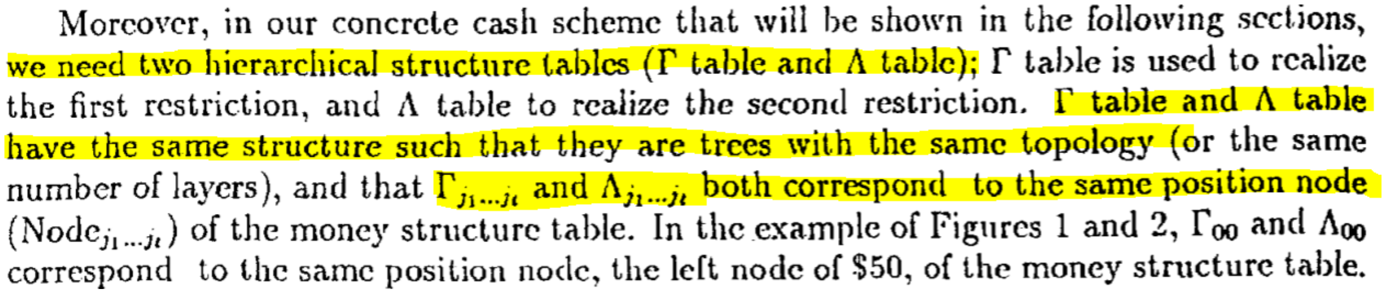
Divisibility can be implemented under the following two rules:

1. **(Route node rule:)** When a node is used, all descendant nodes and all ancestor nodes of this node cannot be used.
2. **(Same node rule:)** No node can be used more than once.

Preserving both rules implies that the set of past transactions involving the coin is legitimate and vice versa. Spending more than $\$w$, the value of a coin, will result in violation of at least one of these rules.

Moreover, in our concrete cash scheme, which will be shown in the following sections, two values are used for each node in the tree (Γ value and Λ value); Γ values are used to realize the route node rule, and Λ values to realize the same node rule. $\Gamma_{j_1 \cdots j_l}$ and $\Lambda_{j_1 \cdots j_l}$ denote Γ value and Λ value for node $n_{j_1 \cdots j_l}$ respectively. In addition, Ω values are introduced to calculate Γ values.

From [2] (Okamoto, Ohta):



Moreover, in our concrete cash scheme that will be shown in the following sections, we need two hierarchical structure tables (Γ table and Λ table); Γ table is used to realize the first restriction, and Λ table to realize the second restriction. Γ table and Λ table have the same structure such that they are trees with the same topology (or the same number of layers), and that $\Gamma_{j_1 \ldots j_l}$ and $\Lambda_{j_1 \ldots j_l}$ both correspond to the same position node (Node$_{j_1 \ldots j_l}$) of the money structure table. In the example of Figures 1 and 2, Γ_{00} and Λ_{00} correspond to the same position node, the left node of $50, of the money structure table.

3. Binary Tree Approach continued, page 3: Hierarchical Structure Table example.

Binary tree approach here is same as “The Hierarchical Structure Table” in [2].
In [2, p.325] the authors claim that in order to satisfy “Divisibility criterion” they combine “square root modulo N with the hierarchical structure table”. Where N is Williams number.

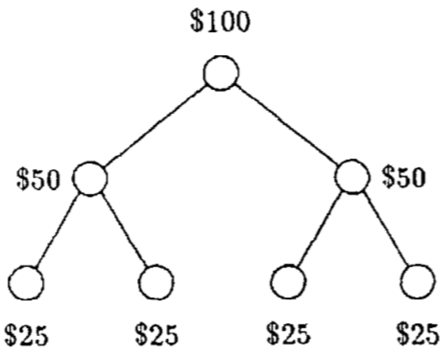


Figure 1: Hierarchical Structure Table (Money Structure)

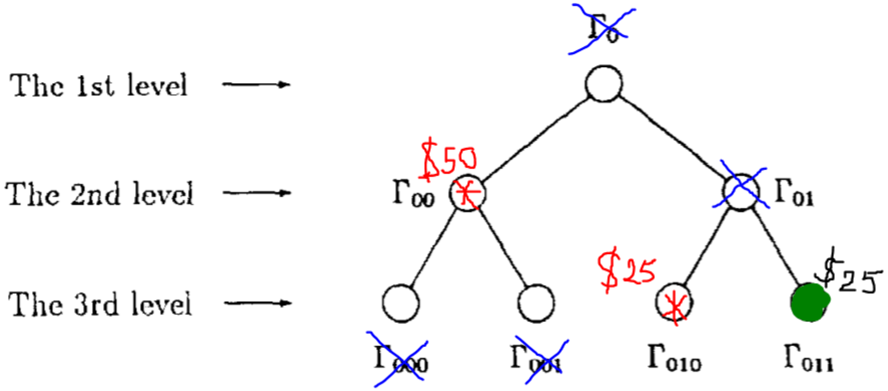
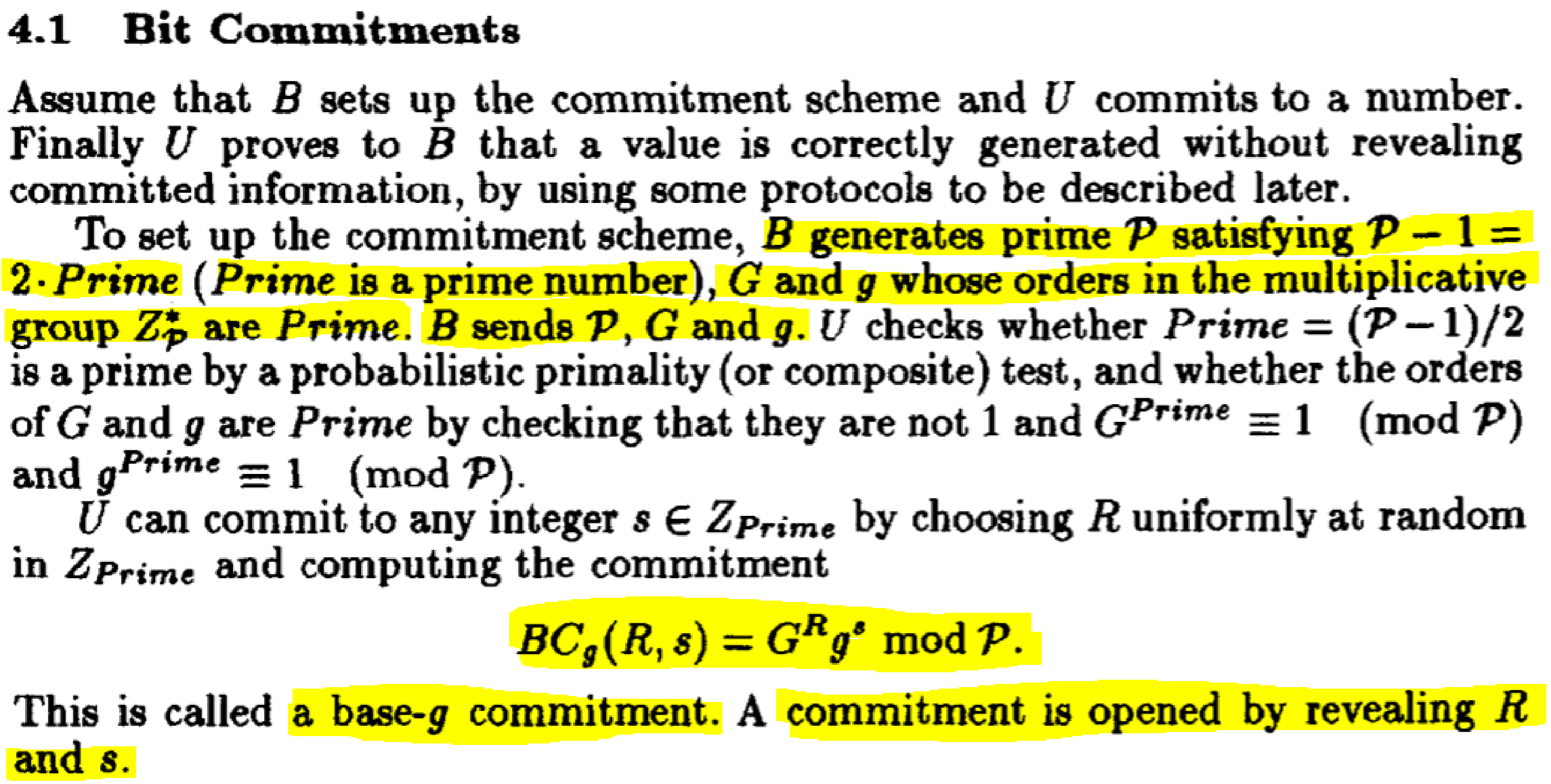


Figure 2: Hierarchical Structure Table (Γ Table)

Case: customer Alice uses $75 first and then uses $25. When she uses $75, she must use node G00 ($50), and node G010 ($25). From the above restrictions, only node G011 ($25) can be used after the use of G00 and G010 (see Figure 2).
Blue nodes are disabled due to “Route node rule” and Red nodes are disabled due to “Same node rule”.

4. Bit Commitment Schemes (BCS-s): Commitments and Protocols

- A BCS is used in the opening stage. Is based on the discrete log problem.
- BCS plays an essential role in realizing the “single-term” property of the proposed cash scheme.
- Previous schemes used the cut-and-choose method.



