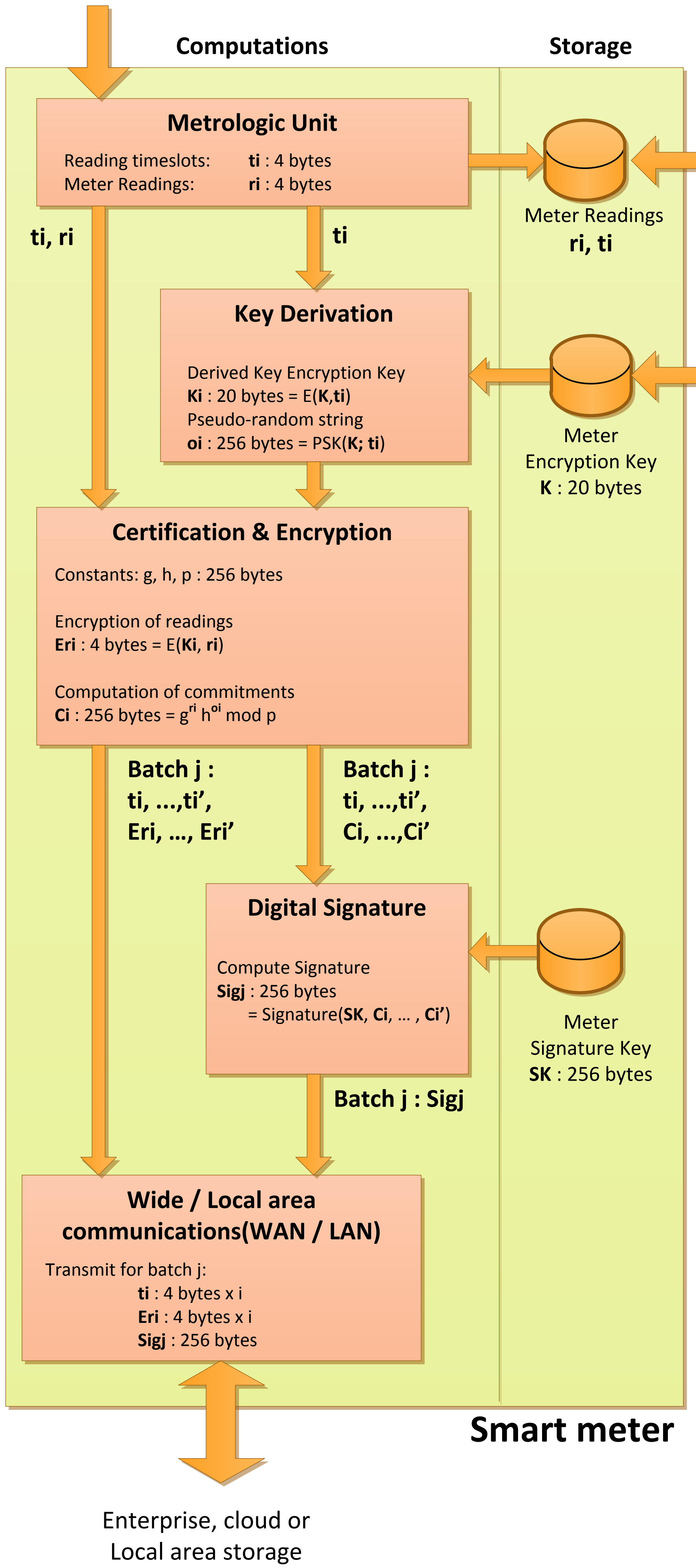
Privacy-friendly smart metering – Meter Modifications

A guide for meter manufacturers



Computations

Storage

Metrologic Unit

Reading timeslots: **ti** : 4 bytes
Meter Readings: **ri** : 4 bytes

ti, ri

ti

Meter Readings
ri, ti

Key Derivation

Derived Key Encryption Key
Ki : 20 bytes = E(**K**,**ti**)
Pseudo-random string
oi : 256 bytes = PSK(**K; ti**)

Meter
Encryption Key
K : 20 bytes

Certification & Encryption

Constants: g, h, p : 256 bytes

Encryption of readings
Eri : 4 bytes = E(**Ki**, **ri**)

Computation of commitments
Ci : 256 bytes = g^{ri} h^{oi} mod p

Batch j :
ti, …,ti',
Eri, …, Eri'

Batch j :
ti, …,ti',
Ci, …,Ci'

Digital Signature

Compute Signature
Sigj : 256 bytes
= Signature(**SK**, **Ci**, … , **Ci'**)

Meter
Signature Key
SK : 256 bytes

Batch j : Sigj

Wide / Local area communications(WAN / LAN)

Transmit for batch j:
ti : 4 bytes x i
Eri : 4 bytes x i
Sigj : 256 bytes

Smart meter

Enterprise, cloud or
Local area storage

Overview

Reading generation & Storage: Current meters generate readings of aggregate consumption within a time slot of ½ hour. Regulators require those readings to be stored within the meter for 6 – 12 months and accessible to users though the LAN or WAN.