4.1 Bit Commitments

Assume that B sets up the commitment scheme and U commits to a number. Finally U proves to B that a value is correctly generated without revealing committed information, by using some protocols to be described later.

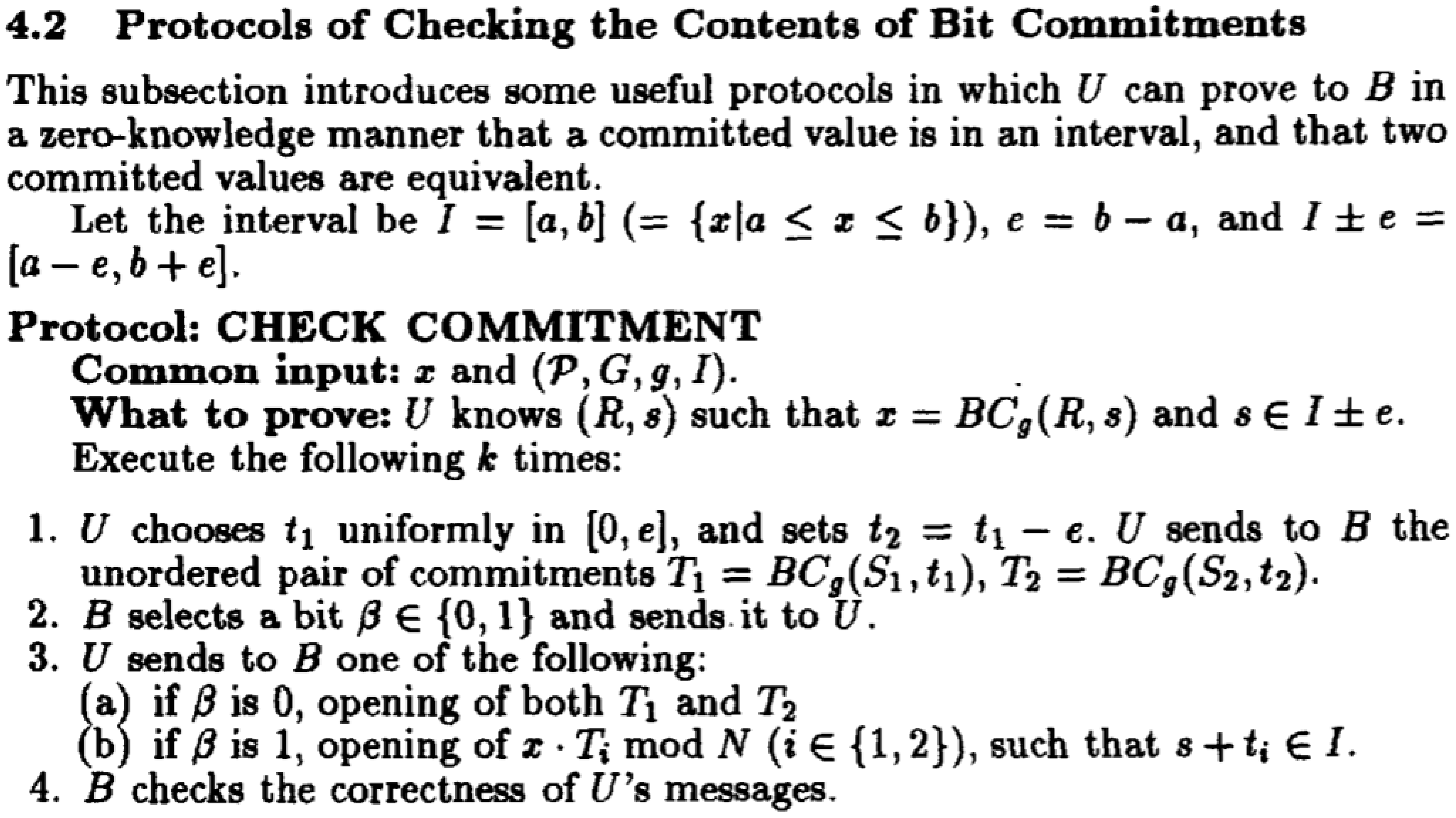
To set up the commitment scheme, B generates prime $\mathcal{P}$ satisfying $\mathcal{P} - 1 = 2 \cdot Prime$ ($Prime$ is a prime number), G and g whose orders in the multiplicative group $Z_{\mathcal{P}}^*$ are $Prime$. B sends $\mathcal{P}$, G and g. U checks whether $Prime = (\mathcal{P}-1)/2$ is a prime by a probabilistic primality (or composite) test, and whether the orders of G and g are $Prime$ by checking that they are not 1 and $G^{Prime} \equiv 1 \pmod{\mathcal{P}}$ and $g^{Prime} \equiv 1 \pmod{\mathcal{P}}$.

U can commit to any integer $s \in Z_{Prime}$ by choosing R uniformly at random in Z_{Prime} and computing the commitment

$$BC_g(R, s) = G^R g^s \bmod \mathcal{P}.$$

This is called a base-g commitment. A commitment is opened by revealing R and s.

4.2. Protocols, page 1



4.2 Protocols of Checking the Contents of Bit Commitments

This subsection introduces some useful protocols in which U can prove to B in a zero-knowledge manner that a committed value is in an interval, and that two committed values are equivalent.

Let the interval be $I = [a, b]$ $(= \{x | a \leq x \leq b\})$, $e = b - a$, and $I \pm e = [a - e, b + e]$.

Protocol: CHECK COMMITMENT

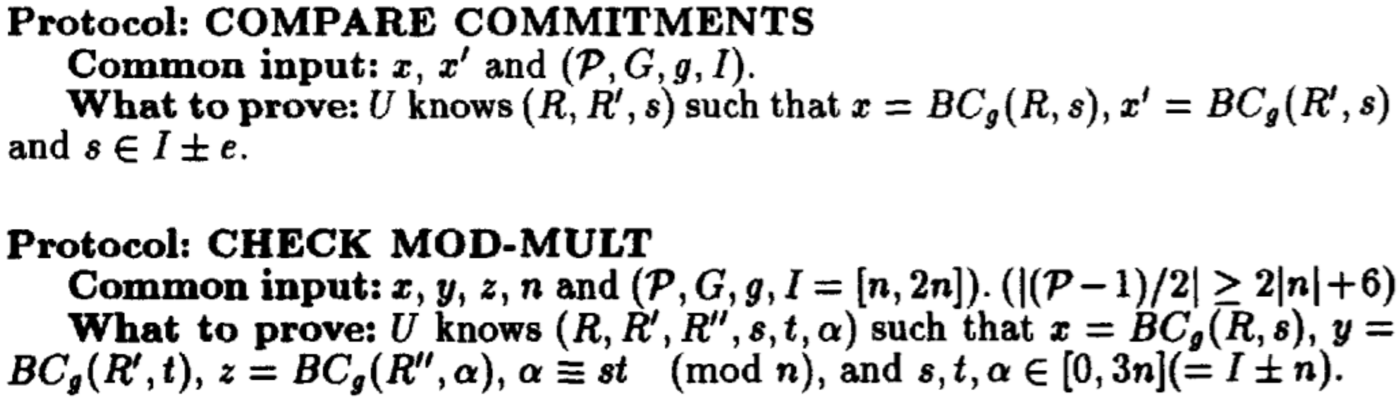
Common input: x and $(\mathcal{P}, G, g, I)$.

What to prove: U knows (R, s) such that $x = BC_g(R, s)$ and $s \in I \pm e$.

Execute the following k times:

1. U chooses t_1 uniformly in $[0, e]$, and sets $t_2 = t_1 - e$. U sends to B the unordered pair of commitments $T_1 = BC_g(S_1, t_1)$, $T_2 = BC_g(S_2, t_2)$.
2. B selects a bit $\beta \in \{0, 1\}$ and sends it to U.
3. U sends to B one of the following:
   (a) if β is 0, opening of both T_1 and T_2
   (b) if β is 1, opening of $x \cdot T_i \bmod N$ $(i \in \{1, 2\})$, such that $s + t_i \in I$.
4. B checks the correctness of U's messages.

4.2. Protocols, page 2: COMPARE COMMITMENTS and CHECK MOD-MULT



Protocol: COMPARE COMMITMENTS

Common input: x, x' and $(\mathcal{P}, G, g, I)$.

What to prove: U knows (R, R', s) such that $x = BC_g(R, s)$, $x' = BC_g(R', s)$ and $s \in I \pm e$.

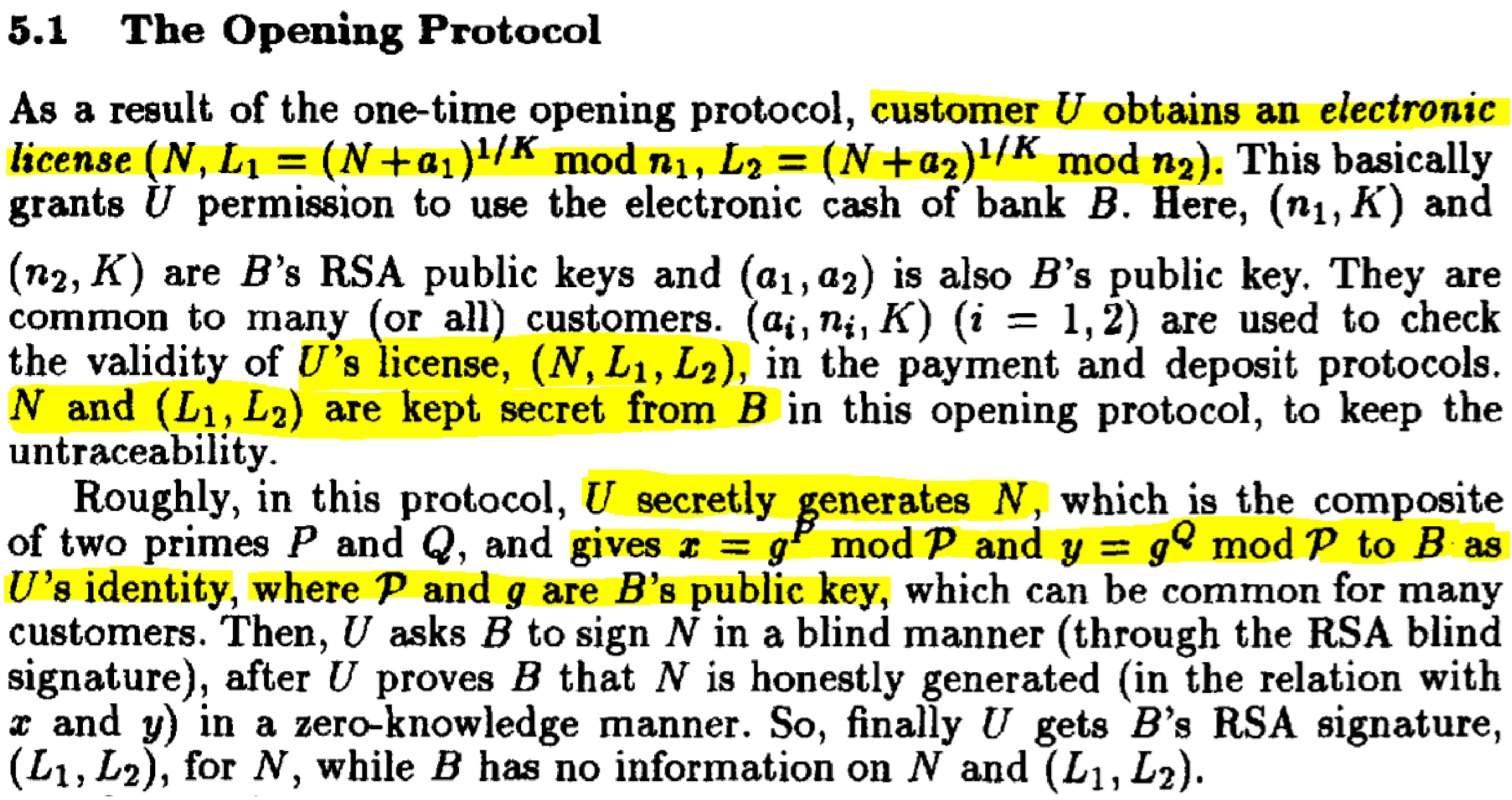
Protocol: CHECK MOD-MULT

Common input: x, y, z, n and $(\mathcal{P}, G, g, I = [n, 2n])$. $(|(\mathcal{P}-1)/2| \geq 2|n|+6)$

What to prove: U knows $(R, R', R'', s, t, \alpha)$ such that $x = BC_g(R, s)$, $y = BC_g(R', t)$, $z = BC_g(R'', \alpha)$, $\alpha \equiv st \pmod{n}$, and $s, t, \alpha \in [0, 3n](= I \pm n)$.

5. Efficient Divisible Cash Scheme

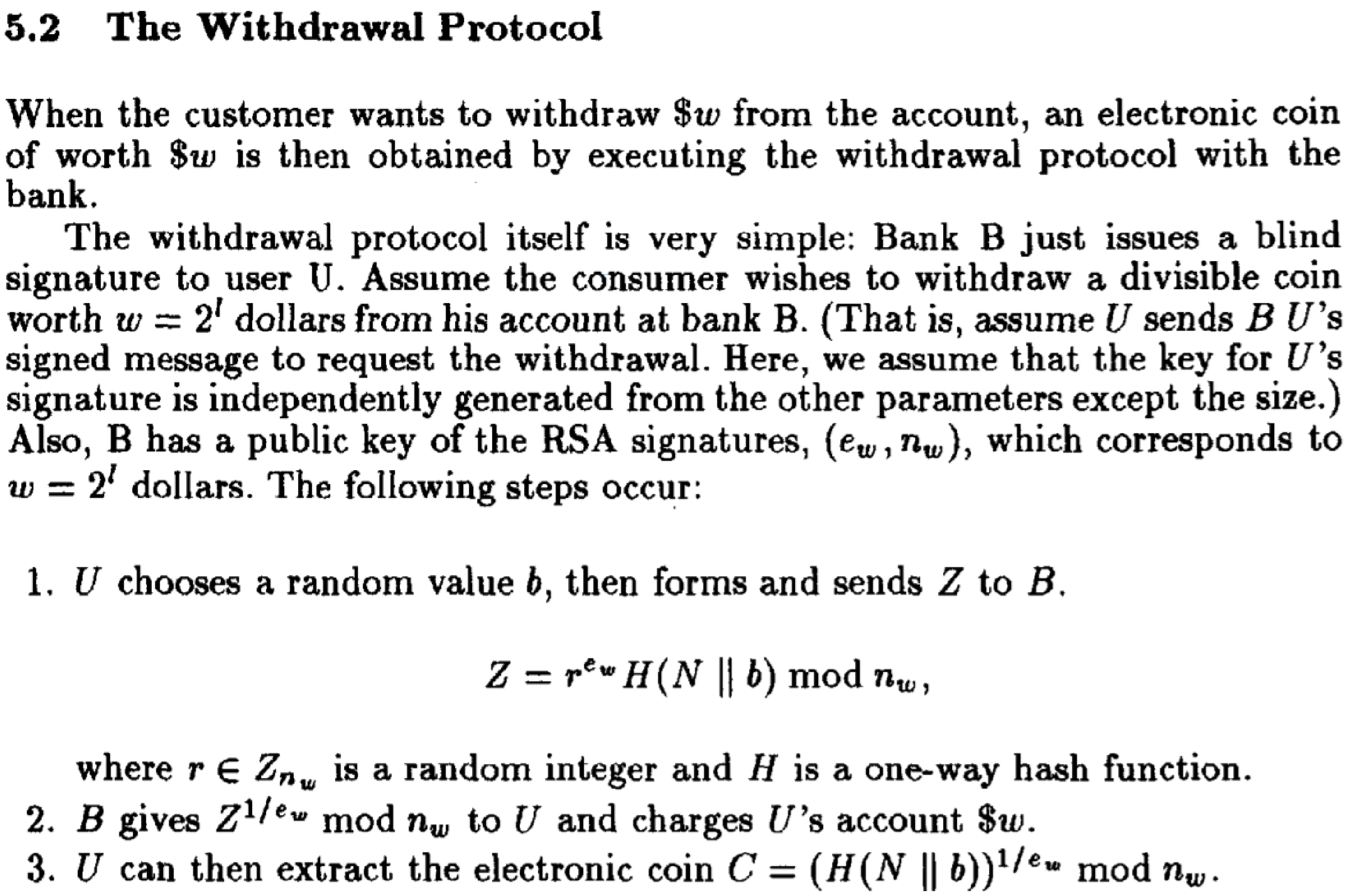
- The electronic cash in our proposed scheme consists of an electronic license and electronic coins.
- The electronic license is issued by the bank to a customer during an "opening protocol".
- This protocol is done once per customer, typically when a customer opens an account. If, however, a customer prefers to change the license at some later time, or desires several licenses, this protocol is run again for each additional license.
- Frequency of license changes: trade-off between the degree of unlinkability and efficiency.



5.1 The Opening Protocol

As a result of the one-time opening protocol, customer U obtains an *electronic license* $(N, L_1 = (N+a_1)^{1/K} \bmod n_1, L_2 = (N+a_2)^{1/K} \bmod n_2)$. This basically grants U permission to use the electronic cash of bank B. Here, (n_1, K) and (n_2, K) are B's RSA public keys and (a_1, a_2) is also B's public key. They are common to many (or all) customers. (a_i, n_i, K) $(i = 1, 2)$ are used to check the validity of U's license, (N, L_1, L_2), in the payment and deposit protocols. N and (L_1, L_2) are kept secret from B in this opening protocol, to keep the untraceability.

Roughly, in this protocol, U secretly generates N, which is the composite of two primes P and Q, and gives $x = g^P \bmod \mathcal{P}$ and $y = g^Q \bmod \mathcal{P}$ to B as U's identity, where $\mathcal{P}$ and g are B's public key, which can be common for many customers. Then, U asks B to sign N in a blind manner (through the RSA blind signature), after U proves B that N is honestly generated (in the relation with x and y) in a zero-knowledge manner. So, finally U gets B's RSA signature, (L_1, L_2), for N, while B has no information on N and (L_1, L_2).



5.2 The Withdrawal Protocol

When the customer wants to withdraw \$$w$ from the account, an electronic coin of worth \$$w$ is then obtained by executing the withdrawal protocol with the bank.