Key storage & derivation: To preserve privacy raw readings are only made available to the customer. When stored or transmitted outside the meter they are encrypted using derivatives of a symmetric key K, known only to the customer. This key can be input by the customer or generated by the meter and provided to the user though labelling, a display or LAN.

Certification of readings: Raw meter readings are encoded & certified in a manner that makes them appropriate for further privacy preserving processing, such as privacy friendly protocols for computing bills. This involves computing a "commitment" for each reading two modular exponentiations, one modular multiplication, and the use of a pseudorandom string derived from the key K.

Batching, signing: readings are batched per day or week to be transmitted to the utility providers. A single digital signature is computed per batch over the commitments and timeslots of the readings. Commitments are not transmitted, only their signature, to reduce communication overheads.

Transmission: Encrypted readings, time slots, and the signature associated with their batch are sent to the supplier. The overhead over a traditional meter only involves a single signature amortised over multiple readings.

	Eri	ti	Sigj	Total	Overhead
Daily	192	192	256	384	66.7%
Weekly	1334	1334	256	2668	9.6%
Monthly	5952	5952	256	11904	2.2%

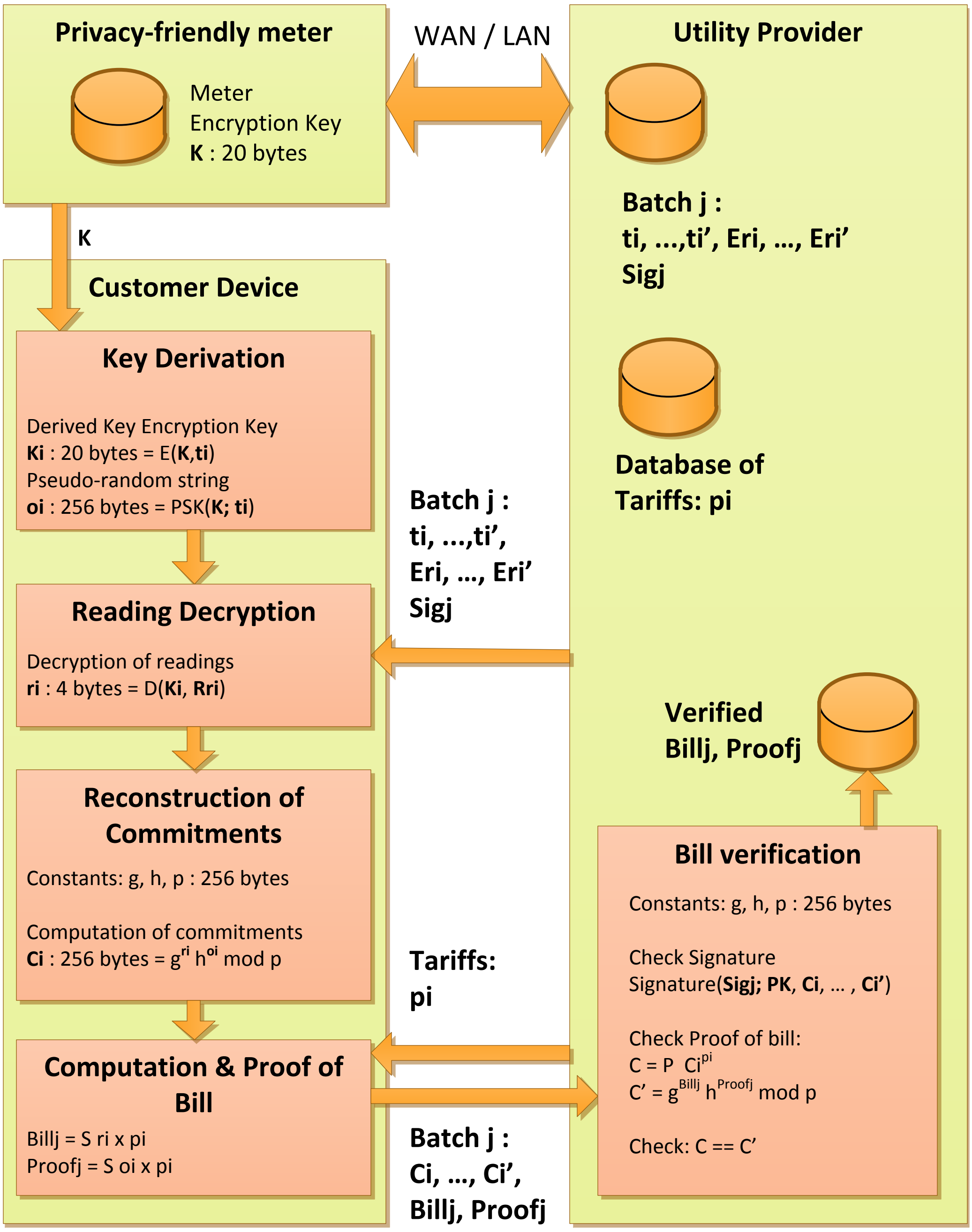
Overheads: The table above shows the communication overhead by batching period. A small amount of storage is required for the symmetric and signing key. Each reading requires 3 key derivation operations, 3 modular operations, and an encryption operation.

For more information contact:
Dr George Danezis (gdane@microsoft.com)

Microsoft Research

Privacy-friendly smart metering – Overall protocol for billing

A guide for meter manufacturers



Privacy-friendly meter

Meter Encryption Key
K : 20 bytes

WAN / LAN

K

Customer Device

Key Derivation

Derived Key Encryption Key
Ki : 20 bytes = E(**K**,**ti**)
Pseudo-random string
oi : 256 bytes = PSK(**K; ti**)

Reading Decryption

Decryption of readings
ri : 4 bytes = D(**Ki**, **Rri**)

Reconstruction of Commitments

Constants: g, h, p : 256 bytes

Computation of commitments
Ci : 256 bytes = $g^{ri} h^{oi}$ mod p

Computation & Proof of Bill

Billj = S ri x pi
Proofj = S oi x pi

Batch j :
ti, …,ti’,
Eri, …, Eri’
Sigj

Tariffs:
pi

Batch j :
Ci, …, Ci’,
Billj, Proofj

Utility Provider

Batch j :
ti, …,ti’, Eri, …, Eri’
Sigj

Database of Tariffs: pi

Verified Billj, Proofj

Bill verification

Constants: g, h, p : 256 bytes

Check Signature
Signature(**Sigj; PK**, **Ci**, … , **Ci’**)

Check Proof of bill:
C = P Ci^{pi}
C’ = $g^{Billj} h^{Proofj}$ mod p

Check: C == C’

Overview

Readings database: Meters transmit encrypted readings, their metadata, and a signature to be stored by the utility provider or any other party. Without the knowledge of key K they cannot be decrypted thus preserving the privacy of fine grained consumption.

Policy database: Utility providers know the tariff pi to be applied to the consumption of each customer by billing timeslot ti.

Key Derivation & Decryption: A customer device – smartphone, computer, web-browser, or even the meter – receives the encryption key K and reconstructs all secrets necessary to decrypt the readings in batch j.

Commitments: The parameters of commitments Ci are reconstructed after decryption and there are recomputed. The signature of batch j can be verified to ensure the readings were genuinely emitted by the meters.

Billing, proof & transmission: The bill is computed using the decrypted readings and appropriate tariffs. A proof of correctness is also computed using the commitment parameters oi. The bill, proof and reconstructed commitments are sent back to the utility provider for verification.

Verification: The utility provider checks the signature matches the received commitments, and that the bill and proof were calculated using the correct commitments and tariffs. Upon verification the bills is stored for further processing.

Overheads: The calculation of the bill a, proof and their verification ensure the correctness of the billing. As a result customer computations can be performed on commodity hardware with access to broadband connectivity. As such the cost of reconstructing commitments, calculating bills, proofs and their transmission is negligible. Verification of bills can be performed on backend servers – it is cheap and can be highly parallelized to support millions of customers.

Advanced capabilities

More advanced billing procedures, using non-linear tariffs, arbitrary functions of consumption, look-up tables keyed by multiple readings, and linear combinations of readings can be supported. The meter logic remains unchanged, and only the proof an verification procedures are different.

Security overview

Meter readings, in an encrypted form, are sent to utility providers and the keys to decode them are within meters ensuring a “utility robust” **high availability** fallback in case customers refuse to compute their bills regularly. Yet, the encryption of readings ensures that utilities do not have access to detailed consumption information on a casual basis. Readings are only in clear within the meter and on customer devices ensuring **privacy**. The billing protocols, the signatures and associated proof, ensure that the computation of the bill cannot be manipulated or underreported providing **high integrity** to the billing process.

For more information contact:
Dr George Danezis (gdane@microsoft.com)

Microsoft Research