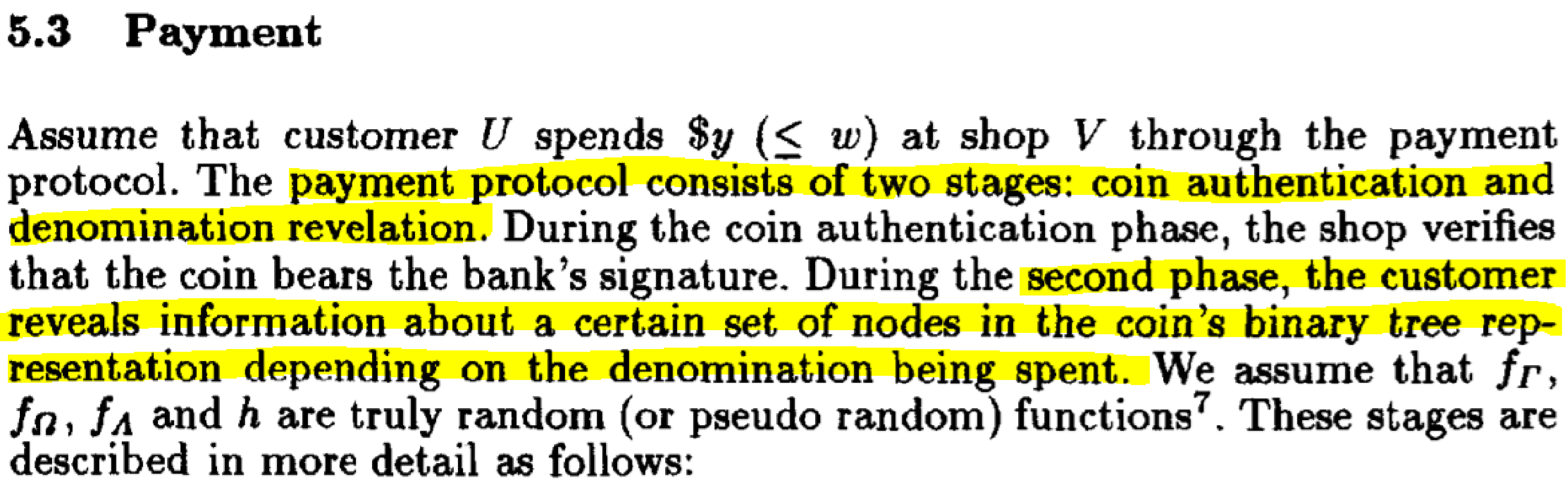
The withdrawal protocol itself is very simple: Bank B just issues a blind signature to user U. Assume the consumer wishes to withdraw a divisible coin worth $w = 2^l$ dollars from his account at bank B. (That is, assume U sends B U's signed message to request the withdrawal. Here, we assume that the key for U's signature is independently generated from the other parameters except the size.) Also, B has a public key of the RSA signatures, (e_w, n_w), which corresponds to $w = 2^l$ dollars. The following steps occur:

1. U chooses a random value b, then forms and sends Z to B.

$$Z = r^{e_w} H(N \parallel b) \bmod n_w,$$

   where $r \in Z_{n_w}$ is a random integer and H is a one-way hash function.
2. B gives $Z^{1/e_w} \bmod n_w$ to U and charges U's account \$$w$.
3. U can then extract the electronic coin $C = (H(N \parallel b))^{1/e_w} \bmod n_w$.

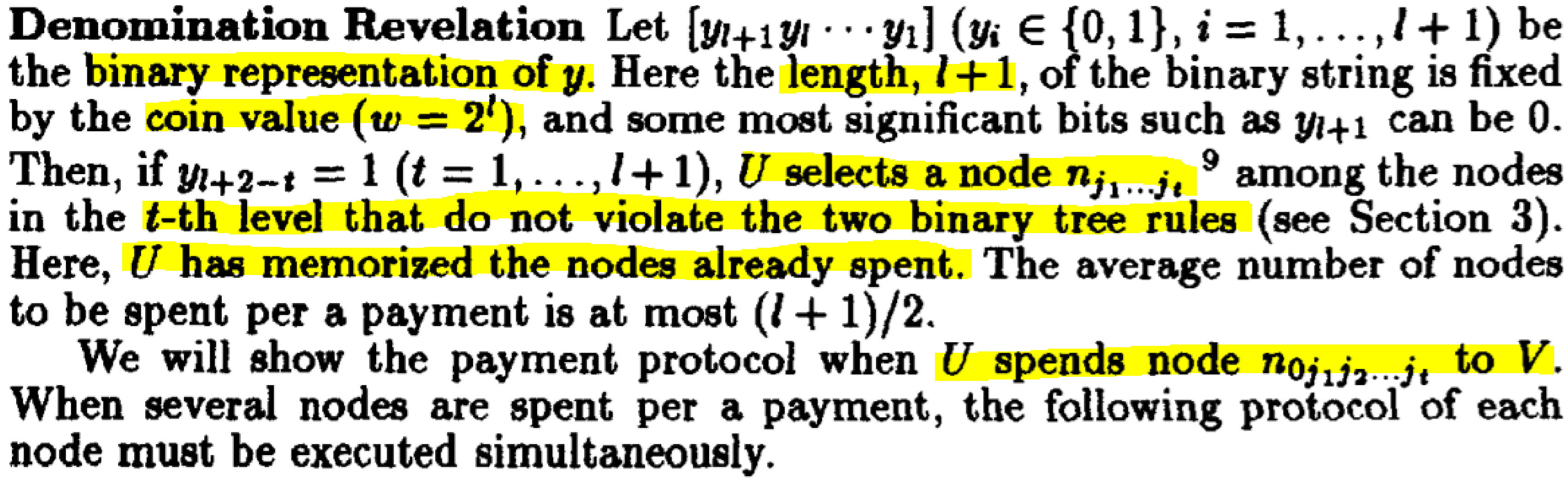
Extracted coin C = “answer from bank B” / (r modulo nw)



5.3 Payment

Assume that customer U spends \$$y$ ($\leq w$) at shop V through the payment protocol. The payment protocol consists of two stages: coin authentication and denomination revelation. During the coin authentication phase, the shop verifies that the coin bears the bank's signature. During the second phase, the customer reveals information about a certain set of nodes in the coin's binary tree representation depending on the denomination being spent. We assume that f_Γ, f_Ω, f_Λ and h are truly random (or pseudo random) functions[7]. These stages are described in more detail as follows:

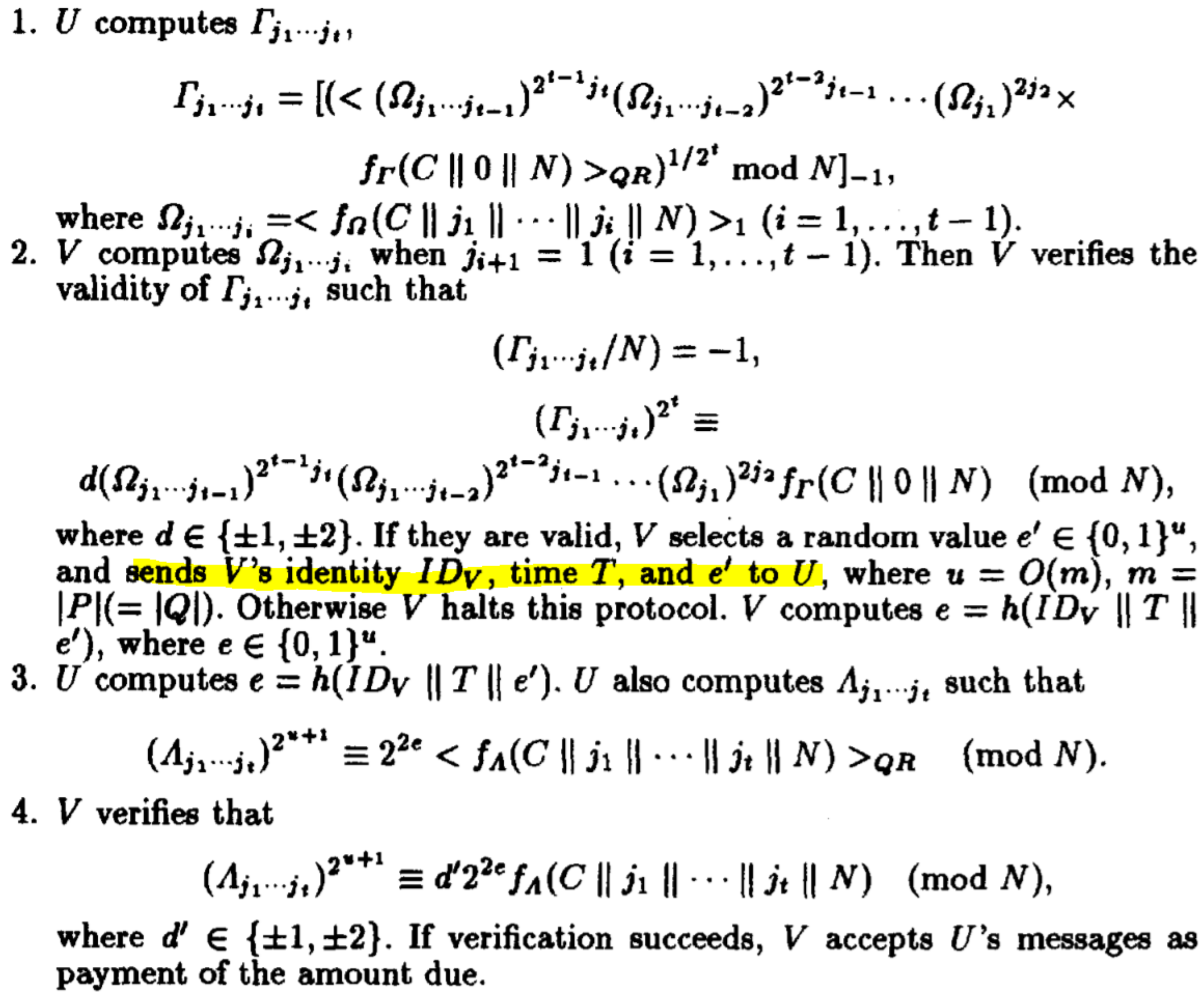
We skip authentication phase and look at denomination revelation because it is directly related to divisibility.



Denomination Revelation Let $[y_{l+1}y_l \cdots y_1]$ ($y_i \in \{0,1\}$, $i = 1, \ldots, l+1$) be the binary representation of y. Here the length, $l+1$, of the binary string is fixed by the coin value ($w = 2^l$), and some most significant bits such as y_{l+1} can be 0. Then, if $y_{l+2-t} = 1$ ($t = 1, \ldots, l+1$), U selects a node $n_{j_1 \ldots j_t}$ [9] among the nodes in the t-th level that do not violate the two binary tree rules (see Section 3). Here, U has memorized the nodes already spent. The average number of nodes to be spent per a payment is at most $(l+1)/2$.

We will show the payment protocol when U spends node $n_{0j_1j_2\ldots j_t}$ to V. When several nodes are spent per a payment, the following protocol of each node must be executed simultaneously.

5.3. Continued: Denomination Revelation details



1. U computes $\Gamma_{j_1\cdots j_t}$,

$$\Gamma_{j_1\cdots j_t} = [(<(\Omega_{j_1\cdots j_{t-1}})^{2^{t-1}j_t}(\Omega_{j_1\cdots j_{t-2}})^{2^{t-2}j_{t-1}}\cdots(\Omega_{j_1})^{2j_2}\times$$
$$f_\Gamma(C \,\|\, 0 \,\|\, N) >_{QR})^{1/2^t} \bmod N]_{-1},$$

where $\Omega_{j_1\cdots j_i} =< f_\Omega(C \,\|\, j_1 \,\|\, \cdots \,\|\, j_i \,\|\, N) >_1$ $(i = 1, \ldots, t-1)$.

2. V computes $\Omega_{j_1\cdots j_i}$ when $j_{i+1} = 1$ $(i = 1, \ldots, t-1)$. Then V verifies the validity of $\Gamma_{j_1\cdots j_t}$ such that

$$(\Gamma_{j_1\cdots j_t}/N) = -1,$$
$$(\Gamma_{j_1\cdots j_t})^{2^t} \equiv$$
$$d(\Omega_{j_1\cdots j_{t-1}})^{2^{t-1}j_t}(\Omega_{j_1\cdots j_{t-2}})^{2^{t-2}j_{t-1}}\cdots(\Omega_{j_1})^{2j_2}f_\Gamma(C \,\|\, 0 \,\|\, N) \pmod N,$$

where $d \in \{\pm1, \pm2\}$. If they are valid, V selects a random value $e' \in \{0,1\}^u$, and sends V's identity ID_V, time T, and e' to U, where $u = O(m)$, $m = |P|(= |Q|)$. Otherwise V halts this protocol. V computes $e = h(ID_V \,\|\, T \,\|\, e')$, where $e \in \{0,1\}^u$.

3. U computes $e = h(ID_V \,\|\, T \,\|\, e')$. U also computes $\Lambda_{j_1\cdots j_t}$ such that

$$(\Lambda_{j_1\cdots j_t})^{2^{u+1}} \equiv 2^{2e} < f_\Lambda(C \,\|\, j_1 \,\|\, \cdots \,\|\, j_t \,\|\, N) >_{QR} \pmod N.$$

4. V verifies that

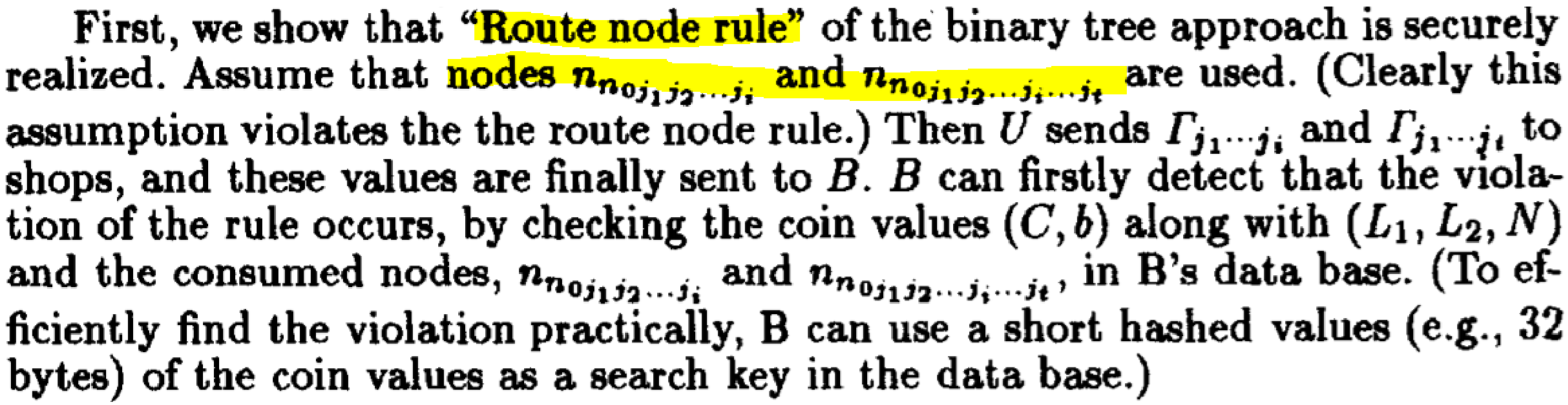
$$(\Lambda_{j_1\cdots j_t})^{2^{u+1}} \equiv d'2^{2e}f_\Lambda(C \,\|\, j_1 \,\|\, \cdots \,\|\, j_t \,\|\, N) \pmod N,$$

where $d' \in \{\pm1, \pm2\}$. If verification succeeds, V accepts U's messages as payment of the amount due.

5.4. Deposit: same as payment p.5.3.

5.5. Detection of Overspending, page 1

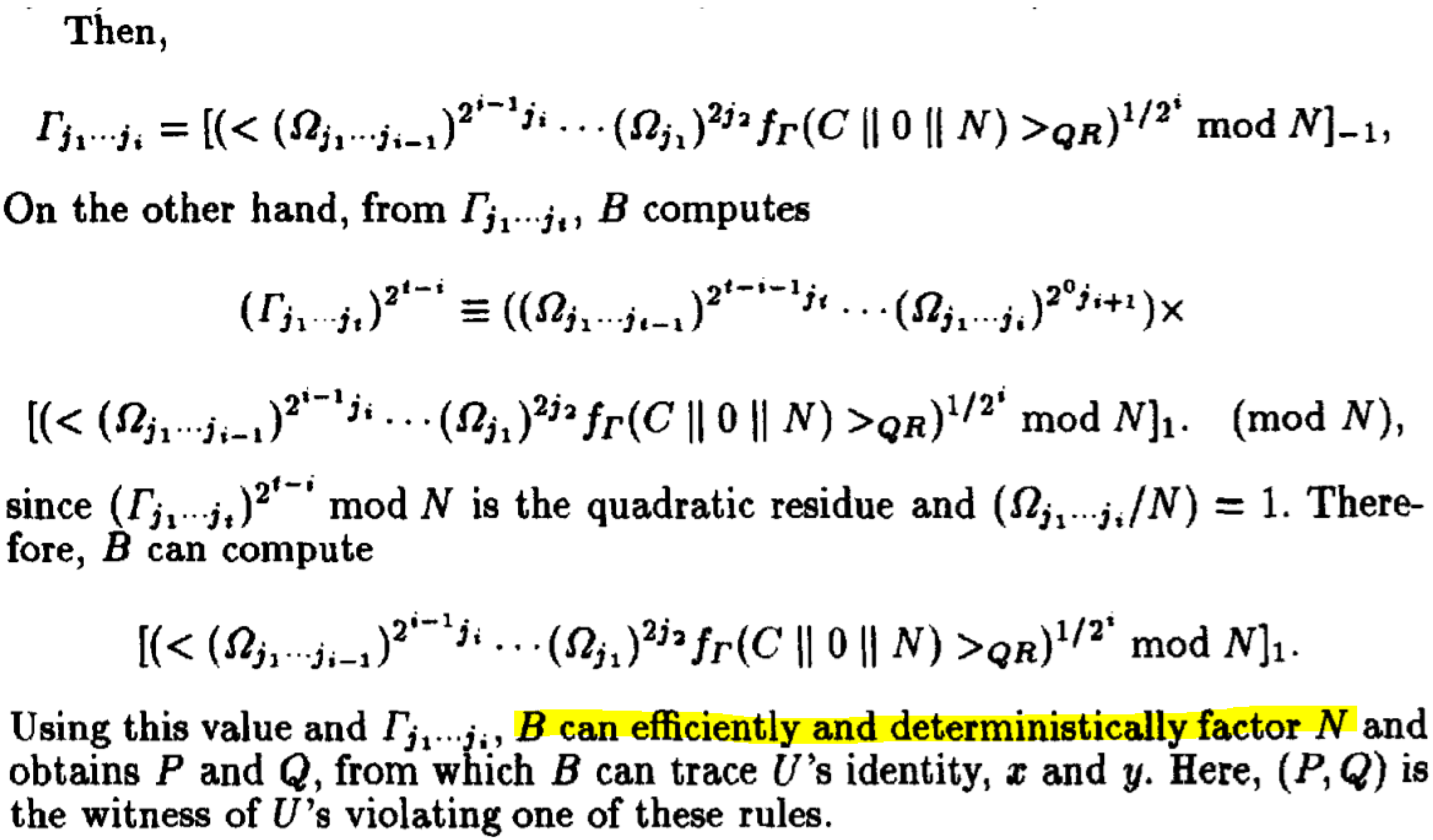
If U overspends a coin, U must violate one of the two rules of the binary tree approach: “Route node rule” or “Same node rule”.



First, we show that “Route node rule” of the binary tree approach is securely realized. Assume that nodes $n_{n_0 j_1 j_2 \cdots j_i}$ and $n_{n_0 j_1 j_2 \cdots j_i \cdots j_t}$ are used. (Clearly this assumption violates the the route node rule.) Then U sends $\Gamma_{j_1 \cdots j_i}$ and $\Gamma_{j_1 \cdots j_t}$ to shops, and these values are finally sent to B. B can firstly detect that the violation of the rule occurs, by checking the coin values (C, b) along with (L_1, L_2, N) and the consumed nodes, $n_{n_0 j_1 j_2 \cdots j_i}$ and $n_{n_0 j_1 j_2 \cdots j_i \cdots j_t}$, in B’s data base. (To efficiently find the violation practically, B can use a short hashed values (e.g., 32 bytes) of the coin values as a search key in the data base.)

Here the node with index j_t is a descendant of the node j_i, therefore violating “route node rule”.

5.5. Detection of Overspending continued, page 2



Then,

$$\Gamma_{j_1\cdots j_i} = [(< (\Omega_{j_1\cdots j_{i-1}})^{2^{i-1}j_i} \cdots (\Omega_{j_1})^{2j_2} f_\Gamma(C \parallel 0 \parallel N) >_{QR})^{1/2^i} \bmod N]_{-1},$$

On the other hand, from $\Gamma_{j_1\cdots j_t}$, B computes

$$(\Gamma_{j_1\cdots j_t})^{2^{t-i}} \equiv ((\Omega_{j_1\cdots j_{t-1}})^{2^{t-i-1}j_t} \cdots (\Omega_{j_1\cdots j_i})^{2^0 j_{i+1}})\times$$

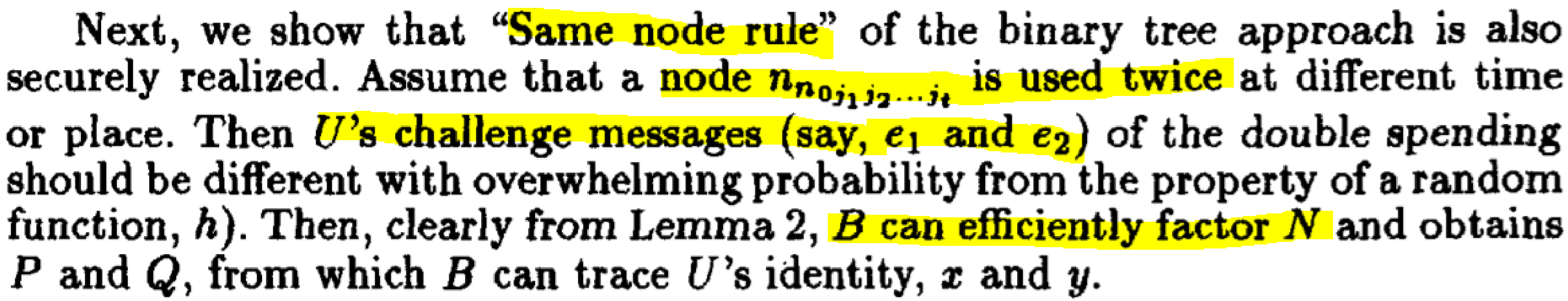
$$[(< (\Omega_{j_1\cdots j_{i-1}})^{2^{i-1}j_i} \cdots (\Omega_{j_1})^{2j_2} f_\Gamma(C \parallel 0 \parallel N) >_{QR})^{1/2^i} \bmod N]_1. \quad (\bmod N),$$

since $(\Gamma_{j_1\cdots j_t})^{2^{t-i}} \bmod N$ is the quadratic residue and $(\Omega_{j_1\cdots j_i}/N) = 1$. Therefore, B can compute

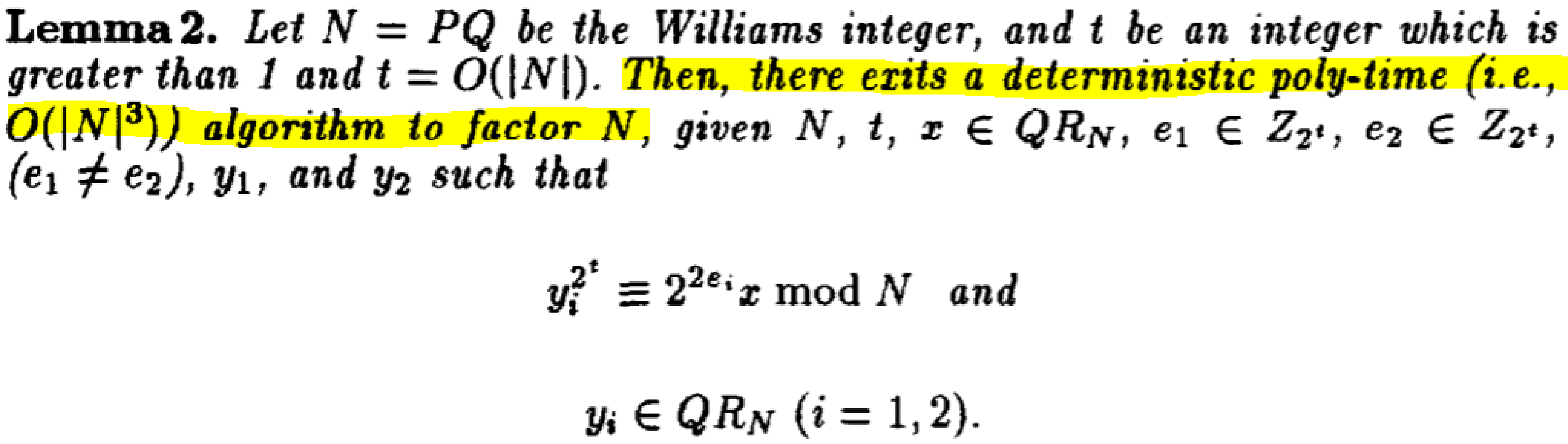
$$[(< (\Omega_{j_1\cdots j_{i-1}})^{2^{i-1}j_i} \cdots (\Omega_{j_1})^{2j_2} f_\Gamma(C \parallel 0 \parallel N) >_{QR})^{1/2^i} \bmod N]_1.$$

Using this value and $\Gamma_{j_1\cdots j_i}$, B can efficiently and deterministically factor N and obtains P and Q, from which B can trace U's identity, x and y. Here, (P, Q) is the witness of U's violating one of these rules.

5.5. Detection of Overspending continued, page 3



Next, we show that "Same node rule" of the binary tree approach is also securely realized. Assume that a node $n_{n_{0 j_1 j_2 \cdots j_t}}$ is used twice at different time or place. Then U's challenge messages (say, e_1 and e_2) of the double spending should be different with overwhelming probability from the property of a random function, h). Then, clearly from Lemma 2, B can efficiently factor N and obtains P and Q, from which B can trace U's identity, x and y.

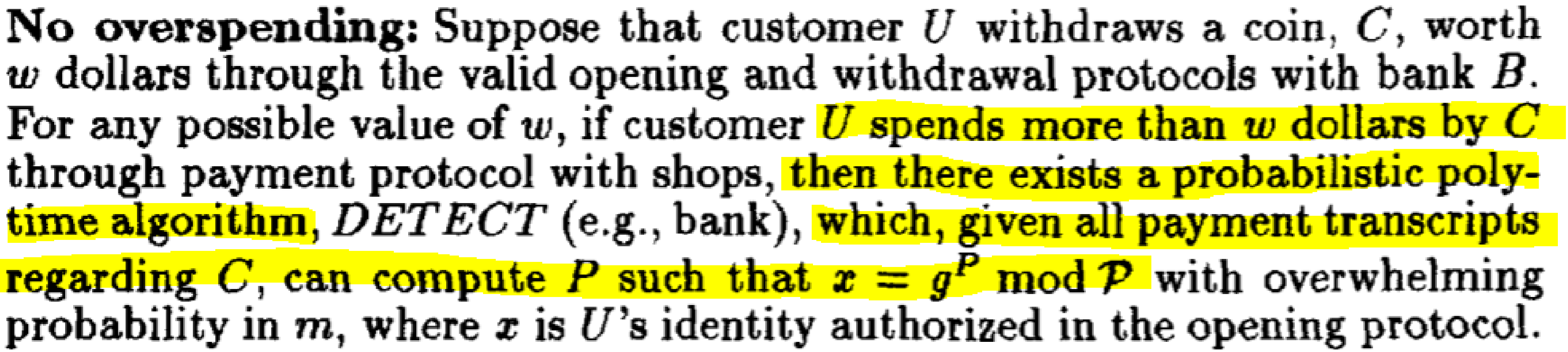


Lemma 2. *Let $N = PQ$ be the Williams integer, and t be an integer which is greater than 1 and $t = O(|N|)$. Then, there exits a deterministic poly-time (i.e., $O(|N|^3)$) algorithm to factor N, given N, t, $x \in QR_N$, $e_1 \in Z_{2^t}$, $e_2 \in Z_{2^t}$, $(e_1 \neq e_2)$, y_1, and y_2 such that*

$$y_i^{2^t} \equiv 2^{2e_i} x \bmod N \quad \textit{and}$$

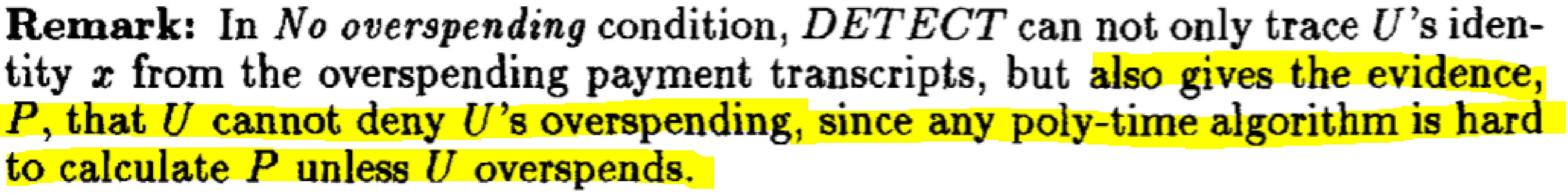
$$y_i \in QR_N \ (i = 1, 2).$$

5.5. Detection of Overspending continued, page 4



No overspending: Suppose that customer U withdraws a coin, C, worth w dollars through the valid opening and withdrawal protocols with bank B. For any possible value of w, if customer U spends more than w dollars by C through payment protocol with shops, then there exists a probabilistic poly-time algorithm, $DETECT$ (e.g., bank), which, given all payment transcripts regarding C, can compute P such that $x = g^P \bmod \mathcal{P}$ with overwhelming probability in m, where x is U's identity authorized in the opening protocol.

Where $m = |P| = |Q|$

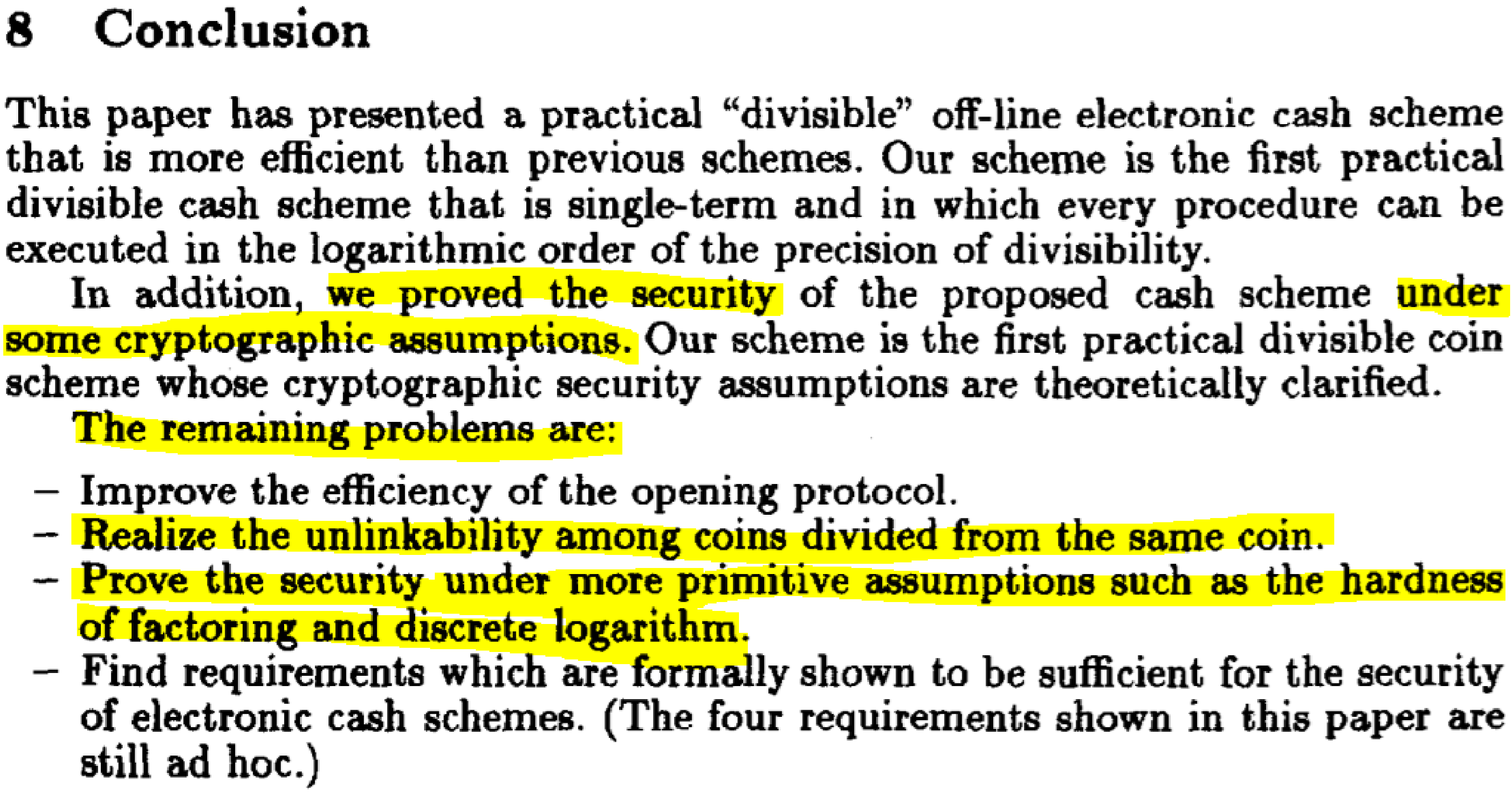


Remark: In *No overspending* condition, $DETECT$ can not only trace U's identity x from the overspending payment transcripts, but also gives the evidence, P, that U cannot deny U's overspending, since any poly-time algorithm is hard to calculate P unless U overspends.

It is important to note that poly-time algorithm "works" only in case overspending happens!

Remember the "opening protocol": customer U secretly generated N, which is the composite of two primes P and Q and gave to the bank: $x = g^P mod\, P_b$

P_b on the other hand is a public key of the bank B.



8 Conclusion

This paper has presented a practical "divisible" off-line electronic cash scheme that is more efficient than previous schemes. Our scheme is the first practical divisible cash scheme that is single-term and in which every procedure can be executed in the logarithmic order of the precision of divisibility.

In addition, we proved the security of the proposed cash scheme under some cryptographic assumptions. Our scheme is the first practical divisible coin scheme whose cryptographic security assumptions are theoretically clarified.

The remaining problems are:

- Improve the efficiency of the opening protocol.
- Realize the unlinkability among coins divided from the same coin.
- Prove the security under more primitive assumptions such as the hardness of factoring and discrete logarithm.
- Find requirements which are formally shown to be sufficient for the security of electronic cash schemes. (The four requirements shown in this paper are still ad hoc.)

My conclusion, take-aways, thoughts

1. Quadratic residues and Williams integers are your best friends.
2. Pretty neat application of number theory in case of double-spending: reveals the identity of the bad actor, otherwise anonymous.
3. Williams integer N can be factored in poly-time from $[\sqrt{x} \; mod \; N]_1$ and $[\sqrt{x} \; mod \; N]_{-1}$ [2] p.334
4. Quite soft wording on security: “Although it is unclear whether these four security requirements are sufficient (!), our security proof guarantees that if there exists an attack on our scheme, then it should reside outside these four security requirements, unless our security assumptions are broken.” (4 security requirements: no forging, no tracing, no overspending and no swindling).
5. Unlinkability: a solution offered in [9].
6. On a more philosophical note: why do we care about Privacy and Unlinkability at all?
7. Number Theory all the way in addition to already mentioned topics: Euler's criterion, Quadratic Reciprocity, Chinese remainder theorem, ... etc.

References

[1] T. Okamoto, “An efficient divisible electronic cash scheme," Annual International Cryptology Conference - CRYPTO 1995: Advances in Cryptology - CRYPTO ‘95, pages 438-451, Springer-Verlag, New York, 1995.

[2] Okamoto, T., and Ohta, K., “Universal Electronic Cash”, Proceedings of Crypto 91, pp. 324–337 (1992).

[3] Eng, T. and Okamoto, T. “Single-Term Divisible Coins,” to appear in the Proceedings of Eurocrypt 94.

[4] Ferguson, N., “Single Term Off-line Coins”, Proceedings of Eurocrypt 93, pp.318–328 (1994).

[5] Franklin, M. and Yung, M., “Secure and Efficient Off-Line Digital Money”, Proceedings of ICALP 93, pp. 449–460 (1993).

[6] https://en.wikipedia.org/wiki/Jacobi_symbol

[7] https://en.wikipedia.org/wiki/Multiplicative_order

[8] Zhu R., Huang Y., Katz J., “The Cut-and-Choose Game and its Application to Cryptographic Protocols”, https://www.cs.umd.edu/~jkatz/papers/cut-and-choose-game.pdf

[9] Sébastien CanardAline Gouget, “Divisible E-Cash Systems Can Be Truly Anonymous”, Annual International Conference on the Theory and Applications of Cryptographic Techniques: EUROCRYPT 2007: Advances in Cryptology - EUROCRYPT 2007 pp 482-497.

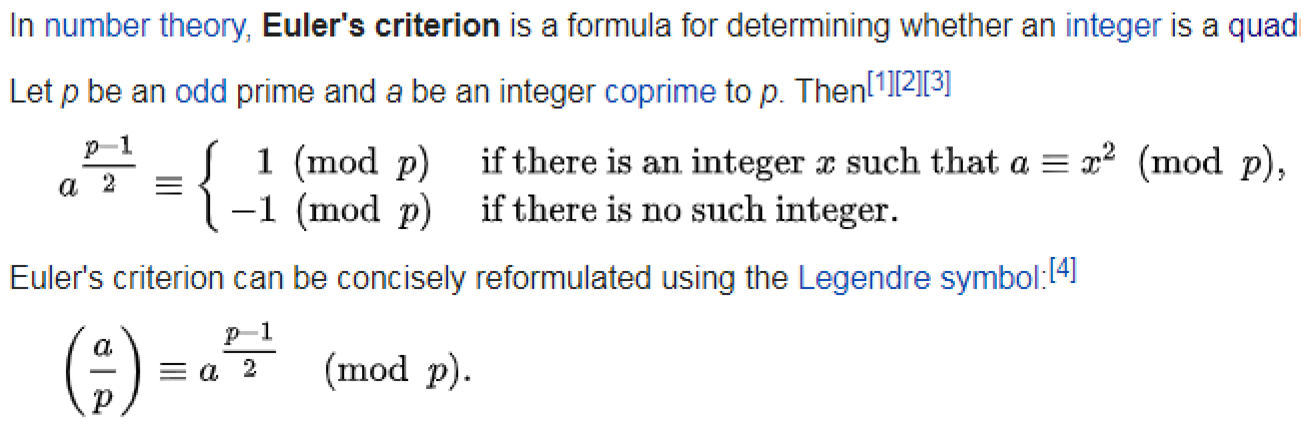
Appendix 1: Glossary

binary tree approach: each coin of worth C = 2^k is associated with a binary tree of (1 + k) levels and C leaves. Each node of the tree represents a certain denomination. The root node, n0, is assigned a monetary value of C, and the value of all other nodes is by halving the value of the node’s parent. Further details see [1].

cut-and-choose method: The basic idea is that one party constructs n versions of a message in a protocol; the other party randomly checks some of them and uses the rest of them in the protocol. Further details see [8].

divisibility (dividability): one issued piece of cash worth value C (dollars) can be subdivided into many pieces such that each subdivided piece is worth any desired values less than C and the total value of all pieces is equivalent to C.

Euler's criterion: https://en.wikipedia.org/wiki/Euler%27s_criterion



In number theory, **Euler's criterion** is a formula for determining whether an integer is a quadratic residue modulo a prime. Precisely,

Let *p* be an odd prime and *a* be an integer coprime to *p*. Then[1][2][3]

$$a^{\frac{p-1}{2}} \equiv \begin{cases} \;\;1 \pmod{p} & \text{if there is an integer } x \text{ such that } a \equiv x^2 \pmod{p}, \\ -1 \pmod{p} & \text{if there is no such integer.} \end{cases}$$

Euler's criterion can be concisely reformulated using the Legendre symbol:[4]

$$\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \pmod{p}.$$

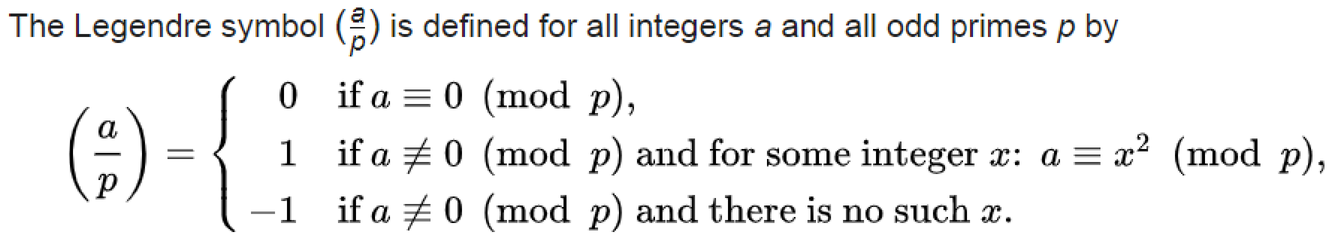
Gamma table/value: used to realize the route node rule. Gamma table is a tree consisting of gamma values of the nodes. Matches exactly the structure of the coin tree. Further details see [1], [2].

hierarchical structure table: same as binary tree approach. Further details see [2] and [1].

Jacobi symbol: the product of Legendre symbols corresponding to the prime factors of N.

Lambda table/value: used to realize the same node rule. Lambda table is a tree consisting of lambda values of the nodes. Matches exactly the structure of the coin tree. Further details see [1], [2].

Legendre symbol: https://en.wikipedia.org/wiki/Jacobi_symbol



The Legendre symbol $\left(\frac{a}{p}\right)$ is defined for all integers *a* and all odd primes *p* by

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{if } a \equiv 0 \pmod{p}, \\ 1 & \text{if } a \not\equiv 0 \pmod{p} \text{ and for some integer } x\text{: } a \equiv x^2 \pmod{p}, \\ -1 & \text{if } a \not\equiv 0 \pmod{p} \text{ and there is no such } x. \end{cases}$$

multiplicative order: the smallest positive integer *k* with $a^k \equiv 1\ (mod\ n)$

https://en.wikipedia.org/wiki/Multiplicative_order

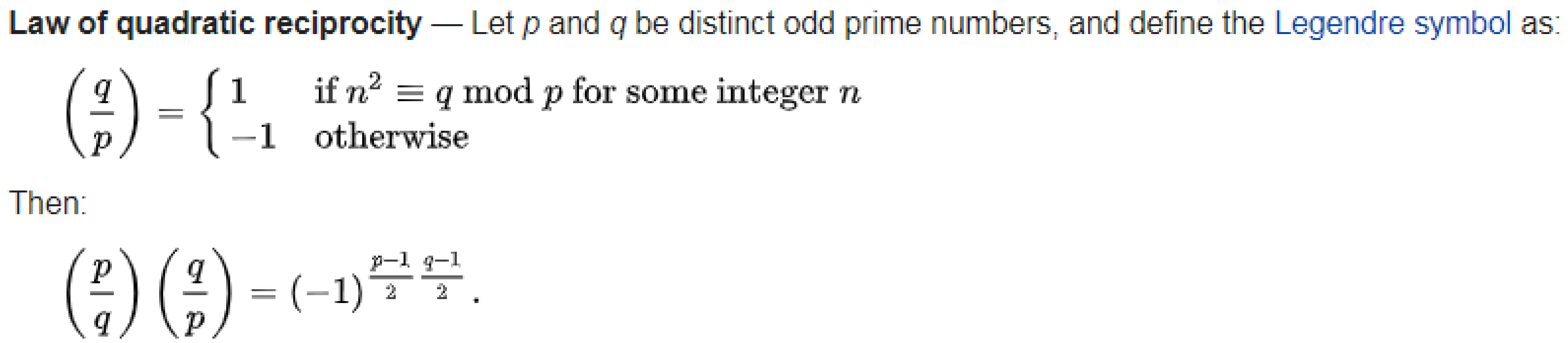
no swindling [2]: For any possible value of *w*, for any poly-time nonuniform algorithm *Adv* (e.g., dishonest shop), given all transcripts of opening and withdrawal protocols and after *Advs* executions as a shop of the payment protocol with customers, in which the total payment value is *w* dollars, the probability that *Adv* can deposit more than *w* dollars at Bank B is negligible in *m*. Where m = |P| = |Q|.

Prime: “special” prime number generated by bank B so that following is satisfied:

- another prime number *P* = *Prime* * 2 + 1, ie *Prime* = (*P* - 1) / 2
- the orders of parameters *G* and *g* in the multiplicative group Z(*P*) are *Prime*. Where *G* and *g* are parameters (large numbers) generated by bank B in the initialization phase of the opening protocol. (*P*,*G*,*g*) are sent to customer U.

QRN: QR_N quadratic residue (mod N). An integer *q* is called a quadratic residue modulo n if it is congruent to a perfect square modulo n; i.e., if there exists an integer x such that: $x^2 \equiv q \ (mod\ n)$. Further details see [2] and https://en.wikipedia.org/wiki/Quadratic_residue.

quadratic reciprocity: https://en.wikipedia.org/wiki/Quadratic_reciprocity



Law of quadratic reciprocity — Let *p* and *q* be distinct odd prime numbers, and define the Legendre symbol as:

$$\left(\frac{q}{p}\right) = \begin{cases} 1 & \text{if } n^2 \equiv q \bmod p \text{ for some integer } n \\ -1 & \text{otherwise} \end{cases}$$

Then:

$$\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{\frac{p-1}{2}\frac{q-1}{2}}.$$

route node rule: No node can be used more than once.

same node rule: When a node is used, all descendant nodes and all ancestor nodes of this node cannot be used.

single-term: Instead of using many terms, each for a single bit of the challenge, the system uses a single term for a large number of possible challenges. Further details see [4].

untraceable: same as anonymous, privacy property: transactions cannot be linked to users’ identities.

unlinkable: coins divided from the same coin can not be linked.

Williams integer: N = P*Q (P, Q are prime) and P = 3 (mod 8) and Q = 7 (mod 8). Further details see [2